

# Relational Classification using Multiple View Approach with Voting

Shraddha Modi  
Computer Engg. Department,  
Silver Oak College of  
Engineering & Technology,  
Ahmedabad, India

## 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. Various algorithms have been proposed to build accurate and scalable classifiers in data mining. These algorithms are only applied to single table. Today most real-world data are stored in relational format which is popular format for structured data which consist of tables connected via relations (primary key/ foreign key). So single table data mining algorithms cannot deal with relational data. To classify data from relational format need of multirelational classification arise which is used to analyze relational data and used to predict behaviour and unknown pattern automatically. For multirelational classification, various techniques are available which include upgrading existing algorithm, flatten relational data and multiple view approach. Multiple view approach learns from multiple views of a relational data and then combines the result of each view to classify unknown data. This paper presents proposed algorithm and experimental results for multiple view approach with voting as a view combination technique.

## General Terms

Multi relational data mining, Multi relational classification, Multi view classification, voting.

## Keywords

Inductive logic programming, Multi relational classification, Multiple view, Multi-view, Relational database.

## 1. INTRODUCTION

Multi relational classification is an important part of Multi relational data mining which involve applications like credit card fraud detection, disease diagnosis system, financial decision making system, information extraction and face recognition applications. There are many algorithms available for classification but they are applied only on single/flat file. For multirelational classification, various techniques are available which include upgrading existing algorithm, flatten relational data and multiple view approach. In upgrading approach, existing data mining algorithms are upgraded to deal with relational format. In flattening approach, relational data are converted into single table and then any conventional data mining algorithm is applied. Flattening is not fruitful option because conversion process can lose some essential information from data. Lots of work has been done in the upgrading and flattening approach. Another approach called multiple view based relational classification is also used to predict behaviour of unknown pattern from relational database

which neither upgrade propositional algorithm nor flatten the data. It makes use of different view of relational data and applies propositional data mining algorithm on views. Results of different view are then combined by voting technique. Multiple view approach uses multi view learning framework which describes the setting of learning from data where observations are represented by multiple independent sets of features and learning from these representations separately can lead to better gains than merging them into a single dataset.

The paper is organized as follows. Section 2 presents related work. Section 3 introduces problem definition. In Section 4 proposed algorithm with weighted voting is introduced. Section 5 presents the detailed discussion for working of the algorithm. Section 6 contains experimental analysis of proposed algorithm. Finally, Section 7 concludes this paper and outlines the future work.

## 2. RELATED WORK

The relational classification across multiple database relations is divided into two steps with the same propositional classification to learn classification model from examples and to classify and test using the model. Relational classification makes use of additional information of related objects using multiple relations.

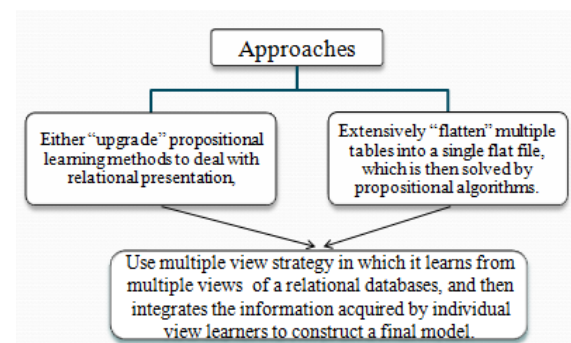


Fig 1: Approaches to Multirelational classification

There are mainly two categories of relational classification as shown in Figure 1., either propositional learning algorithm should be upgraded to deal with relational format (upgrading) or relational format should be converted into flat format and then applying propositional learning algorithm (flattening). Another approach called multiple view approach is also used which combines upgrading and flattening approach because it uses propositional algorithm so no need to upgrade algorithm

and combines the result of different views. All these approaches are described next.

### **Flattening Approach**

In this approach multiple relations are converted into single flat file and then propositional algorithms are used for classification. LINUS algorithm uses flattening approach in which background knowledge is used to introduce new attributes into the learning process. Following the same lines, propositionalization approaches such as DINUS and SINUS were designed. Also RELAGGS propositionalization method is introduced which shows superior predictive performance over the DINUS and SINUS algorithms. In the RELAGGS strategy, new features are constructed for the target relation through an aggregation process, where function dependencies in the database are considered [1]. So in RELAGGS aggregate operators are used to transform multiple relations into a single table in order to be able to use propositional algorithms for the learning. Flattening approach is simpler than any other approaches but yet it is suffered from some problems. It require efforts for pre-processing and after pre-processing multiple files are converted in one big flat file which may contain large amount of missing values and large numbers of additional attributes. So this may cause further scaling challenges and over-fitting problems for propositional algorithms.

### **Upgrading Approach**

In this approach, existing single file data mining algorithms are upgraded to handle relational database. Lots of work has been done in the area of upgrading. It mainly includes ILP based relational classification, Relational database based relational classification, Association or frequent pattern based relational classification. The ILP paradigm employs logic programs to describe patterns. The logic programs need to be induced from a database of logical facts, hence the name Inductive Logic Programming. ILP searches for hypothesis from examples based on background knowledge that will be used to classify unknown data. A major shortcoming of ILP is the computational demand that results from the large hypothesis spaces searched [2]. Also for logic based classification, relational data should be transformed into logic programs in pre-processing stage, which determines the relatively weak relation in database [3]. It also Suffer from scalability problems with regard to the number of relations in database [6]. CART and C 4.5 are famous propositional decision tree algorithms. To deal with relational data, these algorithms are upgraded and named TILDE [4] and SCART [5]. The expressivity of these trees is shown to be larger than the flat logic programs which are typically induced by classical ILP systems [4]. TILDE is more efficient than most traditional ILP approaches due to the divide-and-conquer nature of decision tree algorithm [6]. Multi relational decision tree learning algorithm (MRDTL) [7] is an upgraded propositional algorithm which adds selection graphs as the nodes to the tree through a process of successive refinement until some termination criterion is met. Selection graph represent the relationship between tables which is then easily handle by database language SQL. The paper [8] proposed MRDTL-2 algorithm, which improves the calculation efficiency and information loss of MRDTL. Also For dealing with the noise and uncertainty encountered in most real-world domains, probability is introduced into LBRC to integrate the advantages of both logical and probabilistic approaches to knowledge representation and reasoning. At present, the

method mainly includes Inductive Logic Programming and Bayesian Networks, ILP and Stochastic Grammars. Probabilistic relational models (PRMs) extend Bayesian networks with the concepts of objects, their properties, and relations between them [9]. In relational database, only target table has class variable. So it is necessary to join target table to background tables. For this, Tuple ID propagation is used to perform virtual join between non-target relations and the target relations. The paper [6] proposed CrossMine for relational classification which is an extension of FOIL approach. CrossMine uses tuple ID propagation, in which it propagates data into related tables through foreign key relationships instead of performing a physical-join in the database. Based on tuple ID propagation, in [10] Graph-NB algorithm is presented which is upgraded from Naïve Bayesian classifier. Downside of tuple id propagation is that sometime too many IDs may be propagated to each tuple in a relation, which makes it hard to limit the time/space complexity of the algorithm.

In association or frequent pattern based relational classification, gSpan is proposed for frequent pattern mining in graphs, which can be applied on multi-relational data [11]. In [12] Warmer is presented, which extends APRIORI to mine association rules in multiple relations. In [13], A Multirelational classification algorithm based on association rules (MrCAR) is proposed which uses class frequent closed itemsets and reflects the association between class labels and other itemsets, and used to generate classification rules. In association based classification many rules produced by standard classification systems are difficult to understand because these systems often use only domain independent biases and heuristics [3].

### **Multiple view Approach**

In multiple view approach, problem can be characterized using different representations (views), and that learning from these representations separately can lead to better gains than merging them into a single dataset. A multiple view strategy is proposed in [1], which enable us to classify relational objects by applying conventional data mining methods, while there is no need to flatten multiple relations to a universal one. It employs multiple view learners to separately capture essential information embedded in individual relation. The acquired knowledge is incorporated into a meta learning mechanism to construct the final model. MVC framework is mainly consisting of five stages which are described below:

#### **1) Information Propagation Stage:**

First of all, it constructs training data sets for use by a number of view learners, using a relational database as input. The information propagation Element propagates information from the target relation to the background relations, based on the foreign key links.

- a) Directed foreign key chain propagation: When there is a direct relationship between target relation and background relation, then we can directly propagate tuple id and class label to background relation because there is a direct reference key between both tables.
- b) Undirected foreign key chain propagation: When there is an indirect relationship between target relation and background relation, then tuple id and

class label cannot be propagated to background relation directly because we need another table which connects both tables.

Whether directed or undirected foreign key chain propagation is used, shortest directed or undirected foreign key chain (the foreign key chain with less involved foreign key joins) should be chosen. Because shorter foreign key chain will save storage and computational cost. Also longer chain with too many joins usually becomes very weak in a relational database and provides less information with compared to shorter foreign key.

## 2) Aggregation Stage:

All the views got from previous stage are with one-to-many relationship with respect to primary key so need to apply an aggregation function which summarizes information embedded in multiple tuples and squeeze them into one row using the primary key from the target relation. Different aggregate functions like COUNT, SUM, AVG, MIN, MAX, STDDEV are applied based on the type of attribute whether it's nominal, binary or numeric. For Nominal attribute, COUNT function is used to count total occurrences of values of attribute. For Binary attribute the aggregation procedure creates two new attributes. The attribute1 stores the number of occurrences of one of the values of attribute, and attribute2 stores the number of occurrences of the other. For numeric attribute, six new attributes are created with COUNT, SUM, AVG, MIN, MAX, STDDEV aggregate functions. By applying the basic aggregation functions in SQL, new features are created to summarize information stored in multiple tuples. Each newly constructed background relation is then used as training data for a particular view learner.

## 3) Multiple Views Construction Stage:

It constructs various hypotheses on the target concept, based on the multiple training data sets given by the Aggregation Stage. Conventional single-table data mining algorithms such as such as Decision Trees, SVMs, or Neural Networks are used in order to learn the target concept from each view of the database separately. In this stage, a number of view learners, which differ from one another, are trained. The results from the learners will be validated and combined to construct the final classification model.

## 4) View Validation Stage:

The trained view learners need to be validated before being used by the meta learner. This processing is needed to ensure that they are sufficiently able to learn the target concept on their respective training sets. In this view efficiency is measured. Only learners with predictive performance better than random guessing are used to construct the final model (view efficiency).

- a) View efficiency: The view validation has to be able to differentiate the strong learners from the weak ones. In other words, it must identify those learners which are only able to learn concepts that are strictly more general or specific than the target one. In this method, error rate is used in order to evaluate the predictive quality of a view learner. That is, learners with training error greater than 50% are discarded. In other words, only learners with

predictive performance better than random guessing are used to construct the final model.

## 5) View Combination Stage:

In this the resulting multiple view learners are incorporated into a meta learner to construct the final classification model (Stacking). The meta learner is called upon to produce a function to control how the view learners work together, to achieve maximum classification accuracy. This function, along with the hypotheses constructed by each of the view learners, constitutes the final model. This combination process contains two steps. Firstly, a meta training data set is generated. Each instance of the meta training data consists of the class label predictions, made by the individual learner on a specific training example, along with the original class label for that example. Secondly, a meta-learner is trained using the meta data constructed to achieve a strong predictive performance, as the final classification model.

In the last stage of the learning, this approach returns a set of view learners, along with a meta learner which knows how to combine these multiple learners to achieve a strong predictive performance.

## 3. PROBLEM DEFINITION

For relational classification, relational database is needed 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 database. So we can say relational database  $R$  is a collection of tables  $R = \{R_1, R_2, \dots, R_n\}$ . A table  $R_i$  consists of a set of tuples  $T_{R_i}$  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  $R_{b1}, R_{b2}, \dots, R_{bn}$ . Each tuple  $x \in T_{R_i}$  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:

$$y = F(x, R_t, R_{b1}, R_{b2}, \dots, R_{bn}), x \in T_{R_t}$$

## 4. PROPOSED ALGORITHM

In this section, we present the multi view classification (MVC) algorithm with weighted voting technique. Base of this algorithm is multiple view framework in which learning is done from multiple views of relational data. To combine the result of multiple view, average of probability rule from vote meta classifier is used. Voting is used to combine the predicted classifications from multiple models, or from the same type of model for different learning data.

Algorithm represents MVC with weighted voting technique. Focus points of this algorithm are propagation of essential information to background relations, view construction and propositional learning, Best-worst weighted vote scheme and

voting view combination scheme. Working of the algorithm is discussed in next section.

**Algorithm:** MVC with Weighted voting

**Input:** A relational data  $DB=\{T_{target}, T_1, T_2, \dots, T_n\}$ , consisting of a target relation  $T_{target}$  (with target variable  $Y$  and primary key  $ID$ ) and a set of background relations  $T_i$ . The value  $n$  denotes the number of background tables while  $w$  denotes the number of possible values of  $Y$ .

A conventional single-table learning algorithm  $L$ .

**Output:** Class labels of target tuples.

**Procedure:**

i. Divide tuples of  $T_{target}$  into sets  $S_{train(k)}$  and  $S_{test(k)}$ , using  $k$ -fold cross validation.

ii. Do for each fold  $k$  of the  $k$ -fold cross validation

iii. Do for  $i = 1, 2, \dots, n$

1. Propagate  $Y$  and  $ID$  of each tuple in  $S_{train(k)}$  to every background relation  $T_i$ , using the shortest foreign key chain which references the target table  $S_{train(k)}$  in order to form the data set  $D_i$ .

2. Apply aggregation functions on  $D_i$ , grouping by  $ID$

3. Call algorithm  $L$ , providing it with  $D_i$ , obtaining observation  $V_i$  for the target concept.

4. Calculate the error for  $V_i$ , denoted as  $e_i$ . If  $e_i > 0.5$ , discard  $V_i$ .

5. Obtain accuracy for observation  $V_i$ .

End do

iv. Call algorithm  $L$ , providing it with  $S_{train(k)}$ , obtaining observation  $V_0$ .

v. Apply Best-worst weighted vote to every view.

Apply weight=0 to lowest accurate observation  $V_i$  and apply weight=1 to highest accurate observation  $V_i$ . The rest of the observations  $V_i$  are rated linearly between these extremes and are calculated as following equation:

$$weight_i = 1 - \frac{(e_i - e_b)}{(e_w - e_b)}$$

Where  $e_i$  = Error of view  $V_i$ ,

$e_b$  = Minimum  $\{e_i\}$ ,

$e_w$  = Maximum  $\{e_i\}$

vi. Do for  $x = 1, 2, \dots, m$ , where  $m$  is the number of tuples in  $S_{test(k)}$

Do for  $s = 0, 1, \dots, p$ , where  $p$  is the number of observations learned and accepted in the validation of step 5 of iii.

If ( $s = 0$ )

Call  $V_0$ , providing it with tuples  $x$ . Return predictions  $\{P^{yq}_{V_0}(x)\}$ , where  $q \in \{1, 2, \dots, w\}$ .

else

Retrieve corresponding tuples from background table  $T_i$ , denoted as  $R$ , following the foreign key chains.

If ( $R = 0$ )

Return  $\{P^{yq}_{V_i}(x) = 1.0 / w\}$ , where  $q \in \{1, 2, \dots, w\}$ .

else

Apply aggregation functions to  $R$ , grouping by  $ID$ , yielding  $R_A$

Call  $V_s$ , providing it with  $R_A$ .

End if

End if

End do

vii. Apply voting.

Initialize probs array with null value and size according to no. of class labels.

Do for  $s = 0, 1, \dots, p$ , where  $p$  is the number of observations learned and accepted in the validation of step 4 of iii.

Do for  $i = 1, 2, \dots, s$

Get Observation\_probability $_i$ .

Add  $Weight_i$  of step v. of observation  $V_i$  to each value of Observation\_probability $_i$ .

Apply voting.

End Do

Return class label with highest value in probs.

End do

End do

End do

End Procedure

## 5. WORKING OF ALGORITHM

In Algorithm, 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-view training set from each background relation, and one from the target table. In multi-view learning framework, it is important to provide each multi-view learner 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. These steps are described in step iii (1-2).

After constructing the multi-view training data sets, the algorithm call upon the multi-view learners to learn the target concept from each of these sets. Each learner 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. This procedure is explained in step iii (3). In this way, all multi-view learners make different observations on the target concept based on their perspective. The multi-view learners trained are needs to be validated before being used by the meta-learner. The view validation has to be able to differentiate the strong learners from the weak ones. In other words, it must identify those learners which are only able to learn concepts that are strictly more general or specific than the target one. In this method, error rate is used in order to evaluate the quality of a multi-view learner. Learners with training error greater than 50% are discarded. This procedure is explained in step iii (4).

In step iv target table is learned using propositional learning algorithm. In step v Best-worst weighted vote is assigned to every view. As per below equation

$$weight_i = 1 - \frac{(e_i - e_b)}{(e_w - e_b)}$$

First of all weight=0 is assigned to lowest accurate observation  $V_i$  and weight = 1 to highest accurate observation  $V_i$ . The rest of the observations  $V_i$  are rated linearly between these extremes. This eq. 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 [25].

After training, in steps vi firstly, for each training tuple  $x$  from the target table, a multi-view learner  $V_i$  will retrieve the tuple from background knowledge table  $T_i$  corresponding to the key reference (a directed or undirected foreign key chain). If there is no corresponding tuple in table  $T_i$ , an equal probability is assigned for each class label, i.e.  $P^{yq}_{V_i}(x)$  values equal to  $1/w$  are returned. If a corresponding tuple is found, aggregation operations are applied on tuple 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.

In step *vii* different voting techniques are applied to combine result of different views which consist of Average of probability, Product of probability, Majority voting, Minimum probability and Maximum probability. Here weight acquired in step *v* is added to every Observation probability of observations and then average of probability (one of the voting technique) is applied on observation probability which may return accurate class label to unknown testing tuple.

## 6. EXPERIMENTAL ANALYSIS

In this section, results from all four learning task are presented and compared with algorithms from flattening approach and upgrading approach. Summary of four learning tasks are shown in table 1. RelAggs and SimFlat are of flattening approach where as FOIL, CrossMine, and TILDE are of upgrading approach. All experiments on MVC and MVC with weighted voting algorithm are applied on J48 decision tree and naive bayes classifier. For comparison J48 is taken because the learning and classification steps of decision tree are simple and fast. Bayesian classifier is chosen because it can predict class membership probabilities (probability that a given tuple belongs to a particular class). The default settings of these two learning methods were used. Each of these experiments produces accuracy results using ten-fold cross validation. In MVC with weighted voting, out of five schemes, product of probability is selected for comparison.

**Table 1. Summary of dataset used in experiment**

Dataset	#tuples in target table	#tables	target class distribution
Financial 234	234	8	203:31
Financial 682	682	8	606:76
Mutagenesis	188	3	125:63
Thrombosis	770	5	695:75

**Experiment 1:** To compare accuracy and running-time between algorithms for all database (Homogeneous classifiers).

### 1. Classifiers Used:

In MVC, Classifier for View Learning: J48, MetaLearning:J48

In MVC with weighted voting, Classifier: J48

**Table 2. Accuracy obtained using J48 Classifier**

Database	Accuracy (%)						
	RelAggs	SimFlat	FOIL	CrossMine	TILDE	MVC	MVC with weighted voting
Financial 234	90.2	89.7	74.4	88.0	86.8	93.59	94.02
Financial 682	92.1	92.4	83.9	90.3	88.9	94.13	94.13
Mutagenesis	85.1	88.3	85.7	85.7	85.6	88.30	100.00
Thrombosis	100	90.4	100	90	90.4	99.22	99.48

**Table 3. Running time obtained using J48 Classifier**

Database	MVC	MVC with weighted voting
Running time in ms		
Financial234	964	765
Financial682	3635	1747
Mutagenesis	3416	2901
Thrombosis	719	499

### 2. Classifiers Used:

In MVC, Classifier for View Learning: Naïve bayes, MetaLearning: Naïve bayes

In MVC with weighted voting, Classifier: Naïve bayes

**Table 4. Accuracy obtained using Naïve bayes Classifier**

Database	Accuracy (%)						
	RelAggs	SimFlat	FOIL	CrossMine	TILDE	MVC	MVC with weighted voting
Financial 234	85.9	75.6	74.4	88	86.8	62.82	87.18
Financial 682	73.8	71.4	83.9	90.3	88.9	85.34	93.70
Mutagenesis	85.1	86.2	85.7	85.7	85.6	89.89	100.00
Thrombosis	98.4	86	100	90	90.4	86.10	99.35

**Table 5. Running time obtained using Naïve bayes Classifier**

Database	MVC	MVC with weighted voting
Running time in ms		
Financial234	1482	999
Financial682	4666	2527
Mutagenesis	1700	1311
Thrombosis	1201	546

In this experiment 1, same classifier (naïve bayes, J48) is used for view learning and meta learning in MVC algorithm as well in MVC with weighted voting algorithm. Also to evaluate the performance of both algorithms in terms of running time, execution time of both algorithm is also presented in table 3 and 4. In almost all cases MVC with weighted voting algorithm is giving good predictive performance with compare to existing MVC algorithm. In the case of thrombosis database, RelAggs and FOIL algorithms give 100% accuracy, whereas MVC with weighted voting gives 99.48% accuracy which is comparable predictive performance near to 100%. Also in the case of naive bayes classifier, MVC with weighted voting provides more accurate result which implies that probabilistic classifier improves the result when weight is added to the probability distribution. Now if we consider running time, MVC with weighted voting require very less running time with compare to MVC algorithm.

**Experiment 2:** To compare accuracy and running-time between algorithms for all database (Heterogeneous classifiers).

### 1. Classifiers Used:

In MVC, Classifier for View Learning and meta learning: J48, Naïve bayes

In MVC with weighted voting, Classifier: J48, Naïve bayes

**Table 6. Accuracy obtained using Heterogeneous Classifier**

Database	Accuracy (%)						
	RelAggs	SimFlat	FOIL	CrossMine	TILDE	MVC	MVC with weighted voting
Financial 234	90.2	89.7	74.4	88	86.8	90.60	<b>94.02</b>
Financial 682	92.1	92.4	83.9	90.3	88.9	91.35	<b>94.13</b>
Mutagenesis	85.1	88.3	85.7	85.7	85.6	89.89	<b>100.00</b>
Thrombosis	<b>100</b>	90.4	<b>100</b>	90	90.4	99.22	<b>99.48</b>

**Table 7. Running time obtained using Heterogeneous Classifier**

Database	MVC	MVC with weighted voting
Running time in ms		
Financial234	2134	811
Financial682	4122	1950
Mutagenesis	2144	2883
Thrombosis	1922	561

In this experiment 2, Different classifiers are used for view learning and meta learning in MVC algorithm. For MVC with weighted voting algorithm heterogeneous classifiers are used. Also to evaluate the performance of both algorithms in terms of running time, execution time of both algorithm is also presented in table 7. For mutagenesis database which has fewer relations, MVC with weighted voting predicts 100% accuracy, which is far better than all the algorithms. This implies, as number of background relations decrease, accuracy may improve. As shown in table 7, existing MVC algorithm require double running time with compare to MVC with weighted voting algorithm except mutagenesis database.

## 7. CONCLUSION AND FUTURE EXTENSIONS

Multi-relational data mining (MRDM) approaches look for patterns that involve multiple tables (relations) from a relational database. Relational classification has got more and more great concern from the large number of international researchers. In this paper, an algorithm MVC based on voting combination technique is proposed. Also weighted vote is given to views based on their accuracy and added in probability so that individual performance of views can be considered.

Pre-processing techniques can be incorporated to algorithm which removes irrelevant relations and features from database and may improve result. More techniques for acquiring information from relational data can be incorporated. Further investigation of the method to construct the training data sets from the multi-relational databases is needed. Experiment with different weight assignment techniques for views can vary the result of the algorithm. 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.

## 8. REFERENCES

- [1] 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
- [2] Raymond J. Mooney, Prem Melville, Lappoon Rupert Tang, "Relational Data Mining with Inductive Logic Programming for Link Discovery", Appears in the Proceedings of the National Science Foundation Workshop on Next Generation Data Mining, Nov. 2002
- [3] 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
- [4] Blockeel, H. , Top-down induction of first order logical decision trees, Artificial Intelligence Journal, vol.101,pp.285-297, 1998
- [5] Kramer S. , Widmer G. , "Inducing Classification and Regression Tress in First Order Logic", Relational Data Mining, pp.140-159, Springer, 2001
- [6] 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
- [7] Hector Ariel Leiva, Shashi Gadia, Drena Dobbs, "MRDTL: A multi-relational decision tree learning algorithm", Proceedings of the 13th International Conference on Inductive Logic Programming, 2002
- [8] Atramentov A. , Leiva H. , Honavar V. , "A Multirelational Decision Tree Learning Algorithm - Implementation and Experiments", ILP LNCS, Vol.2835, pp. 38-56, 2003
- [9] Lise Getoor, Nir Friedman, Daphne Koller, Avi Pfeffer, Ben Taskar, "Probabilistic Relational Models", In An Introduction to Statistical Relational Learning, August 2007
- [10] Liu H. , Yin X. ,Han J. , "A Efficient Multirelational Naïve Bayesian Classifier Based on Semantic Relationship Graph", In MRDM'05 Proceedings of 4<sup>th</sup> international workshop on MRDM, 2005
- [11] X. Yan, J. Han, "gSpan: Graph-based substructure pattern mining". In Proc. 2002 Int. Conf. on Data Mining (ICDM'02), Maebashi, Japan, Dec. 2002.
- [12] Dehaspe L. , Raedt D. , "Mining Association Rules in Multiple Relations", In Proceedings of the ILP, Springer-Verlang, London UK, pp.125-132, 1997
- [13] 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
- [14] Francisco Moreno-Seco; Jose M. Inesta; Pedro J. Ponce de Leon; Luisa Mico, "Comparison of classifier fusion methods for Classification in pattern recognition tasks", 20