Journal of Computer Science 7 (12): 1914-1920, 2011
ISSN 1549-3636

BioDARA: Data Summarization Approach to Extracting Bio-Medical Structuring Information

[1]Chung Seng Kheau, [2]Rayner Alfred and [3]Joe Henry Obit
[1,2]School of Engineering and Information Technology,
University Malaysia Sabah, Malaysia
[3]School of Informatics Science Labuan, University Sabah, Malaysia

Abstract: Problem statement: Due to the ever growing amount of biomedical datasets stored in multiple tables, Information Extraction (IE) from these datasets is increasingly recognized as one of the crucial technologies in bioinformatics. However, for IE to be practically applicable, adaptability of a system is crucial, considering extremely diverse demands in biomedical IE application. One should be able to extract a set of hidden patterns from these biomedical datasets at low cost. **Approach:** In this study, a new method is proposed, called Bio-medical Data Aggregation for Relational Attributes (BioDARA), for automatic structuring information extraction for biomedical datasets. BioDARA summarizes biomedical data stored in multiple tables in order to facilitate data modeling efforts in a multi-relational setting. BioDARA has the advantages or capabilities to transform biomedical data stored in multiple tables or databases into a Vector Space model, summarize biomedical data using the Information Retrieval theory and finally extract frequent patterns that describe the characteristics of these biomedical datasets. **Results:** the results show that data summarization performed by DARA, can be beneficial in summarizing biomedical datasets in a complex multi-relational environment, in which biomedical datasets are stored in a multi-level of one-to-many relationships and also in the case of datasets stored in more than one one-to-many relationships with non-target tables. **Conclusion:** This study concludes that data summarization performed by BioDARA, can be beneficial in summarizing biomedical datasets in a complex multi-relational environment, in which biomedical datasets are stored in a multi-level of one-to-many relationships.

Key words: Information extraction, data summarization, relational data mining, relational database, biomedical datasets, summarization performed, datasets stored, multiple tables, relational attributes

INTRODUCTION

Biomedical information extraction from structured biomedical data stored in relational databases refers to data summarization applied to relational biomedical data. One of the approaches of data summarization for relational biomedical data is clustering. Clustering is a process of grouping data that shares similar characteristics into groups. Despite the increase in volume of biomedical datasets stored in relational databases, only few studies handle clustering across multiple relations (Kirsten and Wrobel, 1998; 2000). In a biomedical dataset stored in a relational database with one-to-many associations between records, each table record (or object) can form numerous patterns of association with records from other tables. For example, in a mutagenesis dataset, there are two classes of molecules (active and non-active molecules). These molecules can be represented in molecular structures representation, as shown in Fig. 1. At the same time, the information of these molecules can be stored in relational tables, as shown in Fig. 2.

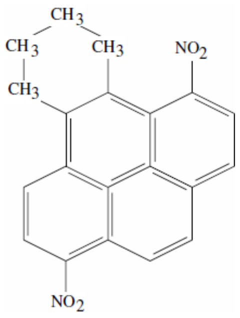

Fig.1: An example of a molecular structure and bonding

Corresponding Author: Chung Seng Kheau, School of Engineering and Information Technology, Universiti Malaysia Sabah Malaysia

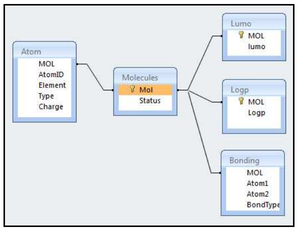

Fig. 2: A biomedical dataset stored in a relational database with two levels of one-to-many relationship

In Fig. 2, the scenario in which a single object has multiple instances is illustrated. In this scenario, relation Molecules has a one-to-many relationship with relations Atom and Bonding, through the association of field Mol.

Clustering in a multi-relational environment has been studied in Relational Distance-Based Clustering (RDBC) (Kirsten and Wrobel, 2000). Clustering (Hofmann and Buhnmann, 1998; Hartigan, 1975) is an unsupervised learning technique, that is, it can operate on un-annotated data. However, it can be used as the first step of a supervised learning tool. For instance, a dataset split into classes can be clustered (without making use of the class labels) and then associations between clusters and classes learned using one of the various well known supervised learning tools. This is the case in RDBC, where the role of this tool is performed by a decision tree learner. The approach proposed in this study follows the same strategy, combining a novel clustering technique with C4.5.

In RDBC, the similarity between two objects is defined on the basis of the tuples that can be joined to each of them. In this way, each of the two objects is expanded into a set of records and the two sets are compared as follows: for each record in one set, the closest match in the other set is found and their distance added. The distance between two such records is measured in the usual ways, comparing each pair of attributes in turn, depending on the types of attributes involved, e.g., as differences of numerical values, or a Hamming distance in the case of categorical values. However, the RDBC process of computing the distance between two objects is very expensive, since the process compares repeatedly components of first-order instances where each comparison is eventually reduced to a propositional comparison of elementary features. In addition to that, the RDBC approach only considers the minimum distance measured between instances to differentiate two objects and may not generate good clustering results, which leads to less meaningful clustering results. RDBC approach is also not able to generate interpretable rules. In our approach to clustering in a multi-relational environment, we consider all instances of an object when the distance between two objects is computed. By clustering objects with multiple instances, objects with the same characteristics are grouped together and objects with different characteristics are separated into different groups. Traditional clustering algorithms are based on one representation space, usually a vector space. However, in a relational database system, multiple instances in a non-target table exist for each object in the target table, due to the one-to-many association between multiple instances and the object. To cluster multiple-instance data using the established methods would require to restrict the analysis to a single representation or to construct a feature space comprising all representations.

In this study, we present a data summarization approach, borrowed from the information retrieval theory, to cluster such multi-instance data. This study proposes a technique that considers all available instances of an object for clustering and we show the evaluation results on the mutagenesis dataset. In addition to that, the effect of the number of relevant features on the classification performance is also evaluated. The rest of the study is organized as follows. First, we present related study on data mining in a multi-relational environment. Next, the problem is formalized and the proposed new pre-processing method for the purposes of clustering, called Dynamic Aggregation of Relational Attributes (DARA) (Alfred and Kazakov, 2006a; 2006b; 2007) is introduced. Finally, the experimental evaluation is discussed and then the conclusion section summarizes the study and presents some ideas for future research.

MATERIALS AND METHODS

Learning data in a multi-relational environment: The most popular approach to supervised learning in a multi-relational environment is relational learning. Relational learning is not a new research area and has a long history. (Muggleton and DeRaedt, 1994) introduce the concept of Inductive Logic Programming (ILP) and its theory, methods and implementations in learning multi-relational domains. ILP methods learn a set of existentially quantified first-order Horn clauses that can be applied as a classifier (Salton *et al*., 1975; Srinivasan *et al*., 1996). In a relational learner based on logic-based propositionalization (Kramer *et al*., 2001), instead of searching the first-order hypothesis space directly, one uses a transformation module to compute a large number of propositional features and then uses a propositional learner.

Variants of relational learning include distance-based methods (Horvath *et al*., 2001; Emde and Wettschereck, 1996). The central idea of distance-based methods is that it is possible to compute the mutual distance Emde and Wettschereck (1996) for each pair of objects. Relational Instance-Based Learning (RIBL) algorithms extend the idea of instance based learning to relational learning (Emde and Wettschereck, 1996). Instance-Based Learning (IBL) algorithms (Aha *et al*., 1991) are very popular and a well studied choice (Wettsschereck and Dietterich, 1995) for propositional learning problems. Probabilistic Relational Models (PRMs) (Getoor *et al*., 2001) provide another approach to relational data mining that is grounded in a sound statistical framework. In PRMs, a model is introduced that specifies, for each attributes of an object, its (probabilistic) dependence on other attributes of that object and on attributes of related objects. Propescul *et al*. (2002) proposed a combined approach called Structural Logistic Regression (SLR) that combines relational and statistical learning.

Data stored in a multi-relational environment can be considered as multiple instances of an object stored in the target table. As a result, learning multiple instances can be applied in learning data in a multi-relational environment. In Multi-Instance (MI) learning, instances are organized into bags that are labeled for training, instead of individual instances. Multiple instance learners assume that all instances, in a bag labeled negative, are negative and at least one instance in a bag labeled positive is positive. Several approaches have been designed to solve the multiple instance learning. Dietterich *et al*., (1997) described an algorithm to learn Axis-Parallel Rectangles (APRs) from MI data. Maron introduced a framework called Diverse Density to learn Gaussian concepts (Maron and Lozano-Perez, 1998). Another approach using lazy learning has been investigated in this context as well (Wang and Zucker, 2000). Unlike the former approaches, a framework for learning rules from multiple data was introduced by Most of the approaches (Maron and Lozano-Perez, 1998; Wang and Zucker, 2000) are not able to generate interpretable rule sets or decision trees.

In Relational Distance-Based Clustering (RDBC) (Kirsten and Wrobel, 2000) the similarity between two objects is defined based on tuples joinable with them. The distance measure uses the idea of computing distances by recursively comparing the components of first-orders instances, in which it is highly expensive if we have many tables. In addition to that, RDBC approach only considers the minimum distance measured between instances to differentiate two objects and may not generate good clustering results, which leads to less meaningful clustering results. RDBC approach is also not able to generate interpretable rules. In our approach, we transform the data representation in a multi-relational environment into a vector space model suitable or applicable to clustering operation. By clustering these objects, one can group bags with multiple instances that have similar characteristics that can be extracted, as an interpretable rule to describe the cluster's behaviors.

Multi-relational learning in DARA: We first describe the concept of multi-relational setting for data stored in a relational database. Then, we describe how a single object stored in a target table that is associated with many objects stored in a non-target table can be represented in a vector space model.

The Multi-Relational Setting

In this subsection, we describe the representation of data for objects stored in multiple tables with one-to-many relations. Let DB be a database consisting of n objects. Let R := {R1,…,Rm} be the set of different representations existing for objects in DB and each object may have zero or more than one representation of each Ri, such that |Ri| ≥ 0, where i = 1,…,m. Each object Oi DB, where i = 1,…,n can be described by maximally m different representations with each representation has its frequency:

$$O_i := \{R_1(O_i):|R_1(O_i)|:|Ob(R_1)|,\ldots,R_m(O_i):|R_m(Oi)|:|Ob(R_m)|\},$$

where, Rj(Oi) represents the j-th representation in the i-th object and |Rj(Oi)| represents the frequency of the j-th representation in the i-th object and finally |Ob(Rj)| represents the frequency of object with j-th representation. If all different representations exist for Oi, then the total different representations for Oi is |Oi| = m else |Oi| < m.

In relational instance-based learning, the distance measures are defined based on the attribute's type (Horvath *et al*., 2001) and the distance between two objects is based on the minimum distance between pair of instances from the two objects. In our approach, we apply the vector-space model (Salton *et al*., 1975) to represent each object. In this model, each object Oi is considered as a vector in the representation-space. In particular, we employed the rf-iof term weighting model borrowed from (Salton *et al*., 1975), where in which each object Oi, i = 1,…,n can be represented as:

$$(rf_1\bullet\log(n/of_1), rf_2\bullet\log(n/of_2), \ldots, rf_m\bullet\log(n/of_m))$$

where, rfj is the frequency of the j-th representation in the object, ofj is the number of objects that contain the j-th representation and n is the number of objects. To account for objects of different lengths, the length of each object vector is normalized so that it is of unit length (||orfiof||= 1), that is each object is a vector on the unit hypersphere. In this experiment, we will assume that the vector representation for each object has been weighted using rf-iof and it has been normalized so that it is of unit length. In the vector-space model, the cosine similarity is the most commonly used method to compute the similarity between two objects Oi and Oj, sim(Oi,Oj), which is defined as cos(Oi,Oj) = Oi•Oj/(||Oi||•|||Oj||). The cosine formula can be simplified to cos(Oi,Oj) = Oi•Oj, when the record vectors are of unit length. This measure becomes one if the records are identical and zero if there is nothing in common between them. The idea of our approach is to transform the data representation for all objects in a multi-relational environment into a vector space model and find the similarity distance measures for all objects to cluster them. These objects then are grouped based on the similarity of their characteristics, taking into account all possible representations and the frequency of each representation for all objects.

Dynamic Aggregation of Relational Attributes (DARA): In relational database, records are stored separately in different tables and they are associated through the matching of primary and foreign keys. With a high degree of one-to-many association, a single record, O, stored in a main table is associated with a large volume of records stored in another table. In our algorithm called the Dynamic Aggregation of Relational Attributes (DARA), we convert the data representation from a relational model into a vector space model. Let O denotes a set of n records stored in the target table and let R denotes a set of m records (T_1, T_2, T_3, … , Tm) stored in the non-target table. Let Ri is in the subset of R, R_i R and is associated with a single record Oa stored in the target table, O_a O. Thus, the association of these records can be described as $O_a \rightarrow R_i$. Since a record can be characterized based on the bag of term/records that are associated with it, we use the vector space model to cluster these records, as described in the study of Salton *et al*. (1975). In vector space model, a record is represented as a vector or 'bag of terms', i.e., by the terms it contains and their frequency, regardless of their order. These terms are encoded based on the number of attributes combined, p and represent instances stored in the non-target table referred by a record stored in the target table (Alfred, 2008). The encoding process to transform relational datasets into data represented in a vector-space model has been implemented in DARA (Alfred and Kazakov, 2006a; 2006b). Given this data representation, we can use clustering techniques (Hofmann and Buhnmann, 1998; Hartigan, 1975) to cluster them, as a means of aggregating them. DARA algorithm simply assigns each record in the target table with the cluster number. Each cluster then can generate more information by looking at the most frequent patterns that describe each cluster.

RESULTS

In this experiment, we employ an algorithm, called DARA that converts the dataset representation in relational model into a space vector model and use a distanced-based method to group objects with multiple representations occurrence. With DARA algorithm, all representations of two objects are taken into consideration in measuring the similarity between these two objects. The DARA algorithm can also be seen as an aggregation function for multiple instances of an object and is coupled with the C4.5 classifier (J48 in WEKA) (Witten and Frank, 2000), as an induction algorithm that is run on the DARA's transformed data representation. We then evaluate the effectiveness of each data transformation with respect to C4.5. The C4.5 learning algorithm (Quinlan, 1993) is a state-of-the-art top-down method for inducing decision trees. All experiments with DARA and C4.5 were performed using a leave-one-out cross validation estimation with different values of p, where p denotes the number of attributes being concatenated. We chose well-known dataset, Mutagenesis (Srinivasan *et al*., 1995).

The mutagenesis data (Srinivasan *et al*., 1996) describes 188 molecules falling in two classes, mutagenic (active) and non-mutagenic (inactive); 125 of these molecules are mutagenic. The description consists of the atoms and bonds that make up the compound. Thus, a molecule is described by listing its atoms atom (AtomID, Element, Type, Charge) and the bonds bond (Atom1, Atom2, BondType) between atoms. In this experiment, we use three different sets of background knowledge: B1, B2 and B3:

B1: The atoms in the molecule are given, as well as the bonds between them; the type of each bond is given as well as the element and type of each atom. The table for B1 has the schema Molecule(ID, ATOM1, ATOM2, TYPE_ATOM1, TYPE_ATOM2, BOND_TYPE), where each molecule is described over several rows, listing all pairs of atoms with a bond and the type of each atom and the type of bond between them

B2: Continuous values about the charge of atoms are added to all data in B1

B3: Two continuous values describing each molecule are added to all data in B2. These values are the log of compound's octanol/water partition coefficient (logP) and energy of the compound's Lowest Unoccupied Molecular Orbital (ЄLUMO)

In B1, there are five attributes that describe an individual molecule, namely first atom, second atom, first element's type, second element's type and bondtype. There are typically several records for each molecule. We performed a leave-one-out cross validation estimation using the C4.5 classifier for p = 1, 2, 3, 4, 5 (p is the number of attributes combined) as we have a total of five attributes for dataset B1. Table 1 shows that the predictive accuracy of the decision tree learned.

In B2, two attributes are added into B1, which are the charges of both atoms. We performed a leave-one-out cross validation estimation using the C4.5 classifier for $p\in\{1,2,3,4,5,6,7\}$, as we now have a total of seven attributes for dataset B2. With additional two more attributes, we have a higher prediction accuracy of the decision tree when p = 5, compared to learning from B1 when p = 5, as shown in Table 1.

In B3, two more attributes are added to the existing dataset B2 and we now have the following row of attributes: [first atom, second atom, first element's type, second element's type, bondtype, first element's charge, second element's charge, log P, ЄLUMO]. Table 1 indicates that the prediction accuracy of a leave-one-out cross validation of C4.5 is the highest when p = 4 and 8.

Table 2 shows the DARA+C4.5 performance in the case of the mutagenesis dataset, using leave-one-out cross-validation and the J48 implementation of C4.5 (Witten and Frank, 2000).

Table 1: Predictive performance of C4.5 on mutagenesis datasets B1, B2 and B3 based on 10-fold cross-validation

	Number of features considered, p								
Datasets	1	2	3	4	5	6	7	8	9
B1	80.9	81.4	77.7	78.8	81.2	-	-	-	-
B2	79.5	80.0	81.2	80.3	82.8	81.8	79.5	-	-
B3	79.5	81.6	79.1	82.7	80.2	79.1	79.0	82.7	78.6

Table 2: Comparison on performance accuracies on mutagenesis datasets

Algorithms	B1 (%)	B2 (%)	B3 (%)
PROGOL (Srinivasan *et al*., 1996)	76	81	83
FOIL (Emde and Wettschereck, 1996)	83	75	83
TILDE	75	75	85
RDBC (Maron and Lozano-Perez, 1998; Wang and Zucker, 2000)	83	84	82
DARA (Hofmann and Buhnmann, 1998; Hartigan, 1975)	81	83	83

DISCUSSION

In B1, the predictive accuracy is the highest when p is 2 or 5. When p = 2, the attributes used for clustering are the following 3 compounds: [first atom, second atom], [first element's type, second element's type] and [bondtype]. When p = 5, the only attribute used is: [first atom, second atom, first element's type, second element's type, bondtype]. A test using the correlation-based feature selection (CFS in WEKA) function (Witten and Frank, 2000) provides a possible explanation of these results. We find that the two attributes, first element's type and second element's type, are highly correlated with the class membership, yet uncorrelated with each other. This means that an attribute combining these two would be relevant to the learning task and split the instance space in a suitable manner. The data contains this composite attribute when p = 2, 4 and 5, but not for the cases of p = 1 and 3.

In B2, when p = 5, we have two compound attributes, [first atom, second atom, first element's type, second element's type, bondtype] and [first element's charge, second element's charge]. Table 1 shows that drop in performance when p = 1 and 2. In contrast, we have higher prediction accuracy when p = 5. We have shown above that in the case of B1, the attributes first element's type and second element's type are highly correlated with the class membership. For B2, we have used the same technique to find that the first element's charge and the second element's charge are also highly correlated with the class membership, yet uncorrelated with each other. This explains the higher prediction accuracy for B2 and p = 5, as in this case 2 useful compound attributes are formed: [first element's type, second element's type] and [first element's charge, second element's charge].

In B3, when p = 4, we have the following compound attributes [first atom, second atom, first element's type, second element's type], [bondtype, first element's charge, second element's charge, logP] and, finally [ЄLUMO]. Each of the first two subsets of attributes contains a pair of attributes that are highly correlated with the class membership. Again, this can be used to explain the high prediction accuracy for a leave-one-out cross validation of C4.5 when p = 4 with dataset B3.

Based on the comparison shown in Table 2, the results show that for each of the other algorithms listed in Table 2, there is a dataset on which our algorithm performed better than the other relational data mining approaches. For instance, our approach outperforms RDBC when all available tables are used. Unlike

RDBC, our approach computes the distance between two different objects based on the representation of its instances (concatenated attributes). As a result, for each cluster, we can find the representations (by taking the representation with highest weight) that best describe the clusters and these representations can be used as an interpretable rules for clustering or classifying unseen objects with multiple instances.

CONCLUSION

This study presents an algorithm transforming biomedical datasets in a multi-relational setting into a vector space model that is suitable to clustering operations, as a means of aggregating or summarizing multiple instances. We carried out an experiment that clusters the objects in a multi-relational setting based on the patterns formed. The results show that varying the number of concatenated attributes p before clustering has an influence on the predictive accuracy of the decision tree learned by the C4.5 classifier. We have found that an increase in accuracy coincides with the cases of grouping together attributes that are highly correlated with the class membership. However, the prediction accuracy is degraded when the number of attributes concatenated is increased further. The results indicate that limiting the number of attributes may be desirable. At the same time, it is beneficial to combine attributes that are highly correlated with the class membership together. In this study, keeping the number of concatenated attributes n relatively small (e.g. $n \leq 5$), results in the best performance in terms of prediction accuracy as measured by leave-one-out cross-validation of the C4.5 decision tree.

Finally, the results show that data summarization performed by DARA, can be beneficial in summarizing biomedical datasets in a complex multi-relational environment, in which biomedical datasets are stored in a multi-level of one-to-many relationships and also in the case of datasets stored in more than one one-to-many relationships with non-target tables.

REFERENCES

Aha, D.W., D. Kibler and M.K. Albert, 1991. Instance-based learning algorithms. Machine Learn., 6: 37-66. DOI: 10.1023/A:1022689900470

Alfred, R. and D. Kazakov, 2006a. Data summarization approach to relational domain learning based on frequent pattern to support the development of decision making. Adv. Data Mining Appl., 4093: 889-898. DOI: 10.1007/11811305_97

Alfred, R. and D. Kazakov, 2006b. Pattern-based transformation approach to relational domain learning using Dynamic Agression for Relationa Attributes. University of York.

Alfred, R. and D. Kazakov, 2007. Aggregating Multiple Instances in Relational Database Using Semi-Supervised Genetic Algorithm-based Clustering Technique. York University.

Alfred, R., 2008. DARA: Data Summarisation with Feature Construction. Proceedings of the 2nd Asia International Conference on Modelling and Simulation, May, 13-15, IEEE Xplore Press, Kuala Lumpur, pp: 830-835, DOI: 10.1109/AMS.2008.131

Dietterich, T.G., R.H. Lathrop and T. Lozano-Perez, 1997. Solving the multiple instance problem with axis-parallel rectangles. Artificial Intell., 89: 31-71. DOI: 10.1016/S0004-3702(96)00034-3

Emde, W. and D. Wettschereck, 1996. Relational instance-based learning. Proceedings of the 13th International Conference on Machine Learning, (ML'96), Morgan Kaufmann, pp: 122-130.

Getoor, L., N. Friedman, D. Koller and A. Pfeffer, 2001. Learning Probabilistic relational models. In: Relational Data mining, Dzeroski, S. and N. Lavrac, (Eds.). Springer, Berlin, ISBN: 3540422897, pp: 307-333.

Hartigan, J.A., 1975. Clustering Algorithms. 1st Edn., Wiley, New York, ISBN: 047135645X, pp: 351.

Hofmann, T. and J.M. Buhnmann, 1998. Active data clustering. Psroceeding of the Advance in Neural Information Processing System, (ANIPS'98), MIT Press Cambridge, MA, USA., pp: 528-534.

Horvath, T., S. Wrobel and U. Bohnebeck, 2001. Relational instance-based learning with lists and terms. Machine Learn., 43: 53-80. DOI: 10.1023/A:1007668716498

Kirsten, M. and S. Wrobel, 1998. Relational distance-based clustering. Introductive Logic Programm., 1446: 261-270. DOI: 10.1007/BFb0027330.

Kirsten, M. and S. Wrobel, 2000. Extending k-means clustering to first-order representations. Introductive Logic Programm., 1866: 112-129. DOI: 10.1007/3-540-44960-4_7

Kramer, S., N. Lavrac and P. Flach, 2001. Propositionalization approaches to relational data mining. In: Relational Data mining, Džeroski, S. and N. Lavrac, (Eds.). Springer, Berlin, ISBN 3540422897, pp: 262-286.

Maron, O. and T. Lozano-Perez, 1998. A framework for multiple-instance learning. Adv. Neural Inform. Process. Syst., 10: 570-576.

Muggleton, S.H. and L. De Raedt, 1994. Inductive logic programming: Theory and methods. J. Logic Programm., 20: 629-679. DOI: 10.1016/0743-1066(94)90035-3

Propescul, A., L.H. Ungar, S. Lawrence and D.M. Pennock, 2002. Towards structural logistic regression: Combining relational and statistical learning. University of Pennsylvinia.

Quinlan, J.R.,1993. C4.5: Programs for Machine Learning. 1st Edn., Morgan Kaufmann, San Mateo, ISBN: 1558602380, pp: 302.

Salton, G., A. Wong and C.S. Yang, 1975. A vector space model for automatic indexing. Communications ACM, 18: 613-620. DOI:10.1145/361219.361220

Srinivasan, A., S. Muggleton and R.D. King, 1995. Comparing the use of background knowledge by Inductive Logic Programming systems. In Oxford University Computing Laboratory.

Srinivasan, A., S.H. Muggleton M.J.E. Sternberg and R.D. King, 1996. Theories for mutagenicity: A study in first-order and feature-based induction. Artificial Intell., 85: 277-299. DOI: 10.1016/0004-3702(95)00122-0

Wang, J. and J.D. Zucker, 2000. Solving the multiple-instance problem: A lazy learning approach. Proceedings of the 17th International Conference on Machine Learning, Morgan Kaufmann Publishers Inc. San Francisco, CA, USA., pp: 1119-1126.

Wettsschereck, D. and T.G. Dietterich, 1995. An experimental comparison of the nearest-neighbor and nearest-hyperrectangle algorithms. Machine Learn., 19: 5-27. DOI: 10.1023/ A:1022603022740

Witten, I.H. and E. Frank, 2000. Data Mining: Practical Machine Learning Tools and Techniques. 1st Edn., Morgan Kaufmann, San Francisco, ISBN: 1558605525, pp: 371.