# Top-Down Induction of Relational Model Trees in Multi-instance Learning

Annalisa Appice, Michelangelo Ceci, and Donato Malerba

Dipartimento di Informatica, Università degli Studi di Bari
via Orabona, 4 - 70126 Bari - Italy
{appice,ceci,malerba}@di.uniba.it

**Abstract.** (Multi-)relational regression consists of predicting continuous response of target objects called reference objects by taking into account interactions with other objects called task-relevant objects. In relational databases, reference objects and task-relevant objects are stored in distinct data relations. Interactions between objects are expressed by means of (many-to-one) foreign key constraints which may allow linking explanatory variables of a task-relevant object in several alternative ways to the response variable. By materializing multiple assignments in distinct attribute-value vectors, a reference object is represented as a bag of multiple instances, although there is only one response value for the entire bag. This works points out the same assumption of multi-instance learning that is a primary instance is responsible for the observed response value of a reference object. We propose a top-down induction multi-relational model tree system which navigates foreign key constraints according to a divide-and-conquer strategy, derives a representation of reference objects as bags of attribute-value vectors and then, for each bag, constructs a primary instance as main responsible of the response value. Coefficients of local hyperplane are estimated in an EM implementation of the stepwise least square regression. Experiments confirm the improved accuracy of our proposal with respect to traditional attribute-value and relational model tree learners.

## 1 Introduction

Regression has received significant attention in supervised learning, where training data consist of observations of an unknown continuous function $f$, and the learning task is to learn a general model $g$ that is close to $f$ on training data and can be subsequently used to reliably predict on new unlabeled observations. Although regression task is most popular with attribute-value learning, several extensions have already been investigated in multi-relational data mining [11], where data is expected to be spread in several database relations.

Handling relational data adds significant difficulties to the regression task since data stored in distinct database relations describe objects of different type. These objects play different roles, and it is necessary to distinguish between the *reference* (or target) *objects*, which are the main subject of analysis, and the *task-relevant objects* (or non-target objects), which are related to the former and can

F. Železný and N. Lavrač (Eds.): ILP 2008, LNAI 5194, pp. 24–41, 2008.

contribute to account for the variation. The response variable of a regression model is a continuous property of the reference objects, while the explanatory variables of the model can be associated to both reference and task-relevant objects. Foreign key constraints model "many-to-one" relationships according to which several task-relevant objects may be associated to the same reference object (non-determinacy) [15]. Since explanatory variables of task-relevant objects, namely non-determinate variables, can be linked to the response variable in several alternative ways, it is necessary to establish how the response value should be estimated due to the multiple instances that are possible for one reference object when taking into account the task-relevant objects. Deriving multi-instance data for a reference object boils down multi-relational regression to a generalization of the *multi-instance* learning.

Multi-instance learning was firstly defined in [6] to deal with predictive tasks where training data consist of bags of several instances and the response value is assigned to the entire bag. In the original proposal, all instances, which are represented by attribute-value vectors, are used for training. However, in many cases not all instances are responsible for the observed value, thus aggravating problems due to noisy and irrelevant observations. To overcome these limitations, several regression methods developed in multi-instance settings base their prediction on only one (or few) instance(s) of the bag. The validity of these methods has been empirically proved in several applications ranging from drug design [6] to invitro fertilization [19] and image analysis [16].

In this work, we follow the same approach to solve the multi-instance learning problem generated from multi-relational data, and we base the prediction on the primary instance (or prototype) selected for each reference object. In particular, we propose a novel multi-relational learner, called MIRT (*M*ulti *I*nstance *R*elation model *T*ree *I*nduction), which follows a multi-instance approach to learning regression models from multi-relational data. MIRT works under the common hypothesis that the underlying function $f$ is a linear model with Gaussian noise on the value of the response variable. To avoid the *a priori* definition of a global form for the regression function [13], the function $f$ is approximated by means of a tree-structured regression model, called *model tree*. This is built by recursively partitioning the set of reference objects according to a computationally efficient divide-and-conquer search strategy. An internal node can either perform a binary test on some explanatory variable or instantiate a new relation on the basis of some foreign key constraint, while a leaf is tagged with a multiple linear function (or hyperplane) $g$ built over a multi-instance representation of the reference objects at the leaf.

Our proposal permits to compute simultaneously both primary instances and regression coefficients together by means of Expectation Maximization (EM) algorithm which minimizes the least square error of a multiple linear function learned in a stepwise fashion [8]. In this way, the final relational model tree reveals local linear dependences between response variable and explanatory ones without suffering of the presence of noise and outliers. The tree can then be used to predict the unknown response of any new reference object (test), whose

primary instance is constructed by choosing the binding of the non-determinate variables involved in the local $g$ that minimizes the Euclidean distance from the training primary instances stored in the leaf.

The paper is organized as follows. In the next Section, we discuss background and motivations of this work. The learning problem is formally defined in Section 3. In Section 4, we present a novel method that enriches the top-down induction of a relational model tree with an EM iterative approach to compute primary instances and regression coefficients. Lastly, experimental results are reported in Section 5 and some conclusions are drawn.

## 2 Background and Motivation

Related research on regression in multi-relational data mining and multi-instance learning are reported below.

### 2.1 Background in Multi-relational Data Mining

Multi-relational data mining methods for regression can be classified into two alternatives: propositional and structural (or relational).

The propositional approach constructs features which capture relational properties of data and transforms original relational data into a single attribute-value representation. The non-determinacy issue is dealt with by deriving boolean features (e.g. there exists at least a molecule conformation that includes an atom with charge greater than 2.5) or aggregate features (the average charge of atoms involved in the conformations of the same molecule). The resulting representation can then be input to a wide range of robust and well-known conventional regression methods which operate on an attribute-value single instance representation [12].

The structural approach provides functionalities to navigate relational structures in its original format and to generate potentially new forms of evidence not readily available in a flattened single table representation. The whole hypothesis space is directly explored by the mining method. Structural regression methods [14,2,1,22,9] are generally obtained by upgrading propositional learners, e.g. regression trees and model trees, to the multi-relational setting. Although several of these methods assume that the function underlying data is a (local) hyperplane, only few of them [1,22] allow regression functions which include non-determinate explanatory variables. These methods use the several instances of a reference object in training and employ aggregate functions after or before learning coefficients of local hyperplanes. In [1], multiple instances are dealt as separate instances and coefficients of hyperplane are computed by solving least square regression in the derived single instance learning. Multiple predictions derived for the same reference object are aggregated by resorting to the average function. In [22], user-defined aggregate functions allow to aggregate a non-determinate variable before adding the variable to the linear model, thus avoiding the problem of multiple predictions. Anyway the use of aggregates

may suffer from problems of information loss and statistical significance when in presence of noise or outliers.

### 2.2 Background in Multi-instance Learning

Although multi-instance learning was initially defined for classification [6], some regression methods have been developed for the multi-instance setting.

A seminal work is in [18], where Ray and Page have assumed that the function $f$ underlying multi-instance data is a linear model and there is one instance for each bag that is the main responsible of the real-valued response value. They have proposed an EM-based iterative method to determine primary instances and regression coefficients of the hyperplane that minimizes the least square error. Regression coefficient are computed for all explanatory variables. The hyperplane can then be used to predict the unknown response of a new object represented by a bag of multiple instances. The problem of determining the test primary instance is naively solved by selecting the first instance stored in the bag. Zhang and GoldmanIn [23] have combined the EM algorithm with a diverse density based algorithm. A two-step gradient ascent search of the standard diverse density algorithm is used to select the primary instances which maximize diverse density value. Dooly et al. [7] have proposed a multiple-instance variant of unweighted k-NN in which the distance between bags is defined as the minimum Euclidean distance between points in the two bags. The regression method investigated in [5] bypasses the problem of choosing a primary instance by computing a nonlinear weighted average of all response values. However, this is justified by the specific task of predicting the molecule activity. A gradient-based optimization procedure is used to determine the best linear model.

The multi-instance approach to relational regression task has also been considered in multi-relational data mining. Srinivasan and Camacho [20] have firstly recognized the multi-instance nature of a relational regression task when learning relational clauses. However no solution was proposed for the issue of binding those non-determinate variables which are the responsible for a response value. Several bindings are either treated as data points for regression analysis or are grouped in a single aggregated value. An important contribution to interpreting multi-relational data mining problems as multi-instance learning problems comes from Blockeel et al. [3], who have upgraded decision tree learning to the multi-instance framework in ILP.

## 3 Problem Statement: Relational vs. Multi-instance

The relational regression task can be formally defined as follows:

*Given*:

1. a set $S$ of reference objects, that is, the target objects of the analysis;
2. some sets $R_i$ of task-relevant objects;

3. a set $I$ of interactions between reference objects and/or task-relevant objects;
4. a response value $y \in \mathbb{R}$ which tags each reference object in $S$ and assumes value according to an unknown function $f : \langle S, \bigcup R_i, I \rangle \rightarrow \mathbb{R}$.

*Find* a function $g$ that is hopefully close to $f$ on the domain $\langle S, \bigcup R_i, I \rangle$.

The set $S$ is stored in one data relation (namely target table) of a relational database $D$, i.e. one tuple for each reference object. Similarly, each set $R_i$ is stored in a distinct data relation of $D$ (one table for each object type). The interactions $I$ are expressed by the foreign key constraints ($FK$) which state how tuples in one data relation relate to tuples in another relation. Foreign key constraints $FK$ allow navigating $D$ and retrieving all the task-relevant objects which are *related* to a reference object and thus potentially relevant to predict the value of $Y$. The definition of the task-relevant objects which are *related* to a reference object according to a *foreign key path* is formally provided in the following.

**Definition 1.** *A task-relevant object $tro \in R_i$ is related to a reference object $ro \in S$ if and only if:*

1. *there exists a foreign key constraint $fk \in FK$ from $R_i$ to $S$ (or vice-versa) such that the foreign key of $tro$ assumes the same value of the primary key of $ro$ (or vice-versa), or*
2. *there exists a task-relevant object $newTro \in R_j$ such that $newTro$ is related to $ro$ and there exists a foreign key constraint $fk$ from $R_j$ to $R_i$ (or vice-versa) such that the foreign key of $newTro$ assumes the same value of the primary key of $tro$ (or vice-versa).*

*The sequence of foreign key constraints $fkp = \langle fk_1, \ldots, fk_n \rangle$ according to a task-relevant object is related to a reference object is called foreign key path.*

A foreign key path provides the schema of the attribute-value vectors which can be constructed by performing the natural join between data relations involved in the foreign key path and projecting over the explanatory variables. Grouping the attribute-value vectors referring to the same reference object allows to represent a reference object as a bag of multiple attribute-value vectors. In this case, the difference with the original formulation of a multi-instance task is that a relational bag may include attribute-value vectors with different attribute schema (one schema for each foreign key path). Independently from this difference, the multi-instance form of relational data poses the same difficulties due to noise and presence of irrelevant instances as traditional multi-instance data. This motivates our focus on the problem of selecting the main responsible of the response values. This attempt corresponds to reformulate the relational regression goal as follows:

*Find (i) the primary instance of each reference object by choosing the best binding for non-determinate variables of possibly related task-relevant object, and (ii) a regression function $g$ that is hopefully close to $f$ on the retrieved primary instances.*

## 4 Multi-instance Induction of Relational Model Trees

MIRT induces a multi-relational model tree by recursively partitioning the reference objects stored in the target table of a relational database and associating different hyperplanes to the separate partitions. Partitioning takes into account the relational arrangement of task-relevant objects stored in the same relational database. Each hyperplane is a linear combination of a subset of the explanatory variables belonging to several data relations in the splitting tests along the unique tree path connecting the root to the leaf.

The approach adopted to estimate the regression coefficients is where MIRT differs from the state-of-art relational model tree learners. MIRT neither resorts to aggregate nor considers multiple binding values as separate instances, but it minimizes the least square error over the primary instances constructed from the training attribute-value vectors falling in the leaf. Under the restriction that only the attribute-value vectors which satisfy the intensional description of the partition at hand are constructed, the primary instances are obtained in an EM-based formulation of the stepwise regression that i) chooses at each step the best explanatory variable to be introduced in the hyperplane, ii) identifies the best binding for this variable among the possible ones in the bag iii) uses these best bindings to compute an estimate of the regression coefficient of this variable in the hyperplane. The primary instances constructed in the training are stored in each leaf in order to provide a baseline to construct the primary instance of an unknown reference object and then to predict its unknown response value.

Details of the relational tree induction, the EM based implementation of multi-instance stepwise regression and the prediction of unknown reference objects are reported in the next sub Sections.

### 4.1 Relational Tree Induction

The construction of the tree proceeds top-down. It starts with a root node $t_0$ with is associated with the entire set of training reference objects and recursively proceeds by choosing from either:

- growing the tree by performing a splitting test on the current node $t$ and introducing the nodes $t_L$ (left child of $t$) and $t_R$ (right child of $t$) or
- stopping the tree's growth at the current node $t$ and then associating an hyperplane at $t$.

At each node, the splitting test is chosen by minimizing the average standard deviation of response value [4]. Coherently with the semantics of a relational tree defined in [2], a variable that is introduced in the splitting test (i.e. this variable does not occur in the higher splitting tests of the tree) must not occur in the negation of the splitting test. This restriction is required to guarantee the *mutually exclusivity* when partitioning reference objects according to the test conditions which may involve several data relations.

A splitting test may either be a *foreign key* test or an *attribute* test. A foreign key test is a binary test to partition reference objects according to the existence of

a non empty relationship between data relations involved in one or more foreign key constraints. This corresponds to perform natural joins between relations involved in the foreign key path associated to the unique tree path connecting the root to the node and relations introduced with the foreign key test. Due to the complexity of computing natural joins, MIRT imposes that a foreign key between two data relations can be introduced at most once in this unique path. The foreign key constraints are selected from the relational database under the restriction that the resulting foreign key path must satisfy the linkedness. This corresponds to consider only foreign key constraints, where either the foreign key or the primary key belong to one data relation already involved in a test along the corresponding tree path from the root to the node.

*Example 1.* An example of splitting test which employs a foreign key condition (from atom to molecule) to partition the entire set of molecules:

molecule(I, L, M)
| − −(yes)[molecule(I, L, **M**),atom(A,**M**,C) ]
| | − − ...
| − −(no) [molecule(I,L,M), **not(molecule(I,L,M),atom(A,M,C))**] ...

An attribute test is a test involving a boolean condition on an attribute $X$. $X$ is neither a primary key nor a foreign key and $X$ belongs to one of the data relations added to the tree by means of a foreign key condition. if $X$ is continuous, the binary test is in the form $X \leq \alpha$ with $\alpha$ one of the points found on range of $X$ in partition at hand, while if $X$ is a discrete variable, the binary test is in the form $X \in \{\alpha_1, \ldots, \alpha_s\}$, where $\{\alpha_1, \ldots, \alpha_s\} \subset S_X$ and $S_X$ is the set of distinct values of $X$ in the partition in hand. MIRT starts with an empty set $Left_X = \oslash$ and a full set $Right_X = S_X$. It moves one element from $Right_X$ to $Left_X$, such that the move results in a better split.

*Example 2.* An example of splitting test which employs an attribute condition (Charge) to partition the set of molecules which have at least one atom related according to the foreign key constraint from atom to molecule:

molecule(I, L, M)
| − −(yes)[molecule(I, L, M),atom(A,M,C) ]
| | − − (yes)[molecule(I,L,M),atom(A,M,C), **C≤2.3**]
| | | − − ...
| | − − (no)[molecule(I,L,M),atom(A,M,C),**not(**
**molecule(I,L,M),atom(A,M,C), C≤2.3)** ]
| | | − − ...
| − −(no) [molecule(I,L,M), not(molecule(I,L, M),atom(A,M,C))] ...

In addition to foreign key test and attribute test, MIRT can perform a test which combines one or more foreign key conditions with one attribute condition.

*Example 3.* An example of a splitting test which employs simultaneously a foreign key condition (from atom to molecule) and an attribute condition (Charge) to partition the entire set of molecules.

molecule(I, L, M)
| − −(yes)[molecule(I,L,**M**),atom(A,**M**,C), **C≤2.3**]
| | − − ...
| − −(no) [molecule(I,L,M),**not(molecule(I,L,M),atom(A,M,C),C≤2.3)**]
| | − − ...

The tree construction is stopped when either the number of reference objects in a node is less than a minimum value or the coefficient of determination is greater than a threshold. The coefficient of determination estimates the strength of the relationship between the average response values on partition at hand and the average response values on the entire training set.

### 4.2 Multi-instance Stepwise Regression

A local hyperplane $y = \mathbf{bx}$ at a leaf node is learned to be close to $f$ on the primary instances constructed for the training reference objects falling in the partition at hand. Primary instances are constructed by fixing one binding among the possible several ones of the (non-determinate) variables involved in the hyperplane and $\mathbf{b}$ is determined by minimizing the least square error (L) on the primary instances. The hyperplane construction starts from representing each training reference object $i$ falling in the partition at hand as a bag $B_i$ of $m_i$ attribute-value vectors (instances). Each instance $j$ of $B_i$ is described by the real valued vector $\mathbf{x_{ij}}$ which includes values of the continuous explanatory variables $X_i$ from the data relations in the foreign key conditions along the tree path from the root to the leaf under analysis. The multiple instances of $B_i$ are only those instances which satisfy the conjunction of test conditions for the reference object $i$. An example of the construction of the bag of attribute-value vectors which describe a reference object falling in a leaf is reported in Example 4.

*Example 4.* Let us consider:
1. a database schema which includes the data relations
   Molecule(MolId, LogP and Muta)
   Atom(Atomid, MolId, Charge)
   Bond(BondId, AtomId1, AtomId2, Type)
2. an instance of this database which collects the tuples:
   molecule(m1, 12, 5.1). molecule(m2, 2, 10.1).
   atom(a1, m1, 2.3). atom(a2, m1, 2.5). atom(a4, m1, -0.5). atom(a5, m2, 5.1).
   bond(b1, a1, a2, 5). bond(b2, a1, a2, 2). bond(b3, a1, a3, 1).
   bond(b4, a5, a6, 2). bond(b5, a6, a5, 1).
3. a relational tree that partitions molecules according to atoms and bond is the following:
   1. molecule(I, L, M)
   2. | − −(yes)[molecule(I, L, M),atom(A,M,C) ]
   3. | | − − (yes)[molecule(I,L,M),atom(A,M,C), C≤2.3]
   4. | |−− (yes)[molecule(I,L,M),atom(A,M,C), C≤2.3, bond(B,A,A2,T]
   5. | | − − (no) ...

6. | | − − (no)...
7. | − −(no) ...

$m1$ satisfies the conjunction of binary conditions along the path from the node 1 to the node 4, that is, at least one atom with a charge less or equal to 2.3 belongs to $m1$ and this atom is involved in at least one bond. The intensional description provided by the tree at this node can be expressed as the query:

*m1(L, C, T) :- molecule(m1,L,Y), atom(M,A,C), C≤2.3, bond(B,A,K,T)*

According to the query reported above $m1$ is mapped into the bag attribute-value vectors ⟨12, 2.3, 5⟩ ⟨12, 2.3, 2⟩ ⟨12, 2.3, 1⟩ and the entire bag is assigned to the response value 5.1.

After constructing the multiple attribute-value vectors which represent each reference object falling in the leaf, primary instances and regression coefficients are computed within an EM algorithm that aims at minimizing the least square error of the hyperplane on the training primary instances constructed at the leaf. The hyperplane is built stepwise by sequencing straight line regressions and removing the linear effect of the introduced variables each time a new explanatory variable (regression term) is added to the model. The selection of this *best* term is based on the strength of the resulting least square error on the chosen primary values for $X_i$. Basics of stepwise construction are provided in Example 5.

*Example 5.* Suppose we are interested in analyzing a response variable $Y$ in a region $R$ of a feature space described by two continuous explanatory variables $X_1$ and $X_2$. In the stepwise construction of a regression model, the initial regression model is approximated by regressing on $X_1$ for the whole region $R$, that is $\hat{Y} = \hat{a}_0 + \hat{b}_0 X_1$. As explained in [8], the correct procedure to follow in order to introduce the effect of another variable in the partially constructed regression model is to eliminate the effect of $X_1$. In practice, we have to compute the regression model for the whole region $R$, that is, $\hat{X_2} = \hat{a}_{20} + \hat{b}_{21} X_1$ and to compute the residuals $X_2' = X_2 - \hat{X_2}$ and $Y' = Y - \hat{Y} = Y - (\hat{a}_0 + \hat{b}_0 X_1)$.
Finally by regressing $Y'$ on $X_2'$ alone $Y' = \widehat{\beta_{0_3}} + \widehat{\beta_{1_3}} X_2'$ and substituting the equations of $X_2'$ and $Y'$ in the last equation we obtain:

$Y - (\widehat{\beta_{0_1}} + \widehat{\beta_{1_1}} X_1) = \widehat{\beta_{0_3}} + \widehat{\beta_{1_3}}(X_2 - (\widehat{\beta_{0_2}} + \widehat{\beta_{1_2}} X_1))$.
that is, $Y = (\widehat{\beta_{0_3}} + \widehat{\beta_{1_1}} - \widehat{\beta_{0_2}\beta_{1_3}}) + (\widehat{\beta_{1_1}} - \widehat{\beta_{1_2}}\widehat{\beta_{1_3}})X_1 + \widehat{\beta_{1_3}} X_2$.

For each bag, the primary value corresponding to the variable to be added to the hyperplane is chosen within the EM implementation described in Algorithm 1. Each time a new variable is added to the hyperplane, primary values of this variable are definitely assigned to the values ($I$) chosen in the E step. The algorithm starts with an initial random guess ($IR$) at the hypothesis which is iteratively refined. Each iteration consists in two main steps. In the E step, a binding of $X_i$ is selected from each bag to obtain an hypothesis with least square error ($L$-error) with respect to the current best guess at the correct hypothesis. In the M step, the current guess of the hypothesis is refined by using linear regression to construct a new regression model from the instances provided at the previous step. The new hyperplane is constructed by determining the coefficient of the

**Algorithm 1.** EM based selection of regression coefficients and primary values

1: **Input:** (1) the vector $B$ of $n$ bags $b_i$, where the bag $b_i$ includes the values $x_{i1}, \ldots x_{im}$ for the continuous explanatory variable $X$, (2) the list $L$ of the explanatory variables already included in the hyperplane, (3) the vector $P'$ of the residuals of the primary instances constructed from the explanatory variables already included in the hyperplane, (4) a vector $Y'$ of the residuals of the response values.
2: **Output:** (1) the regression coefficient of a straight-line regression between Y' and residual of X', (2) the vector $I$ of $n$ primary values of $X$, one for each bag in $B$
3: **function** EM($in : B, L, P', Y'; out : b, I$);
4: globalL:=MAX;
5: **for** r=1, . . . ,R **do**
6: assign randomly the primary instances to IR and determine residuals IR' by removing the effect of variables in $L$;
7: determine the regression coefficient b over IR';
8: bestL=MAX; currentL=0; found=true;
9: **while** found **do**
10: IC:=$\oslash$ ; currentL:=0;
11: **for** each $b_i$ in B **do**
12: Let $x_{ij}$ be the instance value in $b_i$ that minimizes $L(y'_i, x'_{ij}, b)$}; /* $x'_{ij}$ is the residual of $x_{ij}$ */
13: $IC = IC \cup x_{ij}$; currentL:=currentL+ $L(y'_i, x'_{ij}, b')$;
14: **end for**
15: **if** currentL$\geq$bestL **then**
16: found:=false;
17: **else**
18: bestL:=currentL; bR:=b; IR:=IC;
19: perform linear regression over IC to obtain b
20: **end if**
21: **end while**
22: **if** bestL<globalL **then**
23: globalL:=bestL; b:=bR; I:=IR
24: **end if**
25: **end for**
26: **return** $\langle$b,I$\rangle$

straight-line regression between the residual of $Y$ and the residual of $X_i$. Both residuals are computed in a stepwise fashion by iteratively removing the effect of the already performed steps of regression. The coefficients of the (sequence of) straight-line regressions (e.g. $\hat{a}_{20}$ and $\hat{b}_{21}$ in Example 5) to determine the residual values involved in an EM step are computed over the primary instances which have already been constructed within the stepwise construction of the hyperplane. EM steps are repeated until the algorithm converges. Due to the fact that the result of any EM run may be influenced by the initial random hypothesis, it is run several times on the same data collection using random restarts. In Algorithm 1, $R$ is the number of random restarts to be used.

After selecting the variable to be added to the hyperplane, the contribution of this term is evaluated according to the F-test and eventually dropped whenever

it is not statistically significant. In this last case, the addition of any other candidate cannot be statistically significant, hence the hyperplane construction can be stopped. In this way, MIRT integrates a mechanism of variable subselection as a part of the hyperplane construction, thus solving possible problems of collinearity [8]. If no variable is added to the hyperplane, the prediction at the leaf is simply performed by means of the mean of the response values of the reference objects falling in the leaf. Primary instances constructed within the EM based stepwise construction of the hyperplane are stored at the leaves and are subsequently used to determine the primary instance when an unknown reference object has to be predicted.

An example of the stepwise construction of an hyperplane according to Algorithm 1 is provided in Example 6

*Example 6.* Let us consider the molecules m1, m2 and m3 described according to the atoms a1, a2, a3, a4, a5, a6 and a7. Each molecule consists of a bag of attribute-value vectors (*Logp*, $Charge$) and a response value ($Muta$).

| InstanceId | MolID | Logp | Muta | Atomid | Charge |
|---|---|---|---|---|---|
| 1 | m1 | 2 | 7.5 | a1 | 5.1 |
| 2 | m1 | 2 | 7.5 | a2 | 30 |
| 3 | m1 | 2 | 7.5 | a3 | 5 |
| 4 | m2 | 5 | 12.5 | a4 | 11 |
| 5 | m2 | 5 | 12.5 | a5 | 11.5 |
| 6 | m3 | 3 | 9.8 | a6 | 7.1 |
| 7 | m3 | 3 | 9.8 | a7 | 7 |

We use the stepwise procedure to estimate the regression coefficients of an hyperplane to predict $Muta$ that includes $LogP$ and $Charge$. For simplicity, we consider R=1 and no F-test is performed when adding a new variable to the hyperplane.

An initial hyperplane is approximated by choosing to regress $Muta$ on either $LogP$ or $Charge$. At this aim, we compute the regression model for $LogP$ by randomly selecting primary values of $LogP$ from each bag and iterating in order to minimize the least square error (see Algorithm 1). Since $LogP$ assumes a single value on each bag, a single iteration is performed and it computes:

$$Muta = 4.52 + 1.62LogP \text{ with } I = [2, 5, 3] \text{ and } globalL = 0.25 \quad (1)$$

Similarly, we compute coefficients of a regression model for the multi-instance value of $Charge$. Let us consider the case that a random choice returns [5.1,11,7] as primary values for $Charge$. We use these primary values to determine:

$$Muta = 3.61 + 0.82Charge \quad (2)$$

Equation 2 minimizes the least square error on the charge values IC=[5,11.5, 7.1] (with currentL=0.196). By using this new set of primary values, we compute:

$$Muta = 3.65 + 0.81Charge \quad (3)$$

that minimizes the least square error with the charge values IC=[5,11.5, 7.1] (with currentL=0.195). This new set of primary values for $Charge$ will lead to stop EM iteration and return:

$$Muta = 3.65 + 0.81Charge \text{ with } I = [5, 11.5, 7.1] \text{ and } globalL = 0.19. \quad (4)$$

The initial hyperplane is then approximated by regressing $Muta$ on $Charge$ according to Equation 4. The residual attribute $Muta'$ is computed as follows:

$$Muta' = Muta - (3.65 + 0.81Charge)\ Muta' = [-0.23, -0.12, 0.35] \quad (5)$$

Finally, we introduce the effect of $LogP$. Let us consider the case the random choice return [2,5,3] as primary values for $LogP$. We firstly compute the residuals $LogP'$ by using the primary values of $Charge$ in Equation 4, that is:

$$LogP' = LogP - (-0.52 + 0.5Charge) \quad (6)$$

and then we compute:

$$Muta' = -10.99LogP' \quad (7)$$

which is proved to be the best one according to Algorithm 1. In this way we complete the construction of the following hyperplane:

$$Muta = (3.65 + 0.81Charge) - 10.99(LogP - (-0.52 + 0.5Charge)) \quad (8)$$

by constructing as primary instances of m1, m2 and m3, the attribute-value vectors $\langle m1, 2, 5\rangle$, $\langle m2, 5, 11.5\rangle$ and $\langle m3, 3, 7.1\rangle$, respectively.

### 4.3 Predicting Unknown Reference Objects

Let $\tau$ be the relational model tree induced from relational data stored in $D$. Let $H$ be the schema of $D$ and $ro$ be a test reference object that is stored in a new instance ($T$) of the same database schema $H$. $T$ contains the task-relevant objects which interact with $ro$ according to the foreign key constraints defined in $H$. The response value of $ro$ is unknown in $T$.

Starting from the root node of $\tau$, $ro$ is recursively passed down to the left (or right) child according to the fact that the splitting test on left (or right) edge is satisfied. When a leaf node is reached, MIRT constructs instances which describe $ro$ in $T$ according to the task-relevant objects which are related to $ro$ in the partition at hand. The structure (attribute vector) of these instances includes only the explanatory variables $(X_1, X_2, \ldots, X_d)$ which are involved in the hyperplane tagging the leaf. The primary instance of $ro$ is then constructed by fixing, for each variable, one binding over the bag and then by minimizing the distance from the training primary instances stored at the leaf.
Formally speaking, given:

1. the leaf $t$ such that $ro$ reaches $t$;
2. the hyperplane $y = g(\mathbf{X})$ which tags $t$ such that the attribute vector $\mathbf{X}$ is spanned by $d$ continuous variables $X_1, \ldots, X_d$;

3. the set $P$ of the training primary instances defined on $\mathbf{X}$ and stored in $t$;
4. the bag $B_{ro}$ of the database instances defined on $\mathbf{X}$ and constructed over $T$ to represent the reference object $ro$ falling in $t$;
5. the set $\psi_i$ ($\chi_i$), that is, the range of $X_i$ over $P$ ($R$).

MIRT constructs the primary instance $o$ of $ro$ by assigning each $X_i$ with one value over $\chi_i$ in order to minimize the distance from training primary instances stored in $P$, that is:

$$o = \min_{o \in \chi_1 \times \chi_2 \ldots \times \chi_d} \min_{t \in T} distance(p, t). \tag{9}$$

The distance between $p$ and $t$ is computed on the basis of the Euclidean distance measure, that is:

$$distance(p, t) = \sqrt{\sum_{i=1,\ldots,d} (p[X_i]) - (t[X_i])^2}. \tag{10}$$

Adopting the classical Euclidean distance, as in Equation 10, brings the problem of combining variables whose range may differ in several orders of magnitude. To overcome this problem, each value of $X_i$ is scaled within the range $[0, 1]$: the lowest (highest) value is assigned the real value 0 (1). The scaled values of $p[X_i]$ and $t[X_i]$ are obtained as follows:

$$p[X_i]_{scaled} = \frac{p[X_i] - min_j}{max_i - min_i} \quad \text{and} \quad t[X_i]_{scaled} = \frac{t[X_i] - min_j}{max_i - min_i}, \tag{11}$$

where $min_i = \min_{x_i \in \chi_i \cup \psi_i} x_i$ and $max_i = \max_{x_i \in \chi_i \cup \psi_i} x_i$. Once the primary instance $o$ is constructed, it is used to predict the unknown response value by assigning the explanatory variables in the hyperplane at $t$ to the corresponding values in $o$.

## 5 Experiments

MIRT is implemented in a Multi-Relational Data Mining system tightly coupled with a relational database (Oracle $10g$) and it is empirically evaluated on biological and geographical relational databases. Biological databases represent a benchmark application domain in multi-instance learning.

*Experimental Setting.* Each dataset is analyzed by means of a 10-fold cross-validation. Ten databases are created so that MIRT can be trained on nine databases and tested on the hold-out database. The system performance is evaluated on the basis of the average mean square error (MSE), that is:

$$MSE = \frac{1}{k} \sum_{i=1}^{10} \sqrt{\frac{1}{\#S_{D_i}} \sum_{j=1}^{\#S_{D_i}} (y_j - \widehat{y}_j(D/D_i))^2} \tag{12}$$

where $D = \{D_1, \ldots, D_k\}$ is a cross-validation partition, $D_i$ is a set of indices of testing databases, $k$ is the number of folds (i.e., 10), $\#S_{D_i}$ is the number of reference objects stored in $D_i$ and $\widehat{y}_j(D/D_i)$ is the value predicted for the $j$-th testing reference object by the model tree induced on $D/D_i$.

The thresholds for the stopping criteria are fixed as follows: the minimum number of reference objects falling in an internal node must be greater than the square root of the number of reference objects in the entire database, and the coefficient of determination in an internal node must be below 0.8. R is set to 5, while the maximum number of foreign key constraints to be added with a foreign key test is set to 2.

MIRT is compared with Mr-SMOTI [1] which induces a relational model tree that interleaves splitting tests and regression steps. At regression steps, Mr-SMOTI estimates the regression coefficients of straight-line regressions by assuming multiple-bindings of a non-determinate variable as single instances of least square regression. On one database, i.e. Mutagenesis, we compare MIRT with RE-MAUVE and TILDE-RT. RE-MAUVE is a relational model tree learner which resorts to aggregates to deal with non-determinate variables. TILDE-RT is a relational regression tree learner which associates a constant to each leaf. Finally, we compare MIRT with the propositional model tree learners SMOTI and M5'. In this last case, multi-relational data are transformed into a single table format. Two different transformations are considered. The former (P1) creates a single table by computing join operations for all possible foreign key paths rooted in the target table. This transformation may create multiple tuples for the same reference object. The latter transformation (P2) differs from the previous one because it does not generate multiple tuples for the same reference object. It is obtained by computing aggregates (i.e. the average for continuous values and the mode for discrete values) of-non-determinate variables. In the case of P1 we compute the MSE with respect to the average of the multiple response values output for the same test reference object as final prediction (MSE-G). We also compute the MSE by considering the response value as prediction of each single instance (MSE-S). Differently, in the case of P2, a single response is directly output for each test reference object. For the pairwise comparison of systems, the non-parametric Wilcoxon two-sample paired signed rank test [17] is used.

*Data Description.* MIRT is tested on four real databases, that is, Mutagenesis, Biodegradability, North West England (NWE) and Munich. Mutagenesis [21] and Biodegradability [10] are molecular databases used as a benchmark for several ILP systems. Mutagenesis is evaluated in three different settings. B0 consists of those data obtained with the molecular modeling package QUANTA. For each compound it obtains the atoms, bonds, bond types, atom types, and partial charges on atoms. B1 consists of definitions in B0 plus indicators ind1, and inda in molecule table. B2 consists in B1 plus variables (attributes) logp, and lumo. Biodegradability is evaluated in four settings. B0 consists of those data derived with SMILES without any global feature on molecule. B1 adds the numerical attributes mWeight and logP. B2 extends B0 by adding the indicator on molecular activity, while B3 includes all global features describing the molecules.

Details are provided in [1]. NWE is a collection of geo-referenced census data provided by the United Kingdom (UK) 1998 census. NWE census data includes values of mortality rate (response variable) and deprivation indexes geo-referenced at the level of 212 wards. Data also include 1045 rails, 2763 roads, 374 urban areas and 1040 waters crossing wards for a total of 5434 tuples. NWE dataset is provided in the European project SPIN!(http://www.ais.fraunhofer.de/KD/SPIN/project.html). Munich (http://www.di.uniba. it/%7Ececi/micFiles/munich db. tar.gz) describes rent-price (response variable) of 2179 flats geo-referenced within the Munich subquarters. The Munich metropolitan area is divided into 3 areal zones, each decomposed into 64 districts, for a total of 446 quarters for a total of 6808 tuples. This data was collected in 1998 to develop the Munich rental guide in 1999.

*Results.* The average MSE of the multi-relational systems is reported in Table 1. For Mutagensis (B1 and B2), we report MSE of RE-MAUVE and TILDE-RT taken from [22][1][2]. The comparison between MIRT and Mr-SMOTI (as well as Re-MAUVE and TILDE-RT for Mutagenesis) confirms our intuition that the accuracy of a relational model tree is generally improved when regression coefficients of non-determinate variables are estimated according to principles of multi-instance learning. The only database where Mr-SMOTI outperforms MIRT is Mutagenesis B2, in which mutagenecity of molecule strongly depends on the numeric properties of molecules (i.e., lumo and logP) which have single values for each molecule. This result is a confirmation that the stepwise construction of a model tree by interleaving split nodes and regression nodes outperforms the classical construction of a model tree by firstly partitioning data set and then locally deriving the hyperplanes to predict reference objects at each node [1]. Anyway, the advantages of a tree structure with split and regression nodes may be decreased by the presence of outliers values over non-determinate variables. This consideration suggests a future direction of the research described in this work, that is, employing the principles of multi-instance learning in the stepwise induction of relational model trees with split and regression nodes. The results on Mutagenesis show that MIRT also outperforms RE-MAUVE and TILDE-RT.

The results of the comparison between MIRT, SMOTI and M5' are reported in Table 2. The comparison is generally in favor of MIRT. The only statistically significant tests ($p \leq 0.05$) where a propositional learner (SMOTI -P2) outperforms MIRT concern Biodegradability (B2-B3). In general, the comparison of accuracy confirms not only the advantages of the structural approach over the

[1] The MSE of Mr-SMOTI on Mutagenesis significantly differs from the values reported in [22]. Differences may depend from a different tuning of parameters. In this work, we run Mr-SMOTI by allowing the possibility of learning foreign key tests introducing two foreign keys simultaneously (the default is 1), learning a foreign key test and an attribute test simultaneously in the same test (by default this possibility is disabled), filtering splitting tests which select less than 5 molecules on the left and right side of the tree.

[2] The MSE values reported in [22] are without the square radix.

**Table 1.** MIRT vs Mr-SMOTI, Re-Mauve and TILDE-RT: MSE of the model trees induced on the 10-fold CV of databases. Best MSE are in italics. "-" ("+") means that Mr-SMOTI performs worse (better) than MIRT in a Wilcoxon Test. "–" ("++") denotes the statistically significant values ($p \leq 0.05$).

| DB | | MIRT | Mr-SMOTI | Re-MAUVE no agg | Re-MAUVE agg | TILDE-RT | TILDE-RT agg |
|---|---|---|---|---|---|---|---|
| Mutgenesis | B0 | *1.4* | 1.67 – | | | | |
| | B1 | *1.11* | 1.28 - | 1.40 | | 1.4 | |
| | B2 | 1.12 | *0.95*+ | 1.20 | 1.20 | 1.24 | 1.36 |
| Biodegradability | B0 | *1.32* | 1.37 - | | | | |
| | B1 | *1.206* | 1.25 - | | | | |
| | B2 | *0.136* | 0.49 – | | | | |
| | B3 | *0.142* | 0.41 – | | | | |
| NWE | | *0.0024* | 0.0026 = | | | | |
| Munich | | *4.66* | 4.78 – | | | | |

**Table 2.** MIRT vs SMOTI and M5' (P1 and P2): MSE of the model trees induced on the 10-fold CV of databases. For SMOTI (P1) not all values are available, since the system returns error of memory. Best MSE are in italics. "-" ("+") means that SMOTI/M5' performs worse (better) than MIRT in a Wilcoxon Test. "–" ("++") denotes the statistically significant values ($p \leq 0.05$).

| DB | | MIRT | SMOTI (P1) | | M5(P1) | | SMOTI (P2) | M5(P2) |
|---|---|---|---|---|---|---|---|---|
| | | MSE | MSE-G | MSE-S | MSE-G | MSE-S | MSE | MSE |
| Mutgenesis | B0 | *1.4* | 2.44 – | 3.92 – | 1.68 - | 2.48 – | 2.57 - | 1.55 - |
| | B1 | *1.11* | 1.62 – | 1.65 – | 1.16 - | 2.10 – | 1.38 - | 1.13 = |
| | B2 | 1.12 | 2.19 - | 2.15 – | 1.05 - | 1.05 - | 1.009 + | *1.007* + |
| Biodegradability | B0 | *1.32* | | | 1.44 – | 1.35 - | 1.63 – | 1.41 - |
| | B1 | *1.20* | | | 1.68 – | 1.71 – | 2.13 – | 1.28 - |
| | B2 | 0.13 | | | 0.12 + | 0.12 + | *0.06* ++ | 0.18– |
| | B3 | 0.14 | | | 0.14 = | 0.15 - | *0.05* ++ | 0.18 – |
| NWE | | 0.0024 | 0.0028 – | 0.0026 = | 0.0028 - | 0.0026 - | 0.003 – | *0.0023* + |
| Munich | | 4.66 | 5.90 – | 6.03 – | 5.25 – | 5.27 – | 5.49 – | *4.62* + |

propositional one when mining model trees from multi-relational data, but also the validity of multi-instance approaches in relational regression.

## 6 Conclusions

MIRT is a novel multi-relational data mining system which induces a relational model tree to predict the response value of a reference object (target object) by taking into account non-determinate task-relevant objects. This work points out the hypothesis that it is almost never the case that all multiple-bindings of

a non-determinate variable contributed to the observed response value. Under this hypothesis, a piece-wise hyperplane is constructed by locally sequencing straight-line regressions in a stepwise fashion and estimating coefficients of such regressions on the basis of only the best bindings of the regression variables. Bindings are chosen within an EM implementation that minimizes least square error on primary values. Explanatory variables involved into an hyperplane are a subset of the explanatory variables from the data relations which appear in the splitting test along the tree path from the root to a leaf. Problems of collinearity are naturally solved by stopping the hyperplane construction when the addition of a term is not statistically significant. The comparison between MIRT and the relational model tree system Mr-SMOTI as well as the propositional model tree systems, SMOTI and M5', confirm that identifying the primary instances outperforms existing propositional and structural systems.As a future study, we plan to apply principles of multi-instance learning to construct a relational model tree with split and regression nodes.

## Acknowledgments

This work is partial fulfillment of the research objective of ATENEO-2008 project "Knowledge Discovery in Relational Domains". The authors thank anonymous reviewers for their useful suggestions.

## References

1. Appice, A.: Learning Relational model Trees. PhD thesis, Department of Computer Science, University of Bari, Bari, Italy (2005)
2. Blockeel, H.: Top-down induction of first order logical decision trees. PhD thesis, Department of Computer Science, Katholieke Universiteit, Leuven, Belgium (1998)
3. Blockeel, H., Page, D., Srinivasan, A.: Multi-instance tree learning. In: Raedt, L.D., Wrobel, S. (eds.) International Conference on Machine Learning, vol. 119, pp. 57–64. ACM, New York (2005)
4. Breiman, L., Friedman, J., Olshen, R., Stone, J.: Classification and regression tree. Wadsworth & Brooks (1984)
5. Davis, J., Costa, V.S., Ray, S., Page, D.: An integrated approach to feature invention and model construction for drug activity prediction. In: Ghahramani, Z. (ed.) International Conference on Machine Learning, vol. 227, pp. 217–224. ACM, New York (2007)
6. Dietterich, T.G., Lathrop, R.H., Lozano-Pérez, T.: Solving the multiple instance problem with axis-parallel rectangles. Artif. Intell. 89(1-2), 31–71 (1997)
7. Dooly, D.R., Zhang, Q., Goldman, S.A., Amar, R.A.: Multiple-instance learning of real-valued data. Journal of Machine Learning Research 3, 651–678 (2002)
8. Draper, N.R., Smith, H.: Applied regression analysis. John Wiley & Sons, Chichester (1982)
9. Driessens, K., Džeroski, S.: Combining model-based and instance-based learning for first order regression. In: Raedt, L.D., Wrobel, S. (eds.) International Conference on Machine Learning, ICML 2005, pp. 193–205. ACM, New York (2005)

10. Džeroski, S., Blockeel, H., Kramer, S., Kompare, B., Pfahringer, B., Van Laer, W.: Experiments in predicting biodegradability. In: Džeroski, S., Flach, P.A. (eds.) ILP 1999. LNCS (LNAI), vol. 1634, pp. 80–91. Springer, Heidelberg (1999)
11. Džeroski, S., Lavrač, N.: Relational Data Mining. Springer, Heidelberg (2001)
12. Džeroski, S., Todoroski, L., Urbancic, T.: Handling real numbers in inductive logic programming: A step towards better behavioural clones. In: Lavrač, N., Wrobel, S. (eds.) ECML 1995. LNCS, vol. 912, pp. 283–286. Springer, Heidelberg (1995)
13. Hardle, W.: Applied nonparametric Regression. Cambridge University Press, Cambridge (1990)
14. Kramer, S.: Relational Learning vs. Propositionalization: Investigations in Inductive Logic Programming and Propositional Machine Learning. PhD thesis, Vienna University of Technology, Vienna, Austria (1999)
15. Kramer, S., Pfahringer, B., Helma, C.: Stochastic propositionalization of nondeterminate background knowledge. In: International Workshop on Inductive Logic Programming, London, UK, pp. 80–94. Springer, Heidelberg (1998)
16. Maron, O., Lozano-Perez, T.: A framework for multi-instance learning. Advanced in Neural Information Processing Systems
17. Orkin, M., Drogin, R.: Vital Statistics. McGraw-Hill, New York (1990)
18. Ray, S., Page, D.: Multiple instance regression. In: Brodley, C.E., Danyluk, A.P. (eds.) International Conference on Machine Learning, pp. 425–432. Morgan Kaufmann, San Francisco (2001)
19. Saith, R., Sergent, I.: Embryo selection for transfer in human IVF. Assist Reprod. Rev. 5, 145–154 (1995)
20. Srinivasan, A., Camacho, R.: Numerical reasoning with an ilp system capable of lazy evaluation and customised search. J. Log. Program 40(2-3), 185–213 (1999)
21. Srinivasan, A., Muggleton, S., King, R.D., Sternberg, M.J.E.: Mutagenesis: ILP experiments in a non-determinate biological domain. In: Wrobel, S. (ed.) International Workshop on Inductive Logic Programming, pp. 217–232 (1994)
22. Vens, C., Ramon, J., Blockeel, H.: Remauve: A relational model tree learner. In: Muggleton, S., Otero, R., Tamaddoni-Nezhad, A. (eds.) ILP 2006. LNCS (LNAI), vol. 4455, pp. 424–438. Springer, Heidelberg (2007)
23. Zhang, Q., Goldman, S.: EM-DD: An improved multiple-instance learning technique. In: Neural Information Processing Systems (2001)