MR-SMOTI: A Data Mining System for Regression Tasks Tightly-Coupled with a Relational Database

Annalisa Appice Michelangelo Ceci Donato Malerba

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

Abstract. Tight coupling of data mining and database systems is a key issue in inductive databases. It ensures scalability, direct and uniform access to both data and patterns stored in databases, as well as proper exploitation of information embedded in the database schema to drive the mining process. In this paper we present a new data mining system, named Mr-SMOTI, which is able to mine (multi-)relational model trees from a tightly coupled relational database. The induced model tree is a (multi-)relational pattern that can be represented by means of a set of selection graphs, which are translated into SQL expressions and stored in XML format. A peculiarity of induced model trees is that they can represent both local and global effects of variables used in regression models. This distinction between local patterns and global models addresses a limitation of current inductive database perspective, which mainly focus on local pattern mining tasks. Preliminary experiments demonstrate the ability of Mr-SMOTI to mine accurate regression predictors from data stored in multiple tables of a relational database.

1 Introduction

The integration of data mining with database systems is an important issue in inductive database research. Most data mining systems still process data in main memory. This results in high performance for computationally intensive processes when enough memory is available to store all necessary data. However, a common aspect of many data mining algorithms is their frequent access to data that satisfy some selection conditions. For data intensive processes, it is important to exploit powerful mechanisms for accessing, filtering and indexing data, such as those available in database management systems (DBMS). This motivates a tight coupling between data mining and database systems. In an inductive database perspective, this tight coupling also aims to support a direct and uniform access to both data and patterns stored in databases. Other equally important reasons are: i) the applicability of data mining algorithms to large data sets; ii) the exploitation useful knowledge of data model available, free of charge, in the database schema, iii) the possibility to specify directly what data stored in a database have to be mined, without any pre-processing.

The last reason is even more justified by the emergent trend in KDD research, namely (multi-)relational data mining [8], which looks for patterns that involve *multiple* relations of a relational database. Thus data taken as input by relational data mining systems typically consists of several tables and not just a single one. The *single-table assumption* [28] forces the user of traditional data mining systems to perform complex SQL queries in order to compute a single table whose rows (or tuples) represent independ ent units of analysis.

Some examples of integration of data mining and database systems are presented in [24] for association rules, in [18] for clustering and in [25] for decision trees. In [18] a system named MiningMart has been proposed for approaching the knowledge discovery in database by building upon database facilities and integrating data mining algorithms into the database environment. In all these works it has also been advocated the importance of implementing some data mining primitives to implement them using DBMS extension facilities, e.g. packages, cartridges, extenders or datablades. In [1] a package implemented in Oracle Spatial has been presented to support the extraction of spatial relations between geographical objects. This is also a rare example of (multi-)relational data mining system, named SPADA, (loosely) integrated with an object-relational spatial database. Other two examples of tight integration of (multi-)relational data mining systems with a database are MRDTL [14] and SubgroupMiner [11]. These three examples refer to the tasks of association rule mining, classification (with decision trees), and subgroup discovery, respectively.

In this work, we present Mr-SMOTI, a prototypical example of multi-relational data mining system for *regression* tasks that is tightly integrated with a relational database, namely Oracle^{R} 9i. Differently from traditional data mining regression systems (e.g. M5 [22], RETIS, [9], M5' [27], HTL [26], TSIR [15], SMOTI[16]) Mr-SMOTI directly works on complex and structured objects represented through *multiple* tables, and discovers *relational regression models* that involve attributes of several tables related by foreign key constraints.

The idea of mining regression models from data distributed in multiple tables is not new. The problem is generally solved by *moulding* a relational database into a single table format, such that traditional attribute-value algorithms are able to work on [6]. In contrast, relational regression models can be induced by formulating the problem in the *normal* ILP framework [5], where multiple relations can be directly managed through first-order representations. FORS [10], SRT [13], S-CART [8] and TILDE-RT [2] are examples of systems that solve relational regression problems by working on data stored as Prolog facts. This means that a little attention has been given to data stored in relational database and to how knowledge of data model can help to guide the search process.

Contrarily to previous works, Mr-SMOTI directly deals with multiple tables or relations as they are found in today's relational databases. Induced relational model trees can contain both regression nodes, which perform only straight-line regression, and split nodes, which partition the feature space. The model associated to each leaf is then the composition of the straight-line regressions reported along the path from the root to the leaf. In this way, internal regression nodes contribute to the definition of multiple models and capture global effects, while straight-line regressions at leaves can only capture local effects. Global effect refers to the fact that the contribution of an attribute to a regression model can be reliably estimated on more training objects

than those associated to the leaf. This overcomes one limitation of the inductive database prospective proposed in [3] that addresses only local patterns mining tasks. Mr-SMOTI upgrades the propositional system SMOTI, which induces model trees from data stored in main memory in the form of a single table. Therefore, attributes involved in nodes of relational regression models induced by Mr-SMOTI can belong to different tables of the relational database. The join of these tables is dynamically determined on the basis of the database schema.

In the next section we draw on the multi-relational regression framework, based on an extended graphical language (*selection graph*), to mine relational model trees directly from relational databases, through SQL queries. In Section 3 we show how selection graphs can support the stepwise induction of multi-relational model trees from structural data. Some experimental results are reported in Section 4. Finally, we draw some conclusions and sketch possible directions of further research.

2 Regression problem in a multi-relational framework

Traditional research for a regression task in KDD has focused mainly on propositional techniques involving the attribute-value paradigm. This implies that relationships between fields of one tuple can be found, but not relationships between several tuples of one or more tables. It seems that this is an important limitation, since a relational database consists of a set of tables and a set of associations. Each association describes how records in one table relate to records in another table. Most associations correspond to *foreign key relations*. These relations can be seen as having two directions. One goes from a table where the attribute is primary key to a table where the attribute is foreign key (*one-to-many*), and the other one is in the reverse way (*many-to-one*). An object in a relational database can consist of several records fragmented across several tables and connected by associations (Fig. 1). Although the data model can consist of multiple tables, there must be only a single kind of object that is central to the analysis (*target table*). The assumption is that each record in the target table will correspond to a single object in the database. Any information pertaining to each object which is stored in other tables can be retrieved by following the associations in the data model. Once the target table has been selected, a particular numeric attribute of that table can be chosen for regression purposes (*target attribute*).

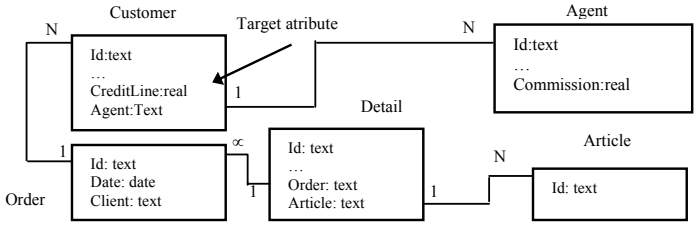

Fig. 1. The data model of an example database used in relational regression.

Thus, a multiple regression problem in a multi-relational framework can be defined as follows. Given a schema of a relational database D, a target table T_0, a target attribute Y within the target table T_0, the goal is to mine a multi-relational multiple regression model to predict the estimated target attribute Y. Mined models

not only involve attribute-value descriptions, but also structural information denoted by the associations in *D*.

Relational regression models, stepwise induced as in SMOTI can be expressed in the graphical language of *selection graphs*. The classical definition of a selection graph is reported in [12]. Nevertheless, we present an extension of this definition in order to make the selection graphs more appropriate to our task. In particular the selection graph must be able to represent a (multi-)relational regression model incrementally built. The incremental construction is based on the idea that when a new independent variable is added to the model its linear effect on remaining variables has to be removed [4].

Definition of *selection graph*

A selection graph G is a directed graph (N, A), such that:

– each node in N is a 4-tuple (T, C, R, s), named *selection node*, where:
  - $T = (X_1, X_2, \ldots X_n)$ is a table in the relational schema D.
  - C is a set of *conditions* on attributes in T of type $T.X'_i \; OP \; c$, where X'_i is one of the attributes X_i in T after the removal of the effects of some variables already introduced in the relational regression model through regression nodes. *OP* is one of the usual comparison operators (<, ≥, in, not in …) and c is a constant value.
  - R is a set of tuples $R=\{(RX_j, \boldsymbol{\alpha_j}, \boldsymbol{\beta_j}) \mid j=1,\ldots,l\}$ where RX_j is a regression term already introduced in the multiple linear model, l is the number of such terms, $\boldsymbol{\alpha_j} = (\alpha_{j1}, \alpha_{j2}, \ldots, \alpha_{jn})$ and $\boldsymbol{\beta_j} = (\beta_{j1}, \beta_{j2}, \ldots, \beta_{jn})$ are the regression coefficients computed to remove the effect of each term RX_j from all numerical attributes in T:
    $X'_i = X_i - \sum_{j=1,\ldots,l} (\alpha_{ji} + \beta_{ji} \times RX_j) \quad \forall \; i = 1,\ldots,n$ and X_i is numerical
  - s is a flag with possible values *open* or *closed.*

– A, a set of tuples (p, q, fk, e), where:
  - p and q are selection nodes.
  - fk is a foreign key association between $p.T$ and $q.T$ in the relational schema D (one-to-many or many-to-one).
  - e is a flag with possible values *present* or *absent*.

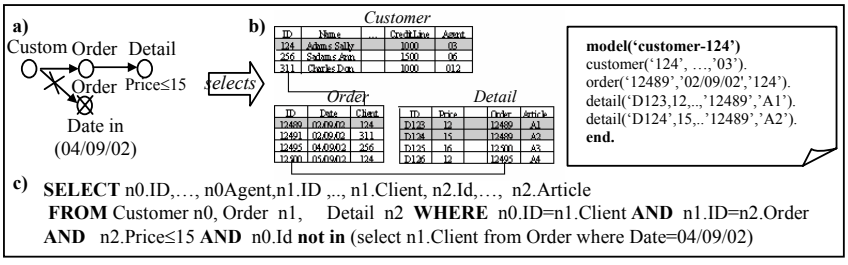

Fig. 2. (a) Example of selection graph; (b) corresponding grouping of data for an instance of the example database and (c) translation into an SQL query.

Selection graphs contain at least a node n_0 that corresponds to the target table T_0. They can be graphically represented by a directed labelled graph (Fig. 2.a). The value of s is expressed by the absence or presence of a cross in the node, representing the value *open* and *close* respectively. The value for e is indicated by the presence (*absent*

value) or absence (*present* value) of a cross on the corresponding arrow representing the labelled arc. The direction of the arrow (left-to-right and right-to-left) corresponds to the multiplicity of the association *fk* (one-to-many and many-to-one, respectively). Every arc between the nodes p and q imposes some constraints on how one or more records in the table $q.T$ are related to each record in table $p.T$ according to the list of conditions in $q.C$. The association between $p.T$ and $q.T$ induces some grouping (Fig. 2.b) in the records in $q.T$, and thus selects some records in $p.T$. In particular, a *present* arc selects those records that belong to the *join* between the tables and match the list of conditions. On the other hand, an *absent* arc corresponds to the negation of the joining condition and the representations of the complementary sets of objects. Intuitively, the tuples in the target table T_0 that are explained by a selection graph G are those for which tuples exist or not in linked tables that satisfy the conditions defined for those tables. The given definition of selection graph does not allow to represent recursive relationships. Therefore a selection graph can be straightforwardly translated into either SQL or into first order logic expressions (Fig. 2.c). In this case a subgraph pointed by an absent arc is translated into a negated inner sub-query.

3 Multi-relational stepwise model tree induction

Mr-SMOTI induces model trees whose nodes (regression, split or leaf) involve multi-relational patterns that can be represented with *selection graphs*, that is *each node of the tree corresponds to a selection graph*. Essentially Mr-SMOTI, like the propositional version SMOTI, builds a tree-structured multi-relational regression model by adding split and/or regression nodes through a process of successive refinements of the current selection graph until a stopping criterion is fulfilled and a leaf node is introduced. Thus, the model associated to each leaf is computed by combining all straight-line regressions in the regression refinements along the path from the root to the leaf.

3.1 The algorithm

Mr-SMOTI is basically a *divide-and-conquer* algorithm that starts with a root selection graph G containing only the target node n_0. This graph corresponds to the entire set of objects of interest in the relational database D (the target table T_0). At each step the system chooses the optimal refinement (split or regression) according to a heuristic function. In particular, a split refinement corresponds to either the updating of an existing node by adding a new selection condition or the introduction of a linked node in the current selection graph. On the other hand, a regression refinement corresponds to update the list of regression terms in existing nodes in order to remove the linear effect of those numeric attributes already included in the model Thus, descendants of a regression node must operate on modified training data. This transformation is coherent with the statistical procedure for the incremental construction of multiple linear regression models, according to which each time a new independent variable is added to the model its linear effect on remaining variables has to be removed [4].

The eventually modified training tuples selected by the optimal refinement (and its complement in case of a split), are used to select the regression functions

associated to the root of the left (/right) branch. This procedure is recursively applied to each branch until a stopping criterion is fulfilled.

Mr-SMOTI (D: database, G: selection_graph)
G_S, G_R, R: selection_graph; T_left, T_right: model_tree;
G_R:= *optimal_regression_refinement* (G, D);
if *stopping_criteria* (G_R, D) then return *leaf* (G_R);
G_S:= *optimal_split_refinement* (G, D);
R:= *best_refinement* (G_R, G_S);
if(R=G_R) T_left := *Mr-SMOTI* (D,R); T_right := ∅;
else T_left := *Mr-SMOTI* (D,R); T_right := *Mr-SMOTI* (D, *comp* (R));
return *model_tree*(R, T_left, T_right).

The functions *optimal_split_refinement* and *optimal_regression_refinement* take the selection graph *G* associated to the current node and consider every possible split and regression refinement. The choice of which refinements are candidates is determined by the current selection graph *G*, the structure of data model in *D*, and notably by the multiplicity of associations within this data model. The validity of either a split refinement (G_S) together with its complement (*comp*(G_S)), or a regression refinement (G_R) is based on two distinct evaluation measures, $\sigma(G_S, comp(G_S))$ and $\rho(G_R)$, respectively. Let *T* be the multi-relational model tree currently stepwise built, *G* the selection graph associated to the node *t* in *T* and t_{Gs} ($t_{comp(Gs)}$) the left (right) child of *t*, associated to a split refinement G_S (the complementary split refinement *comp*(G_S)) of the selection graph *G*, $\sigma(G_S, comp(G_S))$ is defined as:

$$\sigma(G_S, comp(G_S)) = \frac{N(t_{G_S})}{N(t_{G_S}) + N(t_{comp(G_S)})} R(G_S) + \frac{N(t_{comp(G_S)})}{N(t_{G_S}) + N(t_{comp(G_S)})} R(comp(G_S)),$$

where $N(t_{Gs})$ ($N(t_{comp(Gs)})$) is the number of training tuples covered by the refinement G_S (*comp*(G_S)), and $R(G_S)$ ($R(comp(G_S))$ is the resubstitution error of the left (right) child, computed as follows:

$$R(G_S) = \sqrt{\frac{1}{N(t_{G_S})} \sum_{j=1}^{N(t_{G_S})} (y_j - \hat{y}_j)^2} \qquad R(comp(G_S)) = \sqrt{\frac{1}{N(t_{comp(G_S)})} \sum_{j=1}^{N(t_{Comp(G_S)})} (y_j - \hat{y}_j)^2}.$$

Therefore the evaluation measure $\sigma(G_s, comp(G_s))$ is coherently defined on the basis of the partially defined multiple linear regression models $\hat{Y}$ built by combining the best straight-line regression associated to t_{Gs} ($t_{comp(Gs)}$), with all regressions introduced along the path from the root to t_{Gs} ($t_{comp(Gs)}$).

In the case of a regression refinement G_R, the definition of a heuristic evaluation function $\rho(G_R)$ of the effectiveness of G_R cannot be naïvely based on the resubstitution error $R(G_R)$ [16]. Indeed, the splitting test "looks-ahead" to the best multiple linear regressions after the current split is performed, while the regression step does not perform such a look-ahead. A fairer comparison would be to grow the model tree at a further level in order to base the computation of $\rho(G_R)$ on the best split refinement G_{Rs}, after the current regression refinement is performed. Therefore, $\rho(G_R)$ is defined as follows:

$$\rho(G_R) = min\ \{R(G_R),\ \sigma(G_{Rs}, comp(G_{Rs}))\}.$$

The function *stopping_criteria* determines whether the current optimal refinement must be transformed into a leaf according to the minimal number of *target objects*

(*minObject*) covered by the current selection graph and the minimal threshold for the *coefficient of determination* (*minR*) of the prediction function built stepwise [4].

The regression model built stepwise by Mr-SMOTI is a set of SQL queries, each of which is associated to a node in the tree. SQL queries are stored XML format and can be in turn the object of a query according to an inductive database perspective. Moreover, they can be applied to new instances stored in the relational database in order to predict an estimate of the unknown target attribute. The prediction is averaged by means of a grouping on the target objects.

3.2 The refinements

Split refinements are an extension of the refinement operations proposed in [12] to perform a split node in a multi-relational decision tree. Whenever a split is introduced in a model tree, Mr-SMOTI is in fact refining the selection graph associated to the current node, by adding either a condition or an open node linked by a present arc. Given a selection graph G, the *add condition* refinement returns the refined selection graph G_S by simply adding a split condition to an open node $n_i \in G.N$ without changing the structure of G.

The *add linked node* refinement instantiates an association of the data model D by means of a *present arc*, together with its corresponding table, represented as an *open node*, and adds these to the selection graph G. Knowledge of the nature and multiplicity is used to guide and optimise this search. Since the investigated associations are *foreign key associations*, the proposed refinements can have two directions: *backward* or *forward*. The former correspond to *many-to-one* associations, while the latter describe *one-to-many* associations in the data model. This means that a backward refinement of the selection graph G does not partition the set of target objects covered by G but extends their descriptions (training data) by considering tuples joined in the table which are represented by the new added node. Each split refinement G_s of type *add condition* or *add linked node* is introduced together with its complementary refinement ($comp(G_S)$) in order to satisfy the mutual exclusion principle. Let Q_G be the SQL or first order expression translating the selection graph G, and Q_{Gs} ($Q_{comp(Gs)}$) the expression translating the split refinement (complementary refinement) Q_{Gs} ($Q_{comp(Gs)}$). For each target object selected by Q_G exactly one of both queries (Q_{Gs} and $Q_{comp(Gs)}$) should succeed.

In [12], Knobbe *et al.* propose a complementary refinement named *add negative condition* that should solve the problem of mutual exclusion between an *add condition refinement* and its *complement*. If the node that is being refined does not represent the target table, $comp(G_S)$ must be built from G by introducing an absent arc from the parent of n_i to the clone of the entire sub-graph of G that is rooted in n_i. The introduced sub-graph has a root (a clone of the node to be refined) that is a closed node updated with the refinement condition that is not negated. In this way the query translating $comp(G_S)$ negates an entire inner sub-query and not simply a condition. As was observed in [14], this approach fails to build complementary refinements when the node to be refined is *not directly* connected to the target node. The example in Figure 4 proves that the proposed mechanism could build a refinement G_S (Fig 4.a) and a complementary refinement $comp(G_S)$ (Fig 4.b) that are not mutually exclusive. To overcome this problem the complementary refinement $comp(G_S)$ should be

obtained by adding an absent arc from the target node n_0 to the clone of the sub-graph containing the *entire join path* from the target node to the node to be refined. The introduced sub-graph has a root (a clone of n_0) that is a *closed* node and is updated with the refinement condition that is not negated. A new absent arc is also introduced between the target node and its closed clone. This arc is an instance of the implicitly relationship between the primary key of the target table and the own itself (Fig 4.c).

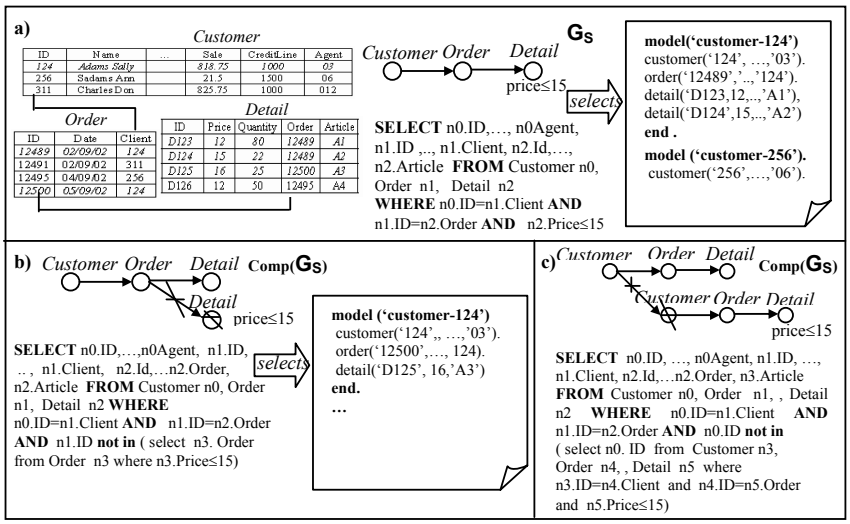

Fig. 4. Example of (a) refinement (G_S) by adding the a condition on a node not directly connected to the target node, (b) the corresponding complementary refinement, proposed in [11], that does not satisfy the mutual exclusion and (c) correct complementary refinement.

Similarly, when we consider the complementary refinement for an *add linked node* refinement we make the same considerations as when a negated condition is going to be added. This means that when the closed node to be added is not directly connected to the target node in G, a procedure similar to that described when an add condition refinement is complemented must be followed.

Finally, a *regression refinement* G_R of the selection graph G corresponds to performing a regression step ($Y'=\alpha_Y+\beta_Y \times n_i.T.X_j'$) on the residuals of a continuous attribute not yet introduced in the model currently. The coefficients α_Y and β_Y are estimated using all (joined) tuples selected by the current selection graph G. This means that the regression refinement is performed by considering a propositionalization of the (multi-) relational descriptions of the training objects selected by G. The regression attribute must belong to a table represented by a node in G. For each node, the list of regressions R is updated by adding the regression term ($n_i.T.X_j'$) introduced in the model and the coefficients α and β computed to update the residuals of all continuous attributes in the node.

4 Experimental evaluation

Mr-SMOTI has been applied to the biological problems of predicting both the mutagenic activity of molecules [19] and the biodegradability of chemical compounds in water [7]. *Mutagenesis* dataset consists of 230 molecules divided into two subsets: 188 molecules for which linear regression yields good results and 42 molecules that are regression-unfriendly. In our experiments we used the atom and bond structure of regression-friendly molecules by adding boolean indicators *Ind1* and *Ind2* as one setting (B_1) and adding *Lumo* and *Logp* properties to get a second setting (B_2). Similarly *Biodegradability* dataset consists of 328 chemical molecules structurally described in terms of atoms and bonds. In all the experimental results reported below the thresholds for stopping criteria are fixed as follows: *minObject*is is set to the square root of the number of target objects in the entire training set and *minR* must be below 0.80. Each dataset is analysed by means of a 10-*fold* cross-validation. Figure 5 shows the test set performance of *Mr-SMOTI* and *TILDE-RT* in both domains, as measured by the *Pearson correlation coefficient* that measures of how much the value of target attribute (y_j) in test objects correlates with the value predicted by the induced model. Since the Pearson correlation coefficient does not measure the quantity error of a prediction, we include several other measures [23] such the average error (*AE*) and the root mean square error (*RMSE*). For pairwise comparison with *TILDE-RT* the non-parametric Wilcoxon two-sample paired signed rank test is used [21]. The results of the Wilcoxon signed rank test on the accuracy of the induced multi-relational prediction model are reported in Table 1.

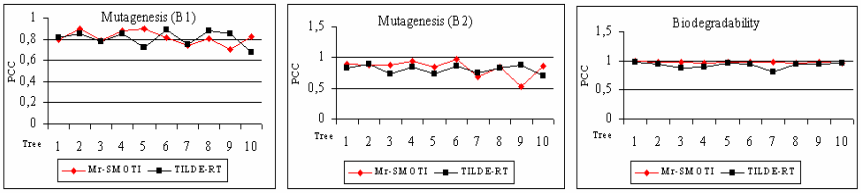

Fig. 5 Pearson correlation coefficient (Y axis) for multi-relational prediction models induced from the 10-fold cross validated datasets (X axis) of Mutagenesis (B1, B2) and Biodegradability datasets. The comparison concerns two systems: TILDE-RT (square) vs. Mr-SMOTI (diamonds).

Table 1. Results of the Wilcoxon signed rank test on the accuracy of the induced models. The best value is in boldface, while the statistically significant values ($p \leq \alpha/2, \alpha=0.05$) are in italics.

Dataset		*Accuracy*	*Mr-SMOTI*	*TILDE-RT*	*W+*	*W-*	*P*
Mutagenesis	*B1*	*Avg.MSE*	**1.165**	1.197	23	32	0.69
		Avg.AE	**0.887**	0.986	12	43	0.13
	B2	*Avg.MSE*	**1.118**	1.193	15	40	0.23
		Avg.AE	**0.845**	0.985	11	44	0.10
Biodegradability		*Avg.MSE*	***0.337***	0.588	0	55	*0.0019*
		Avg.AE	***0.186***	0.363	0	55	*0.0019*

The Wilcoxon test statistics W^+ (W^-) is the sum of the ranks from the positive (negative) differences between TILDE-RT and Mr-SMOTI. Therefore, the smaller

W^+ (W^-), the better for Mr-SMOTI (TILDE-RT). Differences are considered statistically significant when the p-value is less than or equal to $\alpha/2$. Interestingly all experimental results confirm the good performance of Mr-SMOTI.

5 Conclusions

This paper presents a novel approach to mining relational model trees. The proposed algorithm can work effectively when training data are stored in multiple tables of a relational DBMS. Information on the database schema is used to reduce the search space of patterns. Induced relational models are represented by selection graphs whose definition has been extended in order to describe model trees with either split nodes or regression nodes. As future work, we plan to extend the comparison of Mr-SMOTI to other multi-relational data mining systems on a larger set of benchmark datasets. Moreover, we intend to use SQL primitives and parallel database servers to speed up the stepwise construction of multi-relational model trees from data stored in large database. Finally, following the mainstream of our research on data mining query languages for spatial databases with an object-oriented logical model[17], we intend to pursue the investigation of defining a data mining query language appropriate to support both the discovery and the query of model trees.

Acknowledgments

This work has been supported by the annual Scientific Research Project "Metodi di apprendimento automatico e di data mining per sistemi di conoscenza basati sulla semantica" Year 2003, funded by the University of Bari. The authors thank Hendrik Blockeel for providing mutagenesis and biodegradability datasets.

References

[1]Appice A., Ceci M., Lanza A., Lisi F.A., Malerba D.: *Discovery of Spatial Association Rules in Georeferenced Census Data: A Relational Mining Approach*, Intelligent Data Analysis, numero speciale su "Mining Official Data" (in press).

[2]Blockeel H.:*Top-down induction of first order logical decision trees*. Ph.D thesis, Department of Computer Science, Katholieke UniversReit Leuven, 1998.

[3]De Raedt L.: A perspective on inductive databases. In *SIGKDD Explorations ACM,* Gehrke J (Ed.) Volume 4, Issue 2, 2002

[4]Draper N.R. & Smith H.: *Applied regression analysis*, John Wiley & Sons, 1982.

[5]Dzeroski S.: *Numerical Constraints and Learnability in Inductive Logic Programming*. Ph.D thesis, University of Ljubljana, Slovenia, 1995.

[6]Dzeroski S., Todoroski L. & Urbancic T: Handling real numbers in inductive logic programming: A step towards better behavioural clones. In *Machine Learning: ECML-95*, Eds. Lavrac N & Wrobel S., Springer , Berlin Heidelberg New York, 1995.

[7]Dzeroski S., Blockeel H., Kramer S., Kompare B., Pfahringer B., and Van Laer W.. Experiments in predicting biodegradability. *Proceedings of the Ninth International*

Workshop on Inductive Logic Programming (S. Dzeroski and P. Flach, eds.), LNAI, vol. 1634, Springer, pp. 80-91, 1999.
[8]Dzeroski S. & Lavrac N. (Eds). *Relational Data Mining*. Springer-Verlag, 2001.
[9]Karalic A.: Linear regression in regression tree leaves. In *Proc. of ISSEK '92 (International School for Synthesis of Expert Knowledge)*, Bled, Slovenia, 1992.
[10]Karalic A.: First Order regression. Ph.D thesis, University of Ljubljana, Slovenia, 1995.
[11]Klosgen W. & May M.:Spatial Subgroup Mining Integrated in an Object-RelationalSpatial Database. In *Principles of Data Mining and Knowledge Discovery,6th European Conference, PKDD 2002*,Elomaa T.,Mannila H. & Toivonen H.(Eds.),Helsinki, Finland, Springer-Verlag , 2002.
[12]Knobbe J., Siebes A.&Van der Wallen D.M.G: Multi-relational decision tree induction.In *Proc. 3rd European Conf. on Principles and Practice of Knowledge Discovery in Databases*,PKDD'99, 1999.
[13] Kramer S.: Structural regression trees. In *Proc. 13th National Conf. on Artificial Intelligence*, 1996.
[14]Leiva H.A.:MRDTL: *A multi-relational decision tree learning algorithm*. Master thesis, University of Iowa, USA, 2002.
[15]Lubinsky D.: Tree Structured Interpretable Regression. In *Learning from Data*, Fisher D. & Lenz H.J. (Eds.), Lecture Notes in Statistics, 112, Springer, 1994.
[16]Malerba D., Appice A., Ceci M. & Monopoli M.: Trading-off versus global effects or regression nodes in model trees. In *Foundations of Intelligent Systems, 13th International Symposium, ISMIS'2002*, Hacid H.S., Ras Z.W. , Zighed D.A. & Kodratoff Y. (Eds.), Lecture Notes in Artificial Intelligence, 2366, Springer, Germany, 2002.
[17]Malerba D., Appice A. & Vacca N.:SDMOQL: An OQL-based Data Mining Query Language for Map Interpretation Tasks. In *Proc. of the EDBT Workshop on "Database Technologies for Data Mining"*, Prague, Czech Republic, 2002.
[18]Morik K. & Scholz M.: *The MiningMart Approach to Knowledge Discovery in Databases*. In Handbook of Intelligent IT, Ning Zhong and Jiming Liu (Eds.),IOS Press,2003, to appear.
[19]Muggleton S., Srinivasan A., King R. & Sternberg M.: Biochemical knowledge discovery using Inductive Logic Programming. In *Proceedings of the first Conference on Discovery Science*, Motoda H. (ed), Springer-Verlag, Berlin, 1998.
[20]Ordonez C. & Cereghini P.: SQLEM: Fast Clustering in SQL using the EM Algorithm. In *Proc. ACM SIGMOD 2000,* Chen W., Naughton J.& Bernstein P.(Eds.), Dallas, USA, vol. 29, 2000.
[21] Orkin. M. & Drogin. R.: *Vital Statistics*. McGraw Hill. New York . 1990.
[22]Quinlan J. R.: Learning with continuous classes, in *Proceedings AI'92*, Adams & Sterling (Eds.), World Scientific, 1992.
[23]Quinlan J. R.: A case study in Machine Leaning, in *Proceedings ACSC-16*, Sixteenth Australian Computer Science Conferences, 1993.
[24] Sarawagi S., Thomas S. & Agrawal R.: Integrating Mining with Relational Database Systems: Alternatives andImplications. In *Proc. ACM SIGMOD'98,* L. Haas and A. Tiwary (Eds), Seattle, *USA*., 1998.
[25]Sattler K. & Dunemann O.: SQL Database Primitives for Decision Tree Classifiers. In *Proc. of the 10th ACM CIKM Int.Conf. on Information and Knowledge Management*, Atlanta, USA, 2001.
[26]Torgo L.: Functional Models for Regression Tree Leaves. In *Proceedings of the 14th International Conference (ICML 97)*, D. Fisher (Ed.), Nashville, Tennessee, 1997.
[27]Wang Y. & Witten I.H.: Inducing Model Trees for Continuous Classes. In *Poster Papers of the 9th European Conf. on Machine Learning (ECML 97)*, M. van Someren, & G. Widmer (Eds.), Prague, Czech Republic,1997.
[28]Wrobel, S.: Inductive logic programming for knowledge discovery in databases. In Dzeroski S. & Lavrac N. (Eds). *Relational Data Mining*. Springer-Verlag, 2001.