

Precision and comprehensibility in the integration of regression rules

J. B. Pugliesi & S. O. Rezende

*Institute of Mathematics and Computer Sciences—ICMC,
University of São Paulo—USP, São Carlos, Brazil*

Abstract

Data Mining is the process of data analysis and application of algorithms that are able to find a particular pattern relationship from a large amount of data. Regression is an important problem in data analysis and appears in many real world applications. Therefore, there is an increasing interest in the use of the Data Mining to extract patterns from regression problems. The main objective of this paper is to support the users of Data Mining process in the evaluation of integrated knowledge expressed in the form of regression rules. To explore the knowledge evaluation, experiments are executed with Cubist, RT and $\mathcal{M}5$ algorithms. The 10-fold cross-validation is used to find the mean error. All regression rules are transformed to a standard syntax and put together in a unique set of rules. The worst rules are eliminated from this set to generate a more concise regressor. This final regressor has less comprehensibility and higher precision than a single regressor induced from all data. A method is defined to determine how to predict the new examples with this final regressor. From the data analysis, it is possible to take some conclusions, which show that usually it is possible to eliminate the worst regression rules and have an accurate regressor with a good level of comprehensibility.

1 Introduction

The information age has opened ways that lead to a new one, the knowledge age. Thus, efforts to facilitate the manipulation and the understanding of information has gained importance. This is motivated by: the popularization of the main and largest computer network, the Internet; the advances in data collection and storage; the constant development of companies in what concerns to their technological park; and, mainly, the increase of the necessity of information.

Due to this context, there is an increasing interest of researchers in the Knowledge Discovery in Databases process investigation with the purpose of presenting useful solutions in different domains.

The knowledge extraction from great amount of data is an interactive and iterative process. It is a process centered in the interaction among end-users, domain specialists and knowledge engineers. In that way, one cannot simply expect an extraction of useful knowledge submitting a data set to a “black box”.

With the increase of database sizes, the amount of knowledge produced by learning algorithms also increases. Thus it is not productive to give to the user a great amount of patterns mined from a database. Usually, users prefer to look at a small list of interesting patterns that can be generated by the knowledge discovery tool. The knowledge evaluation can include some problems, such as, to find patterns with good precision and high comprehensibility.

The discovery of patterns that are interesting to the end-user is a difficult problem. An used approach is quickly eliminate a set of rules that are not interesting, while it limits the user interaction with few and simple prediction questions [1]. In order to aid the user in the choose of these rules, forms of evaluating the knowledge obtained as answer of one or more regressors are being studied. We do not treat them as a closed module, but as an explicitly knowledge represented in a knowledge base.

A point that earn prominence when working with great databases is that we use a small sample of the data, due to the limitations of the used knowledge acquisition methods. Even trying to select representative samples—what becomes complicated in certain domains—this sort of procedure cuts out part of the data, what can cause the loss of important knowledge, or the generation of different knowledge bases from different samples of the database. Therefore, a combination of obtained results from several samples becomes important, so that we could take advantage of the whole important knowledge that was extracted from the database. On the other hand, with the same sample, knowledge can be extracted using different algorithms with different bias and different concept description languages.

Aiming at the quality of the obtained knowledge, regression rule integration is being used to improve the prediction accuracy of new examples. Besides, interestingness and comprehensibility measures are also being used, in order to choose the most appropriate rule set for a certain problem, reaching the final user’s objectives.

This work is organized as follow. Section 2 defines a regression problem and the notation used in this work. Section 3 discusses the measures for rule evaluation and methods for rule integration. Section 4 presents the accomplished experiments and their results, as well as the used algorithms and databases. Finally, Section 5 shows the conclusion of this work.

2 Characterization of regression problems

Classification problems are more common than regression problems, what explains the greater attention given to classification than regression. However, many important problems are regression-related. Besides the fact that regression problems are

Table 1: Contingency table for the rule $B \rightarrow H$.

	B	$\overline{B}$	
H	$n(HB)$	$n(H\overline{B})$	$n(H)$
$\overline{H}$	$n(\overline{H}B)$	$n(\overline{H}\overline{B})$	$n(\overline{H})$
	$n(B)$	$n(\overline{B})$	N

important *per se*, another reason for the need to focus on regression is that regression methods can be utilized to solve classification problems. Neural networks, for instance, are often applied to classification problems [2].

In regression, each example is given a numerical value instead of a finite set of symbolic labels. In other words, given a set of input variables $x_1, \dots, x_n$, together with a meta-attribute variable y , one attempts to approximate the function presented in the Equation (1).

$$y = f(x_1, \dots, x_n). \quad (1)$$

Regression problem is, sometimes, described as a signal and noise problem. Thus, the model is extended to include a stochastic component or a ϵ threshold in the function, resulting in Equation (2). In this case, the real function cannot produce a zero error distance. Contrary to classification, in which the labels are assumed to be correct, in regression, the predicted y values can be explained by several factors, including a component of random noise, ϵ , in the signal, y .

$$y = f(x_1, \dots, x_n) \pm \epsilon. \quad (2)$$

Since this work is related to the evaluation of knowledge regression rules, consider a rule, that is usually represented as the following:

$$\text{if } \underbrace{\langle \text{condition} \rangle}_{B\text{-Body}} \text{ then } \underbrace{\langle \text{class} = C_i \rangle}_{H\text{-Head}}$$

where $\langle \text{condition} \rangle$ is a disjunction of conjunctions of attributes and C_i is a numeric value or an equation.

For the evaluation measures presented here, the symbolic representation ($B \rightarrow H$) will be adopted. The contingency table can be built, for any rule, starting from the data set. In the Table 1, B denotes the set of examples where the rule condition is true, and $\overline{B}$ denotes its complement (the set of examples where the rule condition is false); and analogously H and $\overline{H}$. HB denotes $H \cap B$, $\overline{H}B$ denotes $\overline{H} \cap B$ and so on. The $n(X)$ denotes the cardinality of the set X . For instance, $n(B)$ denotes the number of examples in the set B , and $n(\overline{H}B)$ is the number of examples where H is false and B is true (i.e., the number of examples erroneously covered by the rule). Also, N is the total number of examples in the database.

Take into account that several evaluation knowledge measures use in their formulas some of the contingency table values, thus the calculation of this table is very important [3, 4].

Since some regression algorithms give as class or head (*Head*) of the rule a mathematical equation based on the attributes, that generates for each example a real value, it is necessary to use a threshold for the construction of the contingency table. This is due to the fact that is necessary to calculate the number of examples for which the head of the rule is true or false.

Therefore, the calculation of the number of false and true examples in terms of the meta-attribute (i.e., the coverage of the rule) is more complex than in classification because it is continuous. To carry out this check, we propose a strategy based on a threshold using standard deviation to verify which examples are correctly covered by the regressor [5].

3 Rule evaluation and integration

A learning algorithm output should be accurate, stable and comprehensible to be qualified as knowledge. In order to verify if the acquired knowledge has such characteristics, a post-processing phase can be accomplished to evaluate the performance and the quality. This evaluation tries to investigate the knowledge precision, the model representation, the complexity and the comprehensibility of the extracted knowledge. Besides, the post-processing phase aims of filtering what was learned, eliminating the spurious, invalid or obvious knowledge.

It should be noted that sometimes, when the performance of the regressor is not considered good, important knowledge might exist in terms of comprehensibility and interestingness. This is one of the factors that motivate researches in knowledge evaluation in machine learning and Data Mining.

The knowledge evaluation can involve some problems, such as, the difficulty of finding reliable knowledge, and the lack of reliable quality measures to serve as support to the user. As consequence, the evaluation is intimately related to the interpretation of the obtained knowledge.

To make an evaluation of the obtained knowledge, it is still necessary to separate the one that has little validity or usefulness for an end-user. To employ this evaluation, some measures can be used in regression problems, such as the ones obtained from the contingency tables [4]. Besides, there are others error measures that do not use the contingency table values, such as MAD (Mean Absolute Deviation), MSE (Mean Squared Error) and RMSE (Relative Mean Squared Error) [6, 7, 8].

The problem of post-processing sets of rules is not trivial. Other researchers have addressed this issue [9, 10, 1, 11, 12, 13].

This is especially true for regression rule discovery, where rule sets are not easily comprehensible, and a rule analysis can be very useful. Some evaluation measures are offered to the user. Furthermore, when the number of rules is large, it is very difficult for the user manually analyze all the rules.

Another point for the realization of knowledge post-processing is the knowledge integration. Knowledge integration refers to the process of incorporating new knowledge to an exist knowledge base. This involves to determine how different knowledge interacts, and how the knowledge from different knowledge bases should be modified to accommodate one another. Knowledge integration involves

the construction of a knowledge base from several learning systems and then the merge of the knowledge. An important point is the usage of different constructed knowledge bases.

In this paper, the main objective is to explore the knowledge base integration. For this, a method for model and rule evaluation has been defined. To increase the model comprehensibility empirical experiments have been done to eliminate the worst regression rules.

4 Experimental results

We use three databases (Auto-Mpg, Housing and Abalone) and three regression algorithms (Cubist, $\mathcal{M}5$ and RT) to execute the experiments. The databases and algorithms are described in the next subsections. All the regressors are transformed into a set of rules using a defined common syntax.

In the experiments, each database was divided in two sets: one with 75% of the examples, and the other with 25% of the examples which is the validation set. To generate the rule sets, the 10-fold cross-validation was executed in the data set with 75% of the examples, and the rules from all the 10 rule sets was gathered in a integrated rule set. Besides, another regressor was generated from all the data set with 75% of the examples to compare its results with the integrated rule set results.

We defined a metric to evaluate the rule set error. When more than one rule cover an example, an average of the predicted values was used as a final prediction of this example. Whether the example was not covered by any rule, a proximity function is used. This function considers a weighted average based on the two rules more closed to the example, i.e., the rules that almost cover the example.

After the integration, a rule selection is realized. This rule selection eliminates 5%, 10%, 25% and 50% of the worst rules. The elimination of the worst rules is based on two different measures. One measure is the MAD (Mean Absolute Deviation) (Equation (3)). This measure quantifies the error of the model for the average in relation to the absolute deviations of the predictions, that is, MAD consists of the average of the difference (in module) among the real and predicted values for an attribute-goal.

$$\text{MAD}(R) = \frac{1}{N} \sum_{i=1}^N |y_i - y'_i|. \quad (3)$$

In this equation y is the meta-attribute value for the training example, N is the number of the training examples, and y' is the predicted value for the training example.

Already, another measure, that uses the contingency table values to regression problems [5], is the coverage and it calculates the percentage of the covered examples that is correct (Equation (4)).

$$\text{Cov}(R) = \frac{n(HB) - n(\overline{H}B)}{N}. \quad (4)$$

4.1 Algorithms

The task of any regression algorithm is estimate the value of the meta-attribute variable of a new example based on the other variables. The regression algorithms used in these initial experiments were Cubist, RT and $\mathcal{M}5$.

Cubist is a tool used to generate predictive numeric models based on rules from a data set [14]. These models are used to predict numeric values, i.e., for regression problems. Each case provided to Cubist has a meta-attribute and the others attributes are used to predict the value of this meta-attribute. Cubist generates unordered rules, disposed in growing order of the mean value of the predicted examples. Cubist estimates the meta-attribute value with a function of the input variables. For that, he builds a model containing one or more rules, in which each rule is a conjunction of conditions associated to a linear expression. Therefore, the generated models by Cubist are linear piecewise models. However, Cubist can also build multiple models and it can combine models based on rules with models based on examples.

RT (Regression Trees) algorithm allows the acquisition of regression models in different representation types based on a given examples set [15]. Some models types that can be obtained are: model based on regression trees, in which the leaves are labelled with the average of the values; parametric global model; and local model, using the K-Nearest Neighbor algorithm. RT still allows the emulation of the regression algorithms RETIS (Regression Tree Induction System) and CART (Classification And Regression Trees).

$\mathcal{M}5$ is an algorithm developed by Quinlan [16] to treat continuous attributes and classes. The $\mathcal{M}5$ belongs to the TDIDT family (Top Down Induction Decision Trees), but with linear regression functions in the leaves, instead of a categorical value predicting the class. The $\mathcal{M}5$ algorithm was based on the initial work of Breiman with the CART algorithm. In the experiments, we used the $\mathcal{M}5$ implementation belonging to the Weka tool. The Weka (Waikato Environment for Knowledge Analysis) was developed in the Waikato University, in New Zealand [17]. This tool can be seen as a collection of machine learning algorithms that can be used to solve real world Data Mining problems.

4.2 Databases

In our experiments, three databases was used: Auto-Mpg, Housing and Abalone. All the databases was obtained from the UCI Machine Learning Repository [18].

Auto-Mpg This database concerns the city-cycle fuel consumption. The data set at UCI had originally 9 attributes (the first being the goal variable, the mpg attribute). The last attribute (the car model) is unique for all examples and it was removed. The first attribute (the goal variable) was moved to the end, so now is the last attribute. Characteristics: 398 cases, 7 attributes (4 continuous, 3 nominal).

Housing This data set concerns the task of predicting housing prices in areas of Boston. Each example from this database describe an area with characteristics, such as: crime rate by town, average number of rooms per dwelling, index of accessibility to radial highways, and percentage lower status of the population. Characteristics: 506 cases, 13 attributes (12 continuous, 1 binary).

Abalone This data set can be used to obtain a model to predict the age of a abalone from physical measurements. The age of abalone is determined by cutting the shell through the cone, staining it, and counting the number of rings through a microscope—a boring and time-consuming task. Other measurements, which are easier to obtain, are used to predict the number of rings which determines the age. Characteristics: 4177 cases, 8 attributes (1 nominal, 7 continuous).

4.3 Result analysis

The Tables 2, 3 and 4 present the errors from different sets of rules. This evaluation is done using the proposed method described previously. The Single column is the error of the unique regressor obtained from the 75% of the database. The 0% column presents the error of the integrated rule set, i.e., all the rules generated by the 10-fold cross-validation. The columns 5%, 10%, 25% and 50% are the errors of the integrated rule set with the elimination of the 5%, 10%, 25% and 50% of the worst rules, respectively. All the errors are calculated with the validation set, that is 25% of the database. Each table shows the results from different databases: Auto-Mpg, Housing and Abalone. Each line of the table represents an algorithm: Cubist, RT and $\mathcal{M}5$.

It should be noted that the algorithms RT and $\mathcal{M}5$ generate regression trees (instead of regression rules). These regression trees are converted to regression rules in our common syntax. So the evaluation of the single and 0% column do not use the default rule.

In Table 2 are present all results obtained from Auto-Mpg database. We can note that the error decreases when all rules are integrated. With the rules elimination, we also have in some cases a gain in precision. This can be explained by the fact that to predict new examples, when there is a great number of rules, the chance that the example is not covered by any rule is small, i.e., more chance to use the average of different rules that cover the example, and less chance to use the proximity function for uncovered examples.

However, using MAD as metric to exclude the worst rules, the error increases considerably. This increase is approximately 70% for RT and 120% for $\mathcal{M}5$ when 50% of the rules are eliminated. The trend is quite different when the coverage measure is used as metric to eliminate the worst rules. In almost all cases the error decreases or stays near to the error in the 0% column. Only in a few cases the error increases slightly when more rules are eliminated.

Most of errors for the Housing database, shown in Table 3, are better than the single rule set. Also in this database the integrated rule set presents error smaller

Table 2: Results for Auto-Mpg database.

	Single	0%	Cov				MAD			
			5%	10%	25%	50%	5%	10%	25%	50%
Cubist	8.10	6.44	7.01	6.95	8.2	8.4	6.54	7.01	7.33	7.73
RT	6.92	6.68	6.64	6.60	6.50	6.37	6.77	6.87	8.59	11.39
M5	6.5	6.5	6.5	6.4	6.2	6.6	6.4	7.2	10.2	14.19

Table 3: Results for Housing database.

	Single	0%	Cov				MAD			
			5%	10%	25%	50%	5%	10%	25%	50%
Cubist	9.06	6.30	5.72	5.82	6.31	11.16	5.54	6.08	6.25	5.72
RT	4.06	4.17	3.96	3.86	3.80	4.2	4.20	4.25	3.92	3.39
M5	5.2	4.5	5.1	5.4	5.3	5.3	5.2	5.2	4.9	4.2

Table 4: Results for Abalone database.

	Single	0%	Cov				MAD			
			5%	10%	25%	50%	5%	10%	25%	50%
Cubist	1.5	1.47	1.47	1.47	1.89	2.4	1.47	1.47	1.46	1.48
RT	1.6	1.54	1.65	1.65	1.71	1.85	1.54	1.59	1.71	1.89
M5	1.6	1.5	1.6	1.6	1.6	1.7	1.6	1.7	1.9	2.1

than the single rule set. Only in one case (Cubist with 50% of elimination) the error had a large increase, approximately 77%.

Table 4 shows the errors for the Abalone database. The best results are in the 0% column that is the integrated rules. The rule elimination methods had similar or worse errors than the integrated rule set.

5 Conclusion

One of the Data Mining topic that have been receiving great attention for several researchers is the performance and quality evaluation of the obtained knowledge, especially the interestingness and comprehensibility.

Besides, another topic involves the integration of the knowledge obtained from one or more regressors, still considering the quality and performance of this knowledge, and the robustness and consistence.

The process of knowledge integration consists of gathering different knowledge, generated from different samples of the same database, in a unique knowledge base, that is more complete than each base in separate. To evaluate the knowledge from a single regressor or the integrated knowledge, the performance and quality will be considered. Due to the great number of patterns, it is difficult for the user to understand them and to identify which are interesting.

In this article, experiments were run with three databases retreats of the UCI Repository and three algorithms. The obtained results are encouraging and show that our approach to integrate regression rules is promising. In all cases, when the rules are integrated in a single rule set, the errors are smaller than a single execution of the algorithm.

In order to define the rule elimination strategy we take into account the MAD and coverage measures for regression rules. Our contribution in this paper is that our proposed approach, as a way to integrate rules in a final rules set, increases the precision. Also, we can increase the model comprehensibility eliminating the worst rules.

Acknowledgment

This research is partially supported by Brazilian Research Council FAPESP.

References

- [1] Sahar, S., Interestingness via what is not interesting. *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, eds. S. Chaudhuri & D. Madigan, ACM Press: N.Y., pp. 332–336, 1999.
- [2] Weiss, S.M. & Indurkha, N., Rule-based machine learning methods for functional prediction. *Journal of Artificial Intelligence Research*, **3**, pp. 383–403, 1995.
- [3] Mitchell, T.M., *Machine Learning*. McGraw-Hill Series in Computer Science, 1997.
- [4] Lavrač, N., Flach, P. & Zupan, R., Rule evaluation measures: A unifying view. *Proceedings of the Ninth International Workshop on Inductive Logic Programming (ILP-99)*, eds. S. Dzeroski & P. Flach, Springer-Verlag, volume 1634, pp. 74–185, 1999. Lecture Notes in Artificial Intelligence.
- [5] Pugliesi, J.B. & Rezende, S.O., Error measures for regression rules. *Anais do XXIII Congresso da Sociedade Brasileira de Computação – IV Encontro Nacional de Inteligência Artificial (ENIA)*, eds. R. de Oliveira Anido & P.C. Masiero, SBC – Campinas, volume VII, pp. 517–526, 2003. (Portuguese).
- [6] Torgo, L., Data fitting with rule-based regression. *Proceedings of the 2nd International Workshop on Artificial Intelligence Techniques (AIT'95)*, J. Zizka & P. Brazdil: Brno, Czech Republic, pp. 223–232, 1995.

- [7] Merz, C.J., *Classification and Regression by Combining Models*. Ph.D. thesis, University of California, Irvine, 1998.
- [8] Torgo, L.F.R.A., *Inductive Learning of Tree-Based Regression Models*. Ph.D. thesis, Faculdade de Ciências da Universidade do Porto, 1999.
- [9] Hilderman, R. & Hamilton, H., Evaluation of interestingness measures for ranking discovered knowledge. *Proceedings of the 5th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'01)*, Hong Kong, pp. 247–259, 2001.
- [10] Netz, A., Chaudhuri, S., Bernhardt, J. & Fayyad, U.M., Integration of data mining with database technology. *VLDB 2000, Proceedings of 26th International Conference on Very Large Data Bases, September 10-14, 2000, Cairo, Egypt*, eds. A.E. Abbadi, M.L. Brodie, S. Chakravarthy, U. Dayal, N. Kamel, G. Schlageter & K.Y. Whang, Morgan Kaufmann, pp. 719–722, 2000.
- [11] Liu, B., Jsu, W., Ma, Y. & Chen, S., Mining interesting knowledge using DM-II. *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, eds. S. Chaudhuri & D. Madigan, ACM Press: N.Y., pp. 430–434, 1999.
- [12] Freitas, A.A., A multi-criteria approach for the evaluation of rule interestingness. *Proceedings of International Conference on Data Mining*, pp. 7–20, 1998.
- [13] Brazdil, P. & Torgo, L., Knowledge acquisition via knowledge integration. *Current Trends in Knowledge Acquisition*, eds. B.J. Wielinga, J. Boose, B. Gaines, G. Schreiber & S.M. Van, IOS Press, 1990.
- [14] Rulequest-Research, Rulequest research – Data mining tools, 2000. <http://www.rulequest.com/cubist-info.html>.
- [15] Torgo, L., Rt 4.1 user's manual, 2001. http://www.ncc.up.pt/~ltorgo/RT/rt_manual.pdf.
- [16] Quinlan, J.R., Learning with continuous classes. *Proc. Australian Joint Conference on Artificial Intelligence*, World Scientific, pp. 343–348, 1992.
- [17] Witten, I.H. & Frank, E., *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann: San Francisco, California, p. 416, 1999.
- [18] Blake, C.L. & Merz, C.J., UCI repository of machine learning databases, 1998. <http://www.ics.uci.edu/~mllearn/MLRepository.html>.