Proceedings of the International Conference on Information and Automation, December 15-18, 2005, Colombo, Sri Lanka.

Learning Predictive Models from Small Sets of Dirty Data

Ashwin Tengli, Artur Dubrawski and Lujie Chen
The Robotics Institute, School of Computer Science
Carnegie Mellon University, Pittsburgh, PA 15213
Email: {tengli,awd,lujiec}@cs.cmu.edu

Abstract— **This paper introduces robust predictive rule lists, new structures which combine ideas of decision lists and model trees with robust statistics. A predictive rule list is an ordered if-then-else list of rules, whose consequents are robust predictive models and the antecedents are conditions of their use. We illustrate the utility of the concept using a selection of problems typically approached with multiple linear regression. Empirical results obtained so far reveal features which may be especially appealing in practical applications: identified outliers can be avoided, instead of causing forceful elimination of potentially valid information; small sets of dirty data can be effectively addressed; resulting models are intuitive and easy to interpret. The presented approach tends to be beneficial when relatively high dimensional data comes in short supply, and when it contains substantial amount of non-ignorable measurement errors.**

I. Introduction

In many real world applications data acquisition and maintenance turns out to be expensive, imposing substantial limits on the quality and quantity of the available data supply. Examples range from industrial manufacturing scenarios in which the ability to conduct experiments and to collect measurements is limited due to ongoing activity on the production lines, to marketing, social sciences or bio-surveillance scenarios when attainable information is limited and sparse due to the nature of its source. Such data sets often contain anomalies like outliers, presence of multiple structures and missing values. If overlooked, such anomalies may seriously limit reliability of the attainable models.

Outliers have been informally defined as observations in a data set which appear to be inconsistent with the remainder of that set of data [1], and also as observations which deviate so much from the bulk of data so as to arouse suspicions that they were generated by a different mechanism [6]. They could be caused by influential measurement errors or produced by a qualitatively different than expected underlying processes. Dealing with outliers is the domain of robust statistics. A typical approach to robust modeling is a two-step procedure consisting of: (i) Identifying outliers; and (ii) Fitting a (standard, non-robust) model to the subset of the available data obtained after ignoring the outliers [14]. This would be a reasonable approach when applied to large data sets where removing outliers would not cause significant loss of data. However, discarding outliers may make the models overly nonchalant when applied to small data sets. There is a need for robust yet data-savvy approaches to building predictive models from small sets of dirty data.

The method presented in this paper attempts to exploit the structure of data in a way that overcomes problems often encountered when applying traditional modeling techniques mentioned above. A data point may look like an outlier in the context of some specific predictor features, but it may be perfectly usable in the sense of other input dimensions. Such nature of data allows us to build hierarchical list of regression models, where the lower level models are built on data points discarded as outliers by models above it, by using a different combination of predictor variables than the ones used above. This hierarchy of component models allows for recycling of the data points which in the context of the top level rules would have been discarded as outliers.

The hierarchy of predictive models can be looked at as a familiar sequence of the if-then-else list of rules similar to the decision lists originally developed by Rivest for classification tasks [13]. The qualifier for each rule is a classifier testing its applicability and the consequent of each rule is a regression model built on a subset of the predictor variables. The rule construction is driven by presence of outliers. Models at each level in the rule list are optimized for predicting the non-outlier points at that level and the outliers are passed on to the lower level to create a new model. The qualifiers for each rule are trained to classify non-outlier and outlier points at that level. The approach presented in this paper is applicable to all kinds of predictive modeling problems, but in order to simplify the explanation, here we focus on multiple linear regression tasks.

In the remainder of the paper we very briefly review relevant published work, provide details of the algorithm and present results of the experimental evaluation of the concept on a few real-life data sets and some synthetic data. Those results indicate that robust hierarchical predictive lists

enable better use of the available training data by reducing losses due to labeling points as outliers. They also indicate the ability of our method to achieve significantly higher predictive accuracies than a popular alternative tree based approach, when data is corrupt by outliers, without loss of simplicity or ease of interpretability of the built models. These results contribute to making the presented robust predictive rule list approach an attractive choice for practicing analysts facing "dirty" and relatively small multi-dimensional sets of data.

II. Related Work

The approach presented in this paper is more or less closely related to a range of previously published work, especially in the area of structural modeling (decision and model trees, and lists). Regression with tree-like structures called CART was first proposed in [2]. The idea was then extended by [11], [12] and [9] to use multivariate linear functions in the leaves. Later work in this domain has included use of other regression models such as kernel methods in the leaves [16],[17] and scaleable methods for model tree construction [3],[4],[10]. The commercial software Cubist [19], used for comparison purposes in our evaluation, is based on M5, a model tree algorithm developed by Quinlan [11],[12]. The motivation for using tree like structure is to create simple understandable models which can approximate complex data by local predictions. Construction methods for model trees usually rely on recursive greedy partitioning approaches splitting the data into mutually exclusive sets. The locality of the predictions is defined by the hypothesis behind the partitioning method. Those approaches however do not explicitly attempt to identify and exploit outliers present in the dirty data in order to avoid their undesirable influence. Instead they are directed at reducing metrics such as the local mean, local median or local sum-squared errors. They reveal certain ability to represent non-linearity in data, but that does not make them robust towards outliers. Rule-based regression lists having predicate-based rules on the left-hand-side and constant predictions of the output value on the right-hand-side were studied in [18]. This method utilized the inherent advantages of decision lists [13], but did not try to separate outliers in the predictor variables. Instead, its partitioning algorithm was solely based on the output attribute.

The partitioning method proposed in this paper is explicitly based on outliers in the data. The top level rule fits as much data as it can and then passes the remainder (data points which were identified as outliers by the robust fit procedure at the current level) to the lower level rules. No demand on specific effect of the partitioning algorithm on the data like reduction of mean or standard deviation is required. Since each rule in the list covers a subset of data, the individual rules tend to be simple, with fewer attributes than a single model covering the whole data set at a comparable accuracy and reliability.

III. Overview

Section A below briefly discusses standard robust approaches to identifying outliers and describes the Least Median of Squares method used as the core modeling technique to illustrate the proposed approach. An overview of our structure of the hierarchical predictive models is presented in Section B. Section C describes the procedure for making predictions using the rule lists. Section D explains the mechanics of the training phase.

A. Underlying Robust Methods

Though outliers have been studied extensively in statistics and data mining community, their definition is somewhat ambiguous. An outlier is traditionally defined as an observation (or subset of observations) which appears to be inconsistent with the remainder of the set of data [1]. They are commonly believed to be extreme (or relatively extreme) observations in the data which may be anomalies or artifacts embodying unexpected knowledge about the underlying processes. Outliers may also occur due to errors in the data acquisition procedure or due to contamination of data by observations produced by processes governed by different than expected distributions. Hence, as the outliers deviate from the assumptions made about the source of data, they may significantly affect the generated model.

Dealing with outliers is the domain of Robust Statistics. Huber [8] defines robustness in statistics as insensitivity to small deviations from assumptions. Rousseeuw and Leroy [14] define regression outliers as observations that deviate from the linear relation followed by the majority of the data. The least squares method of solving linear regression is not robust to outliers. A detailed discussion on this can be found in [14]. Various methods [5],[7],[14] have been proposed for dealing with regression outliers. We have chosen a popular, standard LMS (Least Median of Squares) procedure implemented using PROGRESS algorithm, to illustrate the ideas presented in this paper. Its complete description can be found in [14].

In PROGRESS, robust regression fit is being sought via combinatorial optimization. At each step of the iterative procedure, a small sample of the data is selected (it is usually randomly picked unless the data set size allows for explicit enumeration of all possible samples of a given size) and the exact linear model is fit to it. Then, squared residuals for all data points available for training are being calculated and their median is used as the objective for minimization. The sample with the smallest median is used to indicate outliers (hence the name of the method: LMS). Data points with residuals beyond certain limits (estimated on the basis of the variance of residuals) are flagged out as outliers. The portion of data which passes the test is then used to obtain

the standard ordinary least squares (OLS) regression model which then becomes the resultant of the LMS procedure.

B. Structure

The robust predictive rule list model is an ordered sequence of rules which assumes the following form:

$$\text{IF}\,[c_1(g_1(\mathbf{x}_i)) == \text{TRUE}] \Rightarrow [\hat{y}_i = f_1(h_1(\mathbf{x}_i))]$$
$$\text{ELSE IF}\,[c_2(g_2(\mathbf{x}_i)) == \text{TRUE}] \Rightarrow [\hat{y}_i = f_2(h_2(\mathbf{x}_i))]$$
$$\ldots$$
$$\text{ELSE IF}\,[c_{K-1}(g_{K-1}(\mathbf{x}_i)) == \text{TRUE}] \Rightarrow [\hat{y}_i = f_{K-1}(h_{K-1}(\mathbf{x}_i))]$$
$$\text{ELSE}\,[\hat{y}_i = f_K(h_K(\mathbf{x}_i))]$$

Here c_k denotes rule applicability functions. They test whether specific constraints imposed on the given vector of predictors (the input vector), $\mathbf{x}_i$, are met. The first rule (counting from the top of the list) which satisfies its applicability condition is applied to predict the value of the output variable y_i. The prediction, $\hat{y}_i$, is then made using the corresponding model function, f_k. The model function operates on the specific subset of input features. The particular selection of these features, made for each of the rules individually, is determined by the selector function h_k. The applicability functions have their individual selector functions as well. They are denoted as g_k.

In general, the applicability functions may assume any reasonable form. In the experiments presented later in this paper we use them for one specific purpose: to predict whether the input vector $\mathbf{x}_i$ can be treated as outlier with respect to the corresponding predictor function or not. This can be accomplished by building a classifier which would be trained to tell apart inliers from outliers using the individual rule-specific subsets of features of $\mathbf{x}_i$'s.

The presented approach does not put restrictions on the form of the model functions. They can be of any type: parametric or non-parametric, statistically motivated or not, and they do not even have to be homogenous across the rule list. However, in order to simplify the discourse in this paper, we limit our choices to multiple linear regression models.

The model functions operate on subsets of input features. These functions in our particular case are learned from data using stepwise regression and the learning process determines the particular subset of features that are relevant for the particular rule k. The corresponding selector function, h_k, simply implements the selection determined during training. Typically, the selector functions for the model function and for the applicability condition of one rule are identical, $h_k \equiv g_k$, but that is not a general requirement of the proposed approach.

C. Making Predictions

When a new input vector $\mathbf{x}_i$ is presented to the system, the rules are tried sequentially starting from the top of the list. The first rule to satisfy its applicability criterion is selected and its model function is executed on the subset of $\mathbf{x}_i$'s features determined by the corresponding selector function h_k. That results in the prediction $\hat{y}_i$. In a sense, the proposed structure of the list can be interpreted as exclusive, because exactly only one rule will be executed for an individual input vector.

The last rule in the list does not have a specific applicability condition. It can be used in the case when all higher level rules failed their applicability tests. That may happen if the particular values of the features in the input vector $\mathbf{x}_i$ make it look like an outlier with respect to all of the models included in the higher levels of the list.

D. Training

The general concept of building robust predictive rule list structures from data is primarily based on a search through combinations of input features (and possibly through different available model types) for accurate models which satisfy certain criteria of minimal data coverage. For the purposes of this paper we constrain ourselves to multiple linear regression models. The rule lists are constructed by recursively adding a rule at a time to cover the remaining data points. The model for each rule can be obtained via a stepwise greedy feature selection procedure in which at each step one additional input feature is being added to the regression model. The newly added feature is selected among all other features not yet present in the model because it provides the largest improvement of some goodness of fit criterion such as R^2 [15] or a generalization score calculated via cross-validation. The goodness of fit score is calculated after isolating the outliers using LMS and then fitting the model on the remaining inliers. The complexity of the obtained models can be controlled using some stopping criteria. The classical choices include c_p and F statistics [15], or cross-validation. If the data is not high dimensional or if the component models of the predictive rule list hierarchy are required to be low-dimensional (e.g. due to a desired ease of interpretation), it may be feasible to execute an exhaustive search for the best subset of input features up to a certain size (this is often called the all-subset multiple regression [15]). In fact, in the experiments described in the next section, we use a combination of those two approaches: we begin model search by exhaustively testing all combinations of input features up to a certain subset size (equal 3 in the experiments described below) and then we follow on greedily adding one of the remaining features at a time until the optimal complexity is achieved.

The subsequent levels of the rule list are developed in the same way until all training data points are covered. At that stage, or earlier if we cannot find any model which would meet the minimum coverage criteria, we complete the

development of the hierarchy by adding a default rule to the bottom of it. The default rule implements a function that ignores input features. Typically, a mean or a median of the output attribute are used as predictions of such default rule. In our experiments we have been using the mean of y_i calculated across all available data in the training set.

The complete hierarchy is then subjected to k-fold cross-validation test. We use it to empirically select the allowed complexity level (defined as the highest allowable number of terms in the component regression equations) which leads to the optimal generalization abilities of the predictive rule list.

Practical deployment of the system requires equipping each rule with a binary classifier which would be able to learn to tell apart outliers from inliers among test samples, based on training data. The trained classifier plays the role of the applicability criterion for its rule. We must humbly admit that the fairly simple classifiers we tried so far (such as 5-nearest neighbor) have not revealed a consistently better performance than the alternative used for comparisons in this paper. For the experiments described below, in order to determine the upper limit of the attainable performance, we plugged in an "ideal" classifier – a surrogate made to perfectly tell outliers from inliers by looking at the actual output values of the test cases. Adopting a realistic classifier for practical applicability of the proposed method is the primary objective of our ongoing research.

It is worth noting that the above described learning algorithm highlights key properties of the proposed methodology: (1) Ability to efficiently deal with relatively small sets of dirty data with relatively many predictor variables; (2) The ability to recycle instances dropped at higher levels of the list by the models included at the lower levels; (3) Attainability of hierarchies composed of easy to interpret component models. The following section summarizes the experimental results we have obtained so far.

IV. Experiments

Experiments were conducted to verify the advantages and limitations of the robust predictive rule list models (denoted as "PRL" in the graphs and tables below) compared to standard non-hierarchical robust regression ("LMS") which simply discards outliers, and Cubist [19]: a commercial implementation of Quinlan's model tree algorithm [11],[12], which is not specifically designed to handle dirty data.

We used four real-world data sets and two families of synthetic data. The "industrial" data [110 records, 16 inputs] is similar in flavor to the sets which originally motivated us to pursue this research[1]. The "boston" [506 records, 12 inputs] (predicting housing value in Boston area), "wisconsin" [194 records, 32 inputs] (cancer survival data) and "pyrim" [74 records, 27 inputs] (drug design) are publicly accessible benchmark sets widely used within machine learning community. In each case, we built models to predict a single output using only continuous inputs. Since the data sets used in our experiments were relatively small, each model in the list was built on all the inlier points (with respect to that model) in the data set.

Each of the synthetic data sets consists of 100 records and 10 predictor attributes, and 1 output. They are labeled as "cdata_X" and "uscdata_X", where X denotes the percentage of entries corrupted to become outliers (so that "cdata_1" has about 1% and "uscdata_30" 30% of its contents distorted). These data sets are derived from the same original table filled with numbers randomly sampled from the standard Gaussian distribution. The output column entries are computed from the corresponding inputs and assumed linear models, assigning a constant coefficient to each input dimension; plus additive Gaussian noise component. Then the table entries are subjected to corruption: "cdata_X" set entries are selected to be corrupted uniformly, while "uscdata_X" family is being corrupted in a non-uniform, systematic way.

Table I.
Comparison of scores obtained with plain LMS and robust PRL(idealized classifier)

Local	LMS			PRL (ideal classifier)			Change		
	Cover	MSE	R^2	Cover	MSE	R^2	Cover	MSE	R^2
boston	88%	10.14	0.84	97%	11.60	0.84	**8%**	14%	-1%
wisconsin	92%	953.79	0.16	99%	1,025.36	0.13	**7%**	8%	-20%
pyrim	77%	0.00	0.73	97%	0.00	0.69	**20%**	22%	-6%
industrial	51%	5.22	0.47	95%	7.41	0.44	**44%**	42%	-6%
cdata_1	88%	2.72	0.95	99%	6.01	0.89	**11%**	121%	-7%
cdata_3	86%	5.32	0.90	100%	9.07	0.83	**14%**	70%	-8%
cdata_5	87%	3.52	0.93	99%	5.94	0.89	**12%**	69%	-5%
cdata_10	82%	19.10	0.64	99%	24.70	0.53	**17%**	29%	-16%
cdata_20	79%	40.79	0.26	99%	41.25	0.22	**20%**	1%	-14%
cdata_30	88%	50.17	(0.06)	99%	55.52	(0.06)	**11%**	11%	-11%
uscdata_1	90%	2.00	0.96	99%	4.57	0.91	**9%**	128%	-5%
uscdata_3	88%	2.95	0.94	99%	6.29	0.88	**11%**	113%	-6%
uscdata_5	86%	3.77	0.93	98%	8.75	0.83	**12%**	132%	-10%
uscdata_10	82%	5.27	0.90	99%	11.59	0.78	**17%**	120%	-14%
uscdata_20	77%	18.40	0.63	99%	24.45	0.53	**22%**	33%	-15%
uscdata_30	77%	20.55	0.64	99%	25.68	0.51	**22%**	25%	-20%

Global	LMS		PRL		Change	
	MSE	R^2	MSE	R^2	MSE	R^2
boston	37.14	0.56	27.14	0.68	**-27%**	**21%**
wisconsin	1,016.50	0.14	1,032.45	0.13	2%	-9%
pyrim	0.01	0.43	0.01	0.56	**-23%**	**30%**
industrial	13.34	0.20	10.87	0.35	**-19%**	**74%**
cdata_1	8.90	0.83	6.08	0.88	**-32%**	**6%**
cdata_3	10.16	0.81	9.07	0.83	**-11%**	**3%**
cdata_5	8.84	0.83	6.17	0.88	**-30%**	**6%**
cdata_10	25.05	0.53	24.72	0.53	**-1%**	**1%**
cdata_20	41.87	0.21	41.31	0.22	**-1%**	**5%**
cdata_30	55.64	(0.05)	55.81	(0.06)	0%	**6%**
uscdata_1	7.45	0.86	5.02	0.91	**-33%**	**5%**
uscdata_3	11.05	0.79	6.47	0.88	**-41%**	**11%**
uscdata_5	9.84	0.81	9.17	0.83	**-7%**	**2%**
uscdata_10	12.97	0.75	12.10	0.77	**-7%**	**2%**
uscdata_20	29.06	0.45	25.18	0.52	**-13%**	**16%**
uscdata_30	24.26	0.54	25.88	0.51	7%	-6%

The following metrics were used to evaluate the performance of the systems:

1. R^2 and MSE averaged over 10 independent runs of 10-fold cross-validation procedures (to evaluate predictive

[1] The "industrial" data set is available for download upon request.

accuracy on testing data).
2. Extent of coverage of data: fraction of data points used in the model vs. total number of data points.
3. Number of unique inputs used by the model hierarchy (in order to quantify its cumulative complexity).
4. p-values of the paired t-tests of significance (at the level of 5%) for differences between results obtained through ten independent 10-fold cross-validation runs.

We differentiate between the sets of results: obtained for full prediction mode scores ("Global") where all data points, including known outliers, are used to evaluate performance, (it is also called the "full prediction" mode); and identification mode scores ("Local") where known outliers are ignored in calculating the results (note that this is the standard way of doing business in the robust regression world). Unless otherwise stated, a score refers to the full prediction mode.

Table I summarizes a comparative evaluation of plain LMS vs. robust predictive rule list. With respect to utilizing available data PRL overcomes LMS coverage by 7 to 44% in the identification mode (the higher increases correspond to more outlier-prone data sets) and with respect to accuracy in the full prediction mode (when all test observations must be given a prediction). The observed accuracies are across the board worse than in the local case. As expected; predictive rule list does better job at full prediction than LMS on most data sets – on highly corrupted ("cdata_30", "uscdata_30") and fundamentally challenging ("wisconsin") data sets it do not reveal improvements. The observed benefits are more substantial on data with systematically corrupted entries, as intended and expected.

Table II.
Comparison of accuracy and cumulative complexities of the models obtained with Cubist and robust PRL (idealized classifier).

dataset	Cubist		PRL (w/ idealized classifier)		
	MSE	NUI	MSE	NUI	p-value
boston	**12.466**	10	27.138	8	0.00% -
pyrim	0.015	6	**0.007**	11	0.00% +
industrial	23.723	6	**10.874**	10	0.00% +
wisconsin	1,262.638	9	**1,032.453**	11	0.01% +
cdata_1	6.779	9	**6.083**	10	17.90%
cdata_3	**8.824**	9	9.067	9	36.65%
cdata_5	**6.043**	9	6.171	9	40.49%
cdata_10	58.625	7	**24.716**	9	0.00% +
cdata_20	47.599	6	**41.310**	7	1.32% +
cdata_30	59.062	1	**55.806**	7	0.77% +
uscdata_1	5.148	9	**5.018**	10	42.13%
uscdata_3	30.263	8	**6.471**	10	0.00% +
uscdata_5	33.969	6	**9.174**	10	0.00% +
uscdata_10	42.629	7	**12.102**	10	0.00% +
uscdata_20	47.312	6	**25.181**	9	0.00% +
uscdata_30	38.652	5	**25.877**	8	0.00% +

Table II summarizes the results of comparison of performance and complexity of the robust predictive rule list models obtained with use of the idealized classifier against results obtained with Cubist. The robust PRL significantly outperformed Cubist when dealing with the substantially corrupted synthetic data. A couple of the opposite results observed at uniformly altered data were not significant. Cubist does a much better job than predictive rule list on "boston" data. This data set is the largest of all used in our experiments and it seems to be fairly well covered with just one robust model (cf. high scores for plain LMS reported Table I above). It does not have many available inputs to choose from and it does not seem to contain many outliers, hence it is more difficult to excel at for robust predictive rule lists than it is for the tree-like hierarchy of Cubist which allows for more accurate representation of underlying non-linearity. The data set of our primary interest "industrial" as well as two remaining real-world data sets apparently suit the robust predictive rule list model better than Cubist, due to their limited size ("pyrim") and known contamination ("industrial"). It is worth noting that the complexities of models obtained with predictive rule list and Cubist do not seem to vary much, although whenever robust predictive rule list significantly outperforms Cubist on accuracy, it also tends to use larger number of unique inputs.

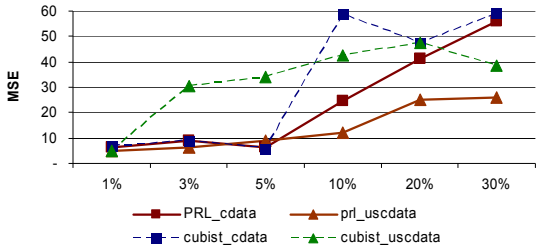

Fig.1. MSE score for PRL vs Cubist under different corruption scenarios.

Figure 1 depicts a comparison of predictive error scores achieved by Cubist and predictive rule list on synthetic data under two corruption scenarios (systematic, labeled "uscdata" and random, "cdata"). Very little corruption does not seem to create much difference, and the most substantial effects can be observed on systematically altered data, where the robust predictive rule list approach leads consistently by a wide margin.

Table III.
Comparison of Cubist vs. robust PRL using the k-nearest-neighbor classifier.

data set	Cubist MSE	PRL (w/ classifier) MSE	PRL vs Cubist p-value
cdata_1	6.779	**4.424**	0.01% +
cdata_3	8.824	**6.589**	0.00% +
cdata_5	6.043	**3.401**	0.00% +
cdata_10	**58.625**	60.631	9.78%
cdata_20	47.599	**47.146**	40.60%
cdata_30	59.062	**58.925**	45.91%
uscdata_1	5.148	**2.335**	0.00% +
uscdata_3	30.263	**27.577**	0.10% +
uscdata_5	33.969	**32.626**	15.28%
uscdata_10	**42.629**	45.033	12.86%
uscdata_20	47.312	**43.010**	2.66%
uscdata_30	38.652	**38.476**	39.71%

Table III summarizes the results of testing the robust predictive rule list equipped with a realistic classifier (5-nearest-neighbor) on synthetic data. It still outperforms Cubist most of the time, sometimes significantly, and when it loses, the difference is not significant. Sometimes, using a less accurate classifier may lead to better overall scores (as seen in Table III) for data sets with low degree of corruption. The naïve classifier tends to create more piecewise rules due to lower accuracy. This can result in a better fit in the presence of the additive Gaussian noise in the output attribute. This advantage is lost at higher degrees of corruption in input attributes. A detailed study of this phenomenon is included in the scope of continuation research hinted on in the next section.

V. Conclusion

We introduced and empirically evaluated the concept of robust predictive rule lists which extends applicability of model-tree and regression-list family of methods towards dealing with "dirty" data. The concept has been experimentally verified to reveal features which may be especially appealing in practical applications.

Firstly, the proposed hierarchical model can make a better use of training data contaminated with outliers than fundamental non-hierarchical robust methods. It is achieved by passing the data points which are considered outliers at one level of the hierarchy to the subsequent levels where they can be utilized as inliers by one of the models, instead of discarding them right away. That is possible if the number of available distinct input features is sufficiently large to enable selections of their subsets leading to varying patterns of data coverage along the model hierarchy.

Secondly, albeit it is not a unique characteristic of the proposed method, the resulting models are user-friendly: components are familiar to most end-users, they are easy to interpret, and the hierarchy is an intuitive if-then-else list of rules.

Most importantly, the robust predictive rule lists enable meaningful modeling of relatively small sets of dirty data which have a relatively high number of available predictor attributes. In our opinion, analysis of such data sets has not been adequately addressed in the available literature yet, although we regularly encounter them in our practice.

Although it is not in the scope of this paper, we would like to mention that it is very easy to make the proposed method robust against missing values without having to make specific assumptions about the nature of missingness and without having to sacrifice precious data points. We have already obtained favorable results of applying the robust predictive rule list approach to practical problems which combined issues of large errors of measurement and missing data.

We plan to continue working on gathering additional experimental evidence of the useful features and limitations of the proposed approach. We also continue work on the issue of selecting classifiers for rule applicability functions, and on meaningfully combining them into the model search process. Another important topic of the ongoing research is computational efficiency of the learning phase. It suffers mostly due to high costs of the used robust regression procedure. We are working on representational ideas which should lead to substantial savings.

We believe that the presented method of robust predictive rule list is an appealing tool for practical applications of predictive analytics because it reduces requirements on the quantity and quality of data used for training. The method proposed in this paper can extend applicability of predictive modeling towards scenarios when data comes in short supply, its quality is poor, and its reconciliation or cleaning is costly or impossible.

References

[1] Barnett V. and Lewis T. Outliers in Statistical Data. John Wiley, New York, 3rd edition, 1994.

[2] Breiman L., Friedman J. Olshen R. and Stone J. *Classification and Regression Trees*. Wadsworth and Brooks. 1984.

[3] Chaudhuri P., Huang M., W. Loh, and R. Yao, "Piecewise-Polynomial Regression Trees," Statistica Sinica, vol. 4, pp. 143-167, 1994.

[4] Dobra A. and Gehrke J. E. "Secret: A Scalable Linear Regression Tree Algorithm," Proc. Eighth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, 2002.

[5] Fischler M.A. and Bolles R.C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Comm. of ACM, 24(6):381-395, 1981.

[6] Hawkins D. Identification of Outliers. Chapman and Hall, 1980.

[7] Huber P. J., Robust estimation of a location parameter, Ann. Math. Statist.35: 73-101. 1964.

[8] Huber P.J. Robust Statistics. Wiley, New York, 1981.

[9] Karalic A. "Linear Regression in Regression Tree Leaves," Proc. Int'l School for Synthesis of Expert Knowledge, pp. 151-163, 1992.

[10] Loh W. "Regression Trees with Unbiased Variable Selection and Interaction Detection," Statistica Sinica, vol. 12, pp. 361-386, 2002.

[11] Quinlan J. R. "Learning with Continuous Classes," Proc. Fifth Australian Joint Conf. Artificial Intelligence, pp. 343-348, 1992.

[12] Quinlan J. R. Combining Instance-based and Model-based Learning. Proceedings of the 10th ICML. Morgan Kaufmann. 1993.

[13] Rivest R. L. Learning Decision Lists. *Machine Learning* 2:229-246. 1987.

[14] Rousseeuw P. J., and Leroy A. M. *Robust Regression and Outlier Detection*. New York: Wiley. 1987.

[15] Ryan T. P. *Modern Regression Methods*. New York: Wiley. 1997.

[16] Torgo L. "Functional Models for Regression Tree Leaves," Proc. 14th Int'l Conf. Machine Learning, D. Fisher, ed., pp. 385-393, 1997.

[17] Torgo L. Kernel Regression Trees, Poster papers of the European Conference on Machine Learning (ECML-97), Internal Report of Faculty of Informatics and Statistics, University of Economics, Prague, ISBN:80-7079-368-6, 1997.

[18] Weiss S. and Indurkhya, N. Rule-based machine learning methods for functional prediction. *Journal of Artificial Intelligence Research,* 3:383-403, 1995.

[19] Cubist commercial data mining software: http://www.rulequest.com