Predicting Rare Extreme Values

Luis Torgo and Rita Ribeiro
LIACC-NIAAD, University of Porto
R. de Ceuta, 118, 6., 4050-190 Porto, Portugal
[ltorgo,rita]@liacc.up.pt

January 9, 2006

Abstract

This paper addresses the problem of rare extreme values prediction. Modeling extreme data is very important in several application domains, like for instance finance, meteorology, ecology, etc. Our target applications have as main objective to be able to anticipate extreme values of a continuous variable. The main distinguishing feature of these applications resides on the fact that these values are rare. Any prediction model is obtained by some sort of search process guided by a pre-specified evaluation criterion. In this work we argue against the use of standard criteria for evaluating regression models in the context of our target applications. We propose a new predictive performance metric for this class of problems that our experiments show to perform better in distinguishing models that are more accurate at rare extreme values. This new evaluation metric can be used as the basis for developing better models in terms of rare extreme values prediction.

1 Introduction

In several applications the main focus of interest is a small proportion of the available data. These unusual cases have a large importance, and as such, anticipating them is a critical task for these domains. An example of such application is the prediction of the future returns of a stock. Unusually high (low) returns are rare, but they are the most interesting values for investors and thus they should be the target of any financial prediction model. In other domains, like for instance several ecological applications, we face similar problems where we want an early forecast of dangerous and fortunately rare extreme values on some biological indicators, so that preventive actions can be carried out.

A related problem has been receiving great attention in the data mining community. This problem is the construction of classification models based on samples with unbalanced classes distribution (e.g. [4, 5, 7]). In most of these applications the class that is more interesting (whose prediction errors are more costly) is usually the less frequent. Facing these tasks has two consequences:- we need to tune up our models to bias their search for the more interesting classes; and we need to use performance metrics that better reflect our preference criteria. Concerning the former two main approaches have been taken: modification of the distribution of the training set; and use of cost-sensitive learning methods (e.g. [3]). Regarding the first approach most work revolves around the notions of under-sampling the majority class (e.g. [6]), or the notion of over-sampling the minority class. With respect to the performance metrics, Provost et al. [7] have shown the inadequacy of the common classification accuracy for these tasks, because of its equal misclassification costs, suggesting alternatives based on ROC analysis.

Predicting extreme values of a continuous variable can be handled through a classification approach (e.g. [10]). This would have the advantage of using all work that has been around in the areas of unbalanced classification problems and evaluation under differentiated misclassification costs. In order to be able to use classification methods in our applications, we would need to discretize the continuous variable and also to provide misclassification costs. In this work we will argue that this approach has several drawbacks. Based on these arguments we will propose an alternative method of evaluating models for rare extreme values prediction in the context of regression approaches.

This paper is organized as follows. Section 2 formalizes the problem of extreme values prediction and exemplifies the problems it brings to standard regression algorithms. In Section 3 we describe how this problem could be addressed using classification algorithms and provide arguments against this approach. Section 4 describes some possible ways of addressing these problems within a regression setup. On Section 5 our proposal is presented. We then present an empirical evaluation of its merits in Section 6, and finish with the conclusions in Section 7.

2 Problem Description

The problem of rare extreme values prediction is a particular case of multiple regression where a target continuous variable Y is being modeled using a set of predictor or input variables $X_1, X_2, \cdots, X_p$. Models are obtained using a

sample of cases of the underlying unknown regression function f, usually called the training set, $D = \{\langle \mathbf{x}_i, y_i \rangle\}_{i=1}^{n}$. A model is an approximation, $\widehat{f}_\beta$ of the unknown function f, where β are the model parameters estimated using the training data D. Any modeling method tries to find the model parameters β that minimize an error function over the training data. Standard used functions are the Mean Squared Error, $MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \widehat{f}_\beta(\mathbf{x}_i)\right)^2$, or the Mean Absolute Deviation, $MAD = \frac{1}{n}\sum_{i=1}^{n}|y_i - \widehat{f}_\beta(\mathbf{x}_i)|$.

The particular applications we are interested have a distribution of the Y target variable that has a normal-like shape but with very heavy tails, i.e. very rare extreme values around the common average values. Moreover, they share the requirement that the prediction error at these tails should be more penalized. As we will see in our experiments, evaluation metrics like the *MSE* and the *MAD*, do not put sufficient emphasis on the errors at extreme values of the variable. This is particularly notorious when these extremes are rare when compared to the less interesting (for our target applications) "normal" values.

3 Classification Approaches

The problems we have described in the previous section could be addressed using a classification approach. For that to be possible we would have to reformulate the problem as a classification task. This could be accomplished through a process of discretization of the target variable Y, and also with the help of misclassification costs to cope with the implicit ordering among the labels resulting from this process (e.g. [10]). The resulting classification problem will inevitably have an unbalanced class distribution if the discretization process takes into account the goals of the original problem, i.e. to be accurate at predicting extreme values. Discretizing a continuous variable involves creating a set of k bins, the resulting classes. Creating these bins requires deciding upon the cut points that divide each bin. All continuous values that fall inside a bin will be labelled as belonging to the same class. This process has two key parameters: the number of bins to create, and the criterion used to select the possible cut points between bins. The choices for these parameters are obviously constrained by our application goals and are thus dependent on domain knowledge.

For our target applications, a possible solution is to use three bins (classes): one for the low extreme values; another for the high extremes; and finally another for the "normal" values. This transformation requires the specification of thresholds that guide the process of discretization, as seen below,

$$\text{class}(Y) = \begin{cases} extr_L & \text{if } Y < thr_L \\ indif & \text{if } thr_L \leq Y \leq thr_H \\ extr_H & \text{if } Y > thr_H \end{cases} \quad (1)$$

The threshold values are critical and we will argue that they are the main drawback of using a classification approach in this type of problems. They create a crisp and artificial division between the values of the continuous variable Y.

There are other alternatives in terms of deciding how many classes to use for discretizing a continuous variable. In particular, for some applications it may be interesting to distinguish among the "normal" (indifferent) values depending on their signal. For instance, in a typical stock forecasting application one is

interested in predicting extreme low returns (negative) and extreme high returns (positive). However, it is also relevant to distinguish between indifferent negative and positive returns. Suppose we predict a high extreme value. This could lead to an investment decision (for instance buy some stocks). If the actual value is indifferent, but still positive, the cost of the classification error is simply the fact that we have less profit than expected. However, the cost if the actual value is non-extreme but negative is higher, because not only we do not win what was expected but also we loose money. For these applications it makes more sense to use four classes: the usual extreme low and high, and indifferent negative and positive.

3.1 Why Classification Approaches Do Not Work?

Let us first focus on the 3 classes case. The discretization process outlined in Equation (1) requires the specification of the thresholds thr_L and thr_H. Assuming our domain knowledge enables us to come up with meaningful values for these thresholds, we still have to differentiate between the classification errors [10], if we want our model to penalize more a confusion between say an $extr_L$ case and an $extr_H$ case, than an $extr_L$ and an $indif$ case, for example. This means that we need to fill in the following cost matrix:

Table 1: 3-classes cost matrix.

		True		
		$extr_L$	$indif$	$extr_H$
Preds.	$extr_L$	$C_{L,L}$	$C_{L,I}$	$C_{L,H}$
	$indif$	$C_{I,L}$	$C_{I,I}$	$C_{I,H}$
	$extr_H$	$C_{H,L}$	$C_{H,I}$	$C_{H,H}$

Assigning these costs precisely is not a trivial task [7]. Still, the critical problem of this approach to extremes prediction lies on the definition of the threshold values. These thresholds may lead to intuitively wrong penalizations in terms of any error measure one can use based on the misclassification costs of Table 1. For instance, suppose $thr_L = -120$. A prediction of -119.99 for a true value of -121, would be considered as an error with cost $C_{I,L}$, while a prediction of say -150 would be considered a correct prediction and moreover with the same benefit of a perfect prediction of -121. This is clearly counter-intuitive in any application we can imagine, and it is caused by the use of a 0/1 loss function on a problem that is essentially metric. A possible solution to this problem could be to increase the number of classes so as to differentiate more between these situations. However, not only this solution would always be a coarse approximation of the continuous case, but also by increasing the number of classes we are turning an already unbalanced classification problem into an even harder task.

We argue that this is a major drawback of all classification-based approaches to the problem of predicting rare extreme values of a continuous variable. In our proposal (Section 5) we try to overcome this limitation by providing a means to smooth the division between classes. Another key drawback of these approaches lies on the fact that we loose granularity. In effect, using classification approaches we can only distinguish between extremes and non-extremes. This

means that all extremes are taken equally, which may not make sense in several applications where the degree of extremeness can lead to different actions.

4 Regression Approaches

Within the multiple regression setup described in Section 2, there are a few alternatives to the standard error measures (e.g. mean squared error or mean absolute deviation), which could be considered as more adequate to the prediction of extreme values.

One possible method for giving more weight to the errors on extreme values is to use higher powers of the difference between predicted and true values. For instance, instead of calculating the mean squared differences, we could use the following error measure,

$$Err_p(y, \hat{y}) = \frac{\sum (y - \hat{y})^p}{n} \tag{2}$$

where p is a positive integer.

The larger the value of p, the more penalized are errors on extreme values. This method is a possible solution to the extreme values prediction problem, which does not have the problems mentioned for classification approaches. One of the main drawbacks of this alternative is its insensivity to the domain characteristics. For instance, with this method it is not possible to penalize less a prediction of 5 for a true value of 3, than a prediction of 1 for a true value of -1. Both situations have the same absolute difference (2) and thus will have the same penalization, but this might be counter-intuitive in several application domains. Notice that this kind of differentiation is possible in the classification approaches described in Section 3, where we can fill in the cost matrices according to our application preferences.

Another alternative that is available in several modelling methods is to use case weights. Some algorithms allow the user to attach a weight to each observation of the training sample. Model parameters can then be obtained by minimizing a criterion that takes into account these weights,

$$\hat{f}_\beta(D) = \min_\beta \frac{\sum_{i \in D} w_i \cdot Err_i}{\sum_{i \in D} w_i} \tag{3}$$

where w_i is the weight associated with the case i, and Err_i is the error of the model in the case i.

If we use case weights that depend on the respective Y value being an extreme, then the obtained model is biased to correctly predict these extreme cases. The main drawback of this approach is that it only sees one side of the problem, the true values. In effect, this method does not try to avoid (or penalize) the cases where an extreme value is predicted by the model, but the truth value is "normal", i.e. false positives according to the classification terminology. This drawback stems from the fact that the weights are dependent solely on the true value of the cases, y_i, instead of being dependent on both y_i and $\hat{y}_i$, as it is the case for instance in the classification approaches, where the cost of errors depends on these two values (e.g. Table 1). Our proposal builds upon this idea by trying to eliminate this drawback through the use of a weight function that depends on both y_i and $\hat{y}_i$.

5 Our Proposal

The overall goal of this work is to have an evaluation metric that is biased towards valuating more the predictions of rare extreme values. Our proposal was developed with the following requirements in mind:

1. *The cost of a prediction error should depend on both the predicted and the true values, i.e. we should penalize both false positives and false negatives.*
   Motivation: In our target applications predictions will generally lead to actions. If the cost was only dependent on the true value of the target, we would not penalize a situation where a model predicts an extreme (thus leading to some action) for a true value that is not an extreme. In summary, we want to anticipate most of the extremes (i.e. have a high recall) and also to be accurate when we predict an extreme (i.e. have a high precision).

2. *The cost of the errors should vary smoothly (no crisp divisions between extremes and non-extremes).*
   Motivation: Defining crisp boundaries is difficult for the user and may lead to counter-intuitive penalizations as shown in the examples presented in Section 3.

3. *The method should have reasonable default costs (according to the overall goal) for cases where knowledge about the costs is not available.*
   Motivation: Filling the costs is sometimes difficult and/or precise information may not be available.

We propose an evaluation metric that is basically a weighted average of the errors along the lines of Equation (3). The key difference lies on the form of calculating the weights. We use a weight function that depends on both the true and predicted values, similarly to classification approaches described in Section 3. Compared to these approaches, we propose to use a smooth cost surface,$w(Y, \hat{Y})$, that can be seen as a continuous version of the cost matrix provided in Table 1 to avoid problems with crisp boundaries. Summarizing, our proposed Rare Extremes Error metric can be obtained by,

$$RExE = \frac{1}{n} \sum_{i=1}^{n} w(y_i, \hat{y}_i) \times L(y_i, \hat{y}_i) \tag{4}$$

where $L(y_i, \hat{y}_i)$ is a loss function that could be for instance the absolute deviation or the squared error.

In order to make the use of smooth cost surfaces practical we need to devise an easy way of specifying them. Our proposal consists of requiring the specification of the cost values at a small set of properly selected points and then using a function approximation method to interpolate the complete surface. The axes of the surface are the true, Y, and predicted, $\hat{Y}$, values of the target variable. The points selected for specifying the cost surface should be related to the most relevant areas of the surface. These are the areas of lower cost (the model accurately predicts and extreme as such), and of the worse performance (the model predicts an extreme high for a true extreme low, or vice versa). All other situations should have a cost between the costs associated to these two situations. Figure 1 provides a better idea of the type of surface we need to specify.

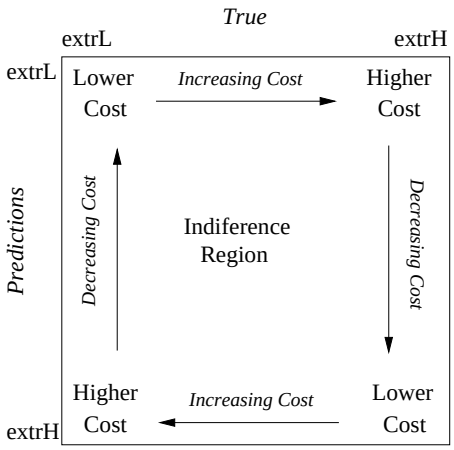

Figure 1: The abstract specification of the cost surface.

For applications without sign differentiation among the indifference region it should be sufficient to have a 3x3 matrix of costs for approximating the full weight surface. For applications with sign differentiation we need at least a 4x4 matrix of costs. Provided cost information is available, filling this matrix is easier than in the classification setup as the user does not have the "responsibility" of specifying the crisp thresholds dividing extremes from non-extremes, because the values will then be smoothed across the range of the axes. For the 3x3 case, the user only needs to select a value representative of $extr_L$, $extr_H$ and $indif$, and then filling in the adequate costs of the respective matrix.

Having a cost matrix specified as mentioned above we can use any function approximation method to obtain a smooth weight surface. In this work we use the function `loess()` implemented by B. Ripley for the R statistical software [8]. This function is a re-implementation of the "cloess" package by Cleveland et al. [2]. We use this function to fit a local polynomial of degree 1, by considering the values in the cost matrix as samples of the continuous cost surface, $w(Y, \hat{Y})$, that we want to approximate. The approximation of the surface $w()$ obtained with `loess()` satisfies our requirements of having a smooth cost surface without crisp thresholds. Still, we should note that many other function approximators could be used with the same type of result.

For applications where no cost information is available but still we want to bias our models towards accurate predictions on extremes, we need to describe means to setup the costs for the key points used for surface approximation. The critical question is to define what is a rare extreme value. We use the same definition as in Torgo and Ribeiro [11]. This means that we set $extr_L = adj_L$ and $extr_H = adj_H$, where $adj_L(adj_H)$ is the smallest observation that is greater or equal to the 1st quartile minus $1.5r$, with r being the interquartile range . After having defined these two extreme values we artificially create n grid points by diving the interval $extr_H - extr_L$ in n equally spaced bins. This means that we will have a $(n+2)$x$(n+2)$ matrix to fill in with costs. We set all but the lines (columns) involving extreme values to the cost 1, meaning an indifferent cost (equivalent to cost insensitive predictions). For the remaining cells of the matrix we use an arithmetic progression to setup the costs from the lowest to the highest

cost. Let us see a concrete example with a 5x5 cost matrix. Suppose $adj_L = -4$ and $adj_H = 15$. The intermediate points are at $-4 + \frac{(15-(-4))}{4} \times i$, for $i = 1, 2, 3$, thus leading to a matrix as shown in Table 2. The cells involving only indifferent values (predictions or true values) are set to 1 as mentioned before. Both $C_{L,L}$ and $C_{H,H}$ should be set to a very small value (less than 1), as these are the most favorable situations for our target applications. In our experiments (to be presented in Section 6) we have set these values to 0.1. Regarding the other values we use an arithmetic progression to fill them according to the change directions shown in Figure 1. For instance, we have set $C_{L,I1}$ to the base of the progression, b, and then defined $C_{L,I2}$ as $C_{L,I1} + r$ and $C_{L,I3}$ as $C_{L,I2} + r$, where r is the rate of the progression.

Table 2: Example cost matrix.

		True				
		-4	0.75	5.50	10.25	15
Preds.	-4	$C_{L,L}$	$C_{L,I1}$	$C_{L,I2}$	$C_{L,I3}$	$C_{L,H}$
	0.75	$C_{I1,L}$	1	1	1	$C_{I1,H}$
	5.50	$C_{I2,L}$	1	1	1	$C_{I2,H}$
	10.25	$C_{I3,L}$	1	1	1	$C_{I3,H}$
	15	$C_{H,L}$	$C_{H,I1}$	$C_{H,I2}$	$C_{H,I3}$	$C_{H,H}$

The method we have just described fulfills all requirements established in the beginning of this section for our evaluation metric. Moreover, it has sufficient generality for coping with different application types and cost surfaces. For instance, it is easy to setup cost matrices for applications with only high (or low) extremes.

The proposed metric could be plugged in to a modeling technique to force it to obtain a model whose parameters minimize the value of Equation (3), that better fits the requirements of domains where rare extreme values are the main targets in terms of predictive accuracy.

6 An Experimental Evaluation of the Proposal

We have carried out a series of experiments with the goal of checking the validity of our proposed metric in the task of identifying the models that are better from the perspective of being more accurate at rare extreme values. With this purpose we have designed the following experimental setup:

1. For each data set we have drawn a stratified test sample with 50% of the cases;

2. We randomly generate a set of prediction errors with the same size as the test sample. The errors are drawn from a normal distribution with average $\tilde{Y}/10$ and standard deviation $IQR(Y)/2$, where $\tilde{Y}$ is the sample median of the target variable; and $IQR(Y)$ is the interquartile range of the same variable. We then pick the n largest errors, where n is the number of outlier values of the distribution of Y, and increase these errors by a constant k. The overall objective of this step is to obtain a set of prediction errors that are credible for a standard model when making predictions for a problem

with some outliers. For these tasks we expect (we have confirmed this experimentally using several modelling techniques and several real world data sets), the models to achieve a performance of this type: normal-shape distribution of the error with some extreme errors typically occurring on the outliers of the target variable.

3. We then artificially allocate this set of generated errors to the test set target values, in two different ways, leading to the "performance" of artificial models A and B. For model A, the smallest errors are allocated to the outliers in the test set, thus leading to what could be considered to be an ideal model from our target applications. On the contrary, model B has the largest errors on the outlier values of the target, in what could be considered a "normal" behavior of a model in this type of tasks.

A performance metric that is biased towards accurate predictions in extremes, should clearly indicate that the performance of Model A is better than the performance of Model B. Notice that, given that the errors of the two models are exactly the same (only occurring at different test cases), metrics like the MSE or the MAD will show both models as having exactly the same score. In effect, both models have exactly the same cumulative distribution of errors (i.e. REC curve [1]), but these occur for different values of the target variable (i.e. they have significantly different REC surfaces [9]).

As we are carrying out these experiments for a large set of domains with a quite different range of target variable values, we have used a normalized version of our performance statistic to allow comparisons across domains, namely we have used,

$$NRExE = \frac{\sum_{i=1}^{n_{test}} w(y_i, \hat{y}_i) \cdot |y_i - \hat{y}_i|}{\sum_{i=1}^{n_{test}} w(y_i, \widetilde{Y}) \cdot |y_i - \widetilde{Y}|} \tag{5}$$

where $\widetilde{Y}$ is the sample median (obtained with the remaining 50%) of the data set cases.

Equation (5) provides a value comparing the performance of a model with the baseline model consisting of always predicting the median of the sample.

The goal of our experiments is to assert the score difference between models A and B. With this purpose we have measured the percentual difference of scores for all data sets,

$$100 \times \frac{Score_{metric}(B) - Score_{metric}(A)}{Score_{metric}(B)} \tag{6}$$

Positive values of this percentual difference indicate that the metric being evaluated is able to identify Model A as performing better than Model B. We obviously want this difference to be as high as possible, as Model A has an almost "ideal" performance from our target applications perspective.

We have compared the percentual difference score obtained with our proposed metric, $NRExE$, with the score obtained when using the most similar alternative, the weighted error measure of Equation (3). For this latter metric we have setup the case weights such that more weight is given to outlier values of the target, in order to make the measure a tough competitor to our proposal. More precisely, we have used a sigmoid function to obtain the weights. The shape of the sigmoid is a function of the distribution of the target variable, so

that more weight is given to the extremes of the distribution. Figure 2 gives an example of this weight function for the Abalone data set.

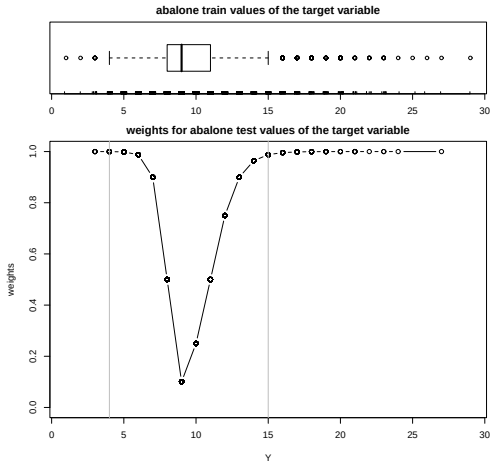

Figure 2: The sigmoidal case weights.

For each data set we have repeated the experiment outlined above 10 times. The results shown on Table 3 are the average and standard deviation of the observed percentual differences between Model A and B, when using $NRExE$ and the metric with sigmoid-based case weights. The best scores for each data set are indicated in bold. The used datasets are real world problems with a diverse set of rare extreme values types. For instance, some include both low and high extremes, while others include only one type of extremes. Due to space reasons we are not able to present the full characteristics of these problems.

The results reported in Table 3 provide good indications concerning the advantages of our proposed metric for domains where the main objective is to be accurate at rare extreme values. In effect, in most problems our metric correctly signals model A as being significantly better than model B. Still, on some domains we have observed a worse performance when compared to the metric based on sigmoid weights. Though this is a metric also biased toward extremes, we would expect out metric to perform always better, given that it is able to penalize false positives, while the sigmoid metric is not. After some initial inspection of these cases we think the cause is on the distribution of the errors used on this experiment and on the way the errors are being allocated to the test cases.

7 Conclusions

In this paper we have described the particular features of a class of problems with high practical importance: the prediction of rare extreme values. We claim that existing metrics for evaluating the performance of different predictive models have several drawbacks and perform poorly in identifying the best models in terms of predictive accuracy on the most important cases for these applications.

We have described several existing and plausible approaches to this type of

Table 3: The results of the comparison in terms of percentage difference between Models A and B.

Data Set	SigMetric (avg±sd)	NRExE (avg±sd)	Data Set	SigMetric (avg±sd)	NRExE (avg±sd)
algae1	52.6±4.8	**82.9±1**	deltaAilerons	**55.2±0.6**	5±0.4
algae2	55±4.6	**79.3±1.6**	ibm	**71.9±0.3**	7.4±0.4
algae3	71.1±2.4	**88.1±1**	abalone	**70.5±1.1**	6.5±0.5
algae4	73.8±14.1	**87.3±5.1**	cpuSmall	63±1	**81.1±0.4**
algae5	56.1±4.6	**83.9±1.5**	servo	74.4±5.8	**85.2±0.9**
algae6	84.2±0.8	**91.1±0.5**	cwDrag	**57.4±1.6**	2.8±2.6
algae7	52.4±11.1	**82.7±2.3**	co2Emission	**58.4±0.6**	17.8±6.5
Boston	**65±1.6**	21±25.3	availablePower	69.7±1.5	**71.5±0.7**
machineCpu	76.5±3.4	**77.9±1**	china	68.9±2.8	**71.8±1.4**
bank8FM	55.3±0.5	**63.6±0.8**	add	**56.6±0.3**	5.5±0.7

tasks. Based on an analysis of their main drawbacks we have presented a new metric that is particularly suited for our target applications.

In a set of experiments using real world data we have shown that this measure is able to identify the best model in terms of accuracy on the rare extreme target values, even on the most difficult scenario where both models have exactly the same error distribution and thus have the same score in "standard" metrics like the mean squared error.

One of the main impacts of the results of this work is that our metric can be used to compare different existing models on tasks where the main goal is the accuracy on rare extreme values. The use of our metric should provide better information concerning the merits of alternative models for these important tasks. Another important side effect of this work is the possibility of using the described metric in the search process of any modelling technique, so as to develop models that are built for maximizing the predictive performance on extreme values. We intend to pursue this research direction in our future work.

Acknowledgments

This work was partially supported by FCT project MODAL (POSI/SRI/40949/2001) co-financed by POSI and by the European fund FEDER, by a sabbatical scholarship of the Portuguese government (FCT/BSAB/388/2003) to L. Torgo and by a PhD scholarship of the Portuguese government (SFRH/BD/1711/2004) to R. Ribeiro.

References

[1] Jinbo Bi and K. P. Bennett. Regression error characteristic curves. In *Proceedings of the 20th International Conference on Machine Learning*, 2003.

[2] W.S. Cleveland, E. Grosse, and W.M. Shyu. *Local regression models*, chapter 8. Statistical Models in S. Wadsworth & Brooks/Cole, 1992.

[3] Chris Drummond and Robert C. Holte. Exploiting the cost of (in)sensitivity of decision tree splitting criteria. In *Proc. 17th International Conf. on Machine Learning*, pages 239–246. Morgan Kaufmann, San Francisco, CA, 2000.

[4] G.Weiss and F. Provost. The effect of class distribution on classifier learning: An empirical study. Technical Report Technical Report ML-TR-44, Department of Computer Science, Rutgers University, 2003.

[5] G.Weiss and F. Provost. Learning when training data are costly: The effect of class distribution on tree induction. *JAIR*, 19:315–354, 2003.

[6] Miroslav Kubat and Stan Matwin. Addressing the curse of imbalanced training sets: one-sided selection. In *Proc. 14th International Conference on Machine Learning*, pages 179–186. Morgan Kaufmann, 1997.

[7] Foster Provost, Tom Fawcett, and Ron Kohavi. The case against accuracy estimation for comparing induction algorithms. In *Proc. 15th International Conf. on Machine Learning*, pages 445–453. Morgan Kaufmann, San Francisco, CA, 1998.

[8] R Development Core Team. *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria, 2004. ISBN 3-900051-07-0.

[9] L. Torgo. Regression error characteristic surfaces. In R. Bayardo and K. Bennet, editors, *to appear in Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2005.

[10] L. Torgo and J. Gama. Regression using classification algorithms. *Intelligent Data Analysis*, 1(4), 1997.

[11] L. Torgo and R. Ribeiro. Predicting outliers. In N. Lavrac, D. Gamberger, L. Todorovski, and H. Blockeel, editors, *Proceedings of Principles of Data Mining and Knowledge Discovery (PKDD'03)*, number LNAI in 2838, pages 447–458. Springer, 2003.