Learning Through Utility Optimization in Regression Tasks

Paula Branco
LIAAD - INESC TEC
DCC - Faculdade de Ciências
Universidade do Porto
Porto, Portugal
Email: paula.branco@dcc.fc.up.pt

Luís Torgo
LIAAD - INESC TEC
DCC - Faculdade de Ciências
Universidade do Porto
Porto, Portugal
Email: ltorgo@dcc.fc.up.pt

Rita P. Ribeiro
LIAAD - INESC TEC
DCC - Faculdade de Ciências
Universidade do Porto
Porto, Portugal
Email: rpribeiro@dcc.fc.up.pt

Eibe Frank
Department of Computer Science
University of Waikato
Hamilton, New Zealand
Email: eibe@waikato.ac.nz

Bernhard Pfahringer
Department of Computer Science
University of Auckland
Auckland, New Zealand
Email: b.pfahringer@auckland.ac.nz

Markus Michael Rau
Ludwig-Maximilians-Universität München
Max-Planck-Institut für extraterrestrische Physik
Germany
Email: markusmichael.rau@gmail.com

***Abstract*—Accounting for misclassification costs is important in many practical applications of machine learning, and cost-sensitive techniques for classification have been studied extensively. Utility-based learning provides a generalization of purely cost-based approaches that considers both costs and benefits, enabling application to domains with complex cost-benefit settings. However, there is little work on utility- or cost-based learning for regression. In this paper, we formally define the problem of utility-based regression and propose a strategy for maximizing the utility of regression models. We verify our findings in a large set of experiments that show the advantage of our proposal in a diverse set of domains, learning algorithms and cost/benefit settings.**

I. Introduction

The task of learning with different costs is an important and well studied problem in the context of classification (e.g., [1], [2], [3]). Predictive approaches that take into account costs have important applications in many real-world domains (e.g., medicine, meteorology, and environmental science). Employing a cost-sensitive approach makes it possible to tailor the model closely to the specific problem domain, e.g., to improve predictive performance on minority classes in imbalanced datasets [4]. The main obstacle to wider usage of cost-sensitive learning is that it can be difficult to establish the cost matrix specifying the misclassification costs. Problem-specific error costs are often unavailable or difficult to obtain, requiring access to domain experts. Applying a purely cost-based approach is also problematic when dealing with real world problems that involve both costs and benefits.

The broader, more general framework of utility-based learning, which considers both benefits and costs, has been introduced in [1], [5]. In utility-based learning, a negative benefit (or cost) is assigned to model errors and a positive benefit to accurate predictions. As shown in [1], [5], this setting makes it possible to establish a baseline from which costs and benefits are defined, rendering the definition of the corresponding benefit matrix less prone to errors. Moreover, it enables differentiation between accurate predictions across the domain of the target variable by specifying correspondingly larger or smaller benefits.

As a result, utility-based learning is focused on maximizing utility, encompassing both costs and benefits, as opposed to cost-sensitive learning, which is focused solely on cost minimization. Our primary motivation for writing this paper is that utility-based learning is suitable for both classification and regression tasks, but research and application in this area has been concerned primarily with classification.

In the context of regression, the notion of a utility matrix needs to be extended to that of a utility surface—a function of the predicted and actual values of the target variable of the domain. In this paper, we address the problem of utility-based learning using such a utility surface. We formalize the problem of utility-based regression and propose and test a method to optimize the utility of regression models.

The main goals of this paper are to i) define the problem of utility-based learning in regression tasks and ii) propose and evaluate a solution for solving this problem. Our main contributions are that we i) define the utility-based learning problem; ii) propose and test a solution for this problem; and iii) analyse the impact of different utility surfaces on the performance achieved. The paper is organized as follows. In Section II, the problem definition is presented. Our proposal is described in Section III, and the results of an extensive experimental evaluation are discussed in Section IV. Section V provides a brief review of related work, and Section VI presents the main conclusions of this paper.

II. Problem Definition

In this section we will formally define the **utility-based regression** framework that will be used in this work. The

goal of regression is to derive a model that approximates an unknown function $Y = f(X_1, X_2, \cdots, X_p)$. This function maps the values of a set of p features onto the values of a target variable. The model m approximating f is fitted using a training set $D = \{\langle \mathbf{x}_i, y_i \rangle\}_{i=1}^{n}$ with known covariates $\mathbf{x}_i$ and target variable y_i. The trained model can then estimate target values $\hat{y}$ for new data with unknown target values. The optimization of the model given n training datums is usually performed by minimizing a given loss function $L(y, \hat{y})$, such as the squared error

$$L(y, \hat{y}) = (y - \hat{y})^2 \ . \tag{1}$$

This traditional regression objective does not take into account expected costs and benefits of the estimates, which can be important in real world applications. Let us consider an example of such an application.

Example 1 (Air Quality Prediction). *Consider the LNO2Emissions data set,* which has a continuous target variable (LNO2) that represents hourly measured values of the logarithm of the concentration of NO2 (particles) in Oslo, Norway, between October 2001 and August 2003. The seven covariates include information on the traffic, temperature, wind, hour and day. Table I presents a more detailed description of this data set. High values of LNO2 indicate a bad air quality as opposed to lower LNO2 values. However, both extremes (low and high) are rare in the data set. This can be observed in Figure 1 which shows the density function of the target variable approximated through a kernel density estimator.*

Considering this data set, suppose that a decision maker is interested in predicting the LNO2 variable, for determining when to impose traffic restrictions to prevent the location from reaching a dangerous atmosphere. In this case, the decision maker's preferences are not uniform across the target variable domain, and his main goal is to obtain a predictive model with high accuracy on high extreme values of LNO2.

We start by introducing some notions, starting with the concept of a *relevance function*, which was proposed by Torgo and Ribeiro [6] and Ribeiro [5]. It expresses the importance that the user assigns to different values of the target variable.

Definition 1 (Relevance Function). *A relevance function, which we will denote by $\phi()$, is a function that maps the target variable into a scale of relevance in [0,1]:*

$$\phi(y) : \mathcal{Y} \rightarrow [0, 1] \tag{2}$$

where 0 represents minimum relevance and 1 represents maximum relevance.

This function represents the benefit of zero error predictions across the target variable domain, and is the analogue of the diagonal of a benefit matrix in regression tasks. The assumption that completely accurate predictions cannot incur costs motivates the introduction of this function.

*A sample of 500 cases from a data set that has its origin in a study relating air pollution with traffic volume and meteorological variables. The data is available from the StatLib Datasets Archive: http://lib.stat.cmu.edu/datasets/.

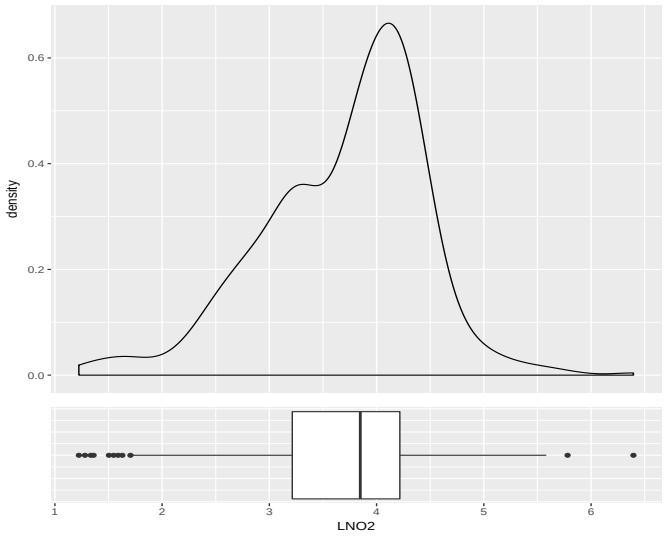

Fig. 1. Distribution of the target variable LNO2.

To encode a decision maker's preferences and incorporate these in our modelling process, we can use the notion of a relevance function to define a function that assigns a utility score to pairs of estimated and actual target values. This function can be viewed as a *utility surface*.

Definition 2 (Utility Surface). *A utility surface is a function that maps pairs of values $(y, \hat{y})$ into a utility scale in [-1,1]:*

$$\begin{aligned} U : \mathcal{Y} \times \mathcal{Y} &\longrightarrow [-1, 1] \\ (y, \hat{y}) &\longmapsto U(y, \hat{y}) = g(L(y, \hat{y}), \phi(y), \phi(\hat{y})) \end{aligned} \tag{3}$$

where a positive utility value represents the benefit and a negative utility value represents the negative benefit (cost) associated with predicting $\hat{y}$ for the true value y.

The notion of a utility surface was presented by Torgo and Ribeiro [6] and Ribeiro [5] to extend the concept of benefit matrices for classification, as proposed by Elkan [1], to regression. A utility surface can be thought of as a continuous version of the benefit matrix. Our definition of the utility of $(y, \hat{y})$ is based on a function $g(.)$ that establishes utility using three components: i) the loss $L(y, \hat{y})$, ii) the relevance of y, and iii) the relevance of $\hat{y}$. This means that both the magnitude of the error observed and the user-assigned relevance scores for the true and predicted values contribute to the utility score.

Note that a utility surface includes all the information in the corresponding relevance function. The utility of perfect predictions corresponds to their relevance scores: for all pairs of points (y, y), $U(y, y) = g(L(y, y), \phi(y), \phi(y)) = g(0, \phi(y), \phi(y)) = \phi(y)$. We make the notion of relevance explicit because it is used in an elegant framework for eliciting a utility surface presented in [5]. Establishing a benefit matrix for classification is challenging; establishing a utility surface even more so. To address this problem, Ribeiro [5] developed

TABLE I
LNO2 DATA SET CHARACTERISTICS

	Features							Target
	LCarsH	*Temp*	*WSpeed*	*TempDiff*	*WDir*	*Hour*	*Day*	*LNO2*
Min	4.13	-18.60	0.30	-5.40	2.00	1.00	32.00	1.22
1st Qu.	6.18	-3.90	1.68	-0.20	72.00	6.00	118.80	3.21
Median	7.43	1.10	2.80	0.00	97.00	12.50	212.00	3.85
Mean	6.97	0.85	3.06	0.15	143.40	12.38	310.50	3.70
3rd Qu.	7.79	4.90	4.20	0.60	220.00	18.00	513.00	4.22
Max	8.35	21.10	9.90	4.30	359.00	24.00	608.00	6.40

a method for automatically obtaining both a relevance function and a utility surface for regression tasks. The method establishes relevance scores assuming that the rarest extreme values are likely to be the most important ones. A utility score for each pair of values $(y, \hat{y})$ is derived by taking into account both the error measured through a given loss function and the relevance of y and $\hat{y}$. More precisely, this method resorts to the notions of benefits and costs of numeric predictions for providing the following definition of the utility of the predictions of a regression model,

$$\begin{aligned} U^p_\phi(\hat{y}, y) &= B_\phi(\hat{y}, y) \; - \; C^p_\phi(\hat{y}, y) \\ &= \phi(y) \cdot (1 - \Gamma_B(\hat{y}, y)) \; - \; \phi^p(\hat{y}, y) \cdot \Gamma_C(\hat{y}, y) \end{aligned} \tag{4}$$

where $B_\phi(\hat{y}, y)$, $C^p_\phi(\hat{y}, y)$, $\Gamma_B(\hat{y}, y)$ and $\Gamma_C(\hat{y}, y)$ are functions related to the notions of costs and benefits of predictions that are defined in [5].

The method is based on the assumption that the user is primarily interested in either one or both extreme ends of the spectrum of target values. The motivation behind this assumption is that rare and important values are often located at the extremes of the distribution of the target variable. The framework allows the user some flexibility in deciding which type of errors should be more or less penalized, providing an automatic mechanism that adjusts the costs to different settings. This control is accomplished through a parameter $p \in [0, 1]$ that specifies which types of errors should incur higher costs. Selecting the value 0.5 for p assigns the same weight for all types of errors. This mechanism can be thought of as the parallel in regression to the decision of assigning more costs to false positives, false negatives or both types of errors in a classification problem.

Example 2 (Relevance Function and Utility Surface for the LNO2 Variable). *Assume that a group of domain experts has provided to the decision maker the relevance function in Figure 2 and the utility surface displayed in Figure 3. The latter specifies the benefits and costs, i.e., the utility for pairs of true and predicted values of LNO2.*

In this case, high predictive accuracy of a model on the high values of the target variable yields large benefits, while high accuracy on the remaining values has benefits that tend to zero as the values get lower. Simultaneously, the models incurs large costs when substantially mispredicting on high values of the target variable while the costs for mispredictions on the low LNO2 values are lower. This is controlled with the parameter p, which in this case is set to 0.75. This means that mistakes occuring on high LNO2 values are more costly than those occuring on the lower values. This surface reflects the domains knowledge that high predictive accuracy on the high values of the LNO2 variable is more important than high predictive accuracy on the low values.

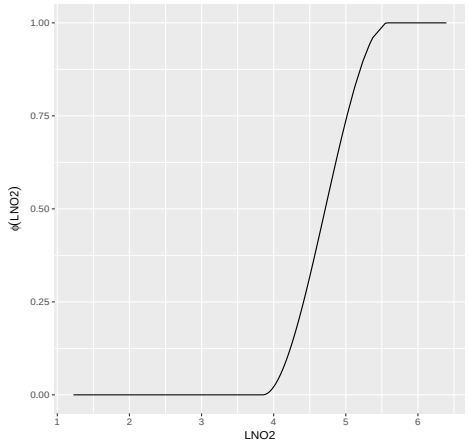

Fig. 2. Relevance function for the LNO2 variable considering that high extreme values are the most important ones.

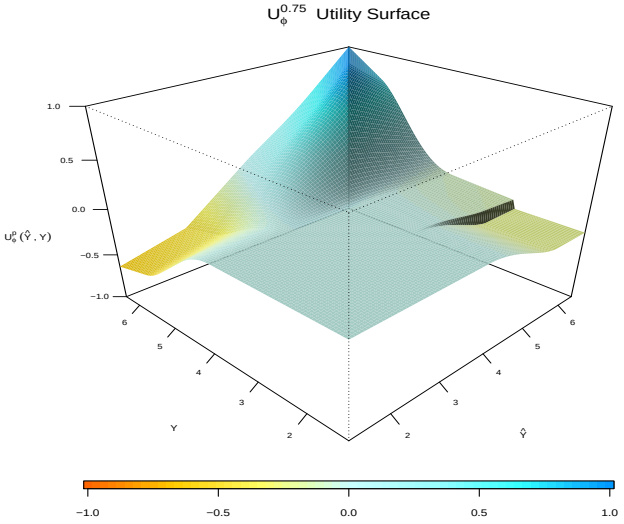

Fig. 3. Utility surface for LNO2.

Given the concept of a utility surface, we are now ready

to state the task of utility-based regression, which is based on the assumption that a rational agent should maximize expected utility.

Definition 3 (Utility-based Regression). *Consider a predictive task with a continuous target variable Y whose domain is $\mathcal{Y}$ and a user-defined utility surface $U(y, \hat{y})$. The goal of utility-based regression is to obtain the model that provides the maximum utility.*

The main goal of utility-based regression tasks is to obtain predictions that achieve high expected utility according to the user preferences expressed through a utility surface. This contrasts with standard regression approaches, which are focused on minimizing expected loss.

III. Learning by Optimizing Utility

Traditional loss function minimization is not appropriate when performing utility-based regression; more suitable performance metrics must be applied. Observing scores of a loss function is insufficient because the utility surface must be taken into account. The user's goal is to maximize utility; therefore, the model's performance must be assessed by considering the utility of the predictions. This is analogous to the case of cost-sensitive classification where the performance assessment metrics reflect the expected cost rather than the average error. Ribeiro [5] proposed two metrics that are suitable for evaluating utility-based regression: Mean Utility (MU) and Normalized Mean Utility (NMU). Equations 5 and 6 provide the definitions of these two evaluation metrics for the setting considered in this paper. NMU is a normalized version of MU that yields scores in the $[0, 1]$ interval. We will use NMU in this paper.

$$MU = \frac{1}{n}\sum_{i=1}^{n} U(y_i, \hat{y}_i) \tag{5}$$

$$NMU = \frac{\sum_{i=1}^{n} U(y_i, \hat{y}_i) + n}{2n} \tag{6}$$

Example 3 (Unsuitability of traditional metrics). *Let us consider a test sample containing 10 examples of the previously considered LNO2 Emissions data set. For this test set, we generated three artificial model predictions. These predictions are shown in Figure 4. The models were generated so that model m1 obtains the best performance on the high values of the target variable, model m2 displays the best performance on the mean values of the target variable distribution and model m3 performs well on low Y values. Table II shows the performance assessment of these models according to several different settings, including the standard loss functions Mean Absolute Deviation (MAD) and Mean Squared Error (MSE). The different utility-based evaluation settings were obtained by applying corresponding utility surfaces. For each of these settings, the superscript pi represents the type of penalization in the utility surface used and the subscript H, L, or B indicates whether high, low or both low and high extremes were considered relevant in the utility surface. The best model for each metric and setting considered is shown in bold. The rankings in Table II show that the use of MAD or MSE would lead to the selection of the $m2$ model. In contrast, when considering the NMU measure, this algorithm is never selected. Moreover, it is clear that depending on the setting considered, i.e., depending on the utility surface defined, the ranking of the three models can be very different: NMU is able to adapt to different user preferences bias and the results obtained are strongly influenced by the utility-surface considered, as expected. Figure 5 shows the three models' predictions for the setting $NMU^{B}_{p0.5}$, where extreme values at both ends of the spectrum are considered relevant and the same penalization is assigned for both types of errors. The utility is shown in color-coded fashion. The resulting heat map exhibits symmetry because of the symmetric setting considered for constructing the utility function.*

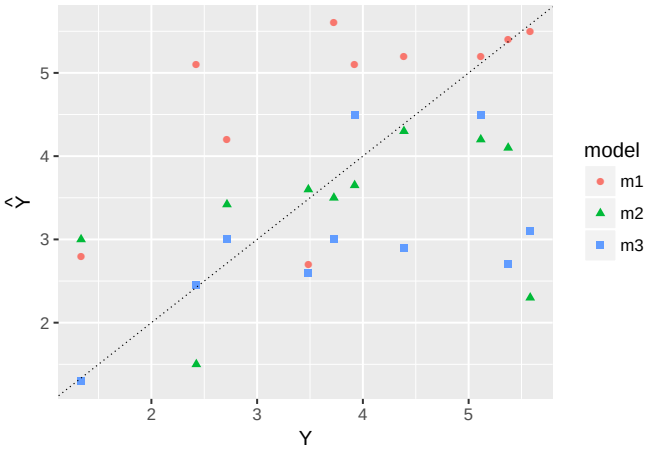

Fig. 4. Predictions of three artificial models on a 10 sample test set.

Note that the above example was constructed to exhibit a complete mismatch between the models' performance obtained using standard metrics and that obtained using utility-based metrics. Such extreme discrepancies may not occur in every situation in practice. The important fact that we want to highlight is that standard metrics do not reflect the model's performance in accordance with the user preferences.

Let us now formally define how the optimal prediction for a given case can be determined in a utility-based context. This definition is based on establishing the expected utility of a prediction using the conditional density of the target variable. Let $f_{Y|X}$ represent the conditional probability density function of Y given the occurrence of the value x of X. Equation 7 provides the definition of the conditional probability density function based on both the joint and marginal density functions.

$$f_{Y|X}(y|X = x) = \frac{f_{X,Y}(x, y)}{f_X(x)} \tag{7}$$

where $f_{X,Y}(x, y)$ represents the joint density of X and Y, and $f_X(x)$ is the marginal density of X.

TABLE II
DIFFERENT METRICS RESULTS ON THE THREE ARTIFICIAL MODELS DISPLAYED IN FIG. 4.

	Scores			Decision Rank		
	m1	*m2*	*m3*	*m1*	m2	m3
MAD	1.05	**0.95**	0.98	3	**1**	2
MSE	1.77	**1.75**	1.76	3	**1**	2
NMU^{H}_{p0}	**0.58**	0.56	0.54	**1**	2	3
$NMU^{H}_{p0.5}$	**0.61**	0.54	0.52	**1**	2	3
NMU^{H}_{p1}	**0.64**	0.52	0.50	**1**	2	3
NMU^{L}_{p0}	**0.45**	0.38	0.42	**1**	3	2
$NMU^{L}_{p0.5}$	0.41	0.38	**0.47**	2	3	**1**
NMU^{L}_{p1}	0.38	0.38	**0.52**	2	2	**1**
NMU^{B}_{p0}	0.54	0.53	**0.57**	2	3	**1**
$NMU^{B}_{p0.5}$	**0.57**	0.50	0.56	**1**	3	2
NMU^{B}_{p1}	**0.60**	0.48	0.54	**1**	2	3

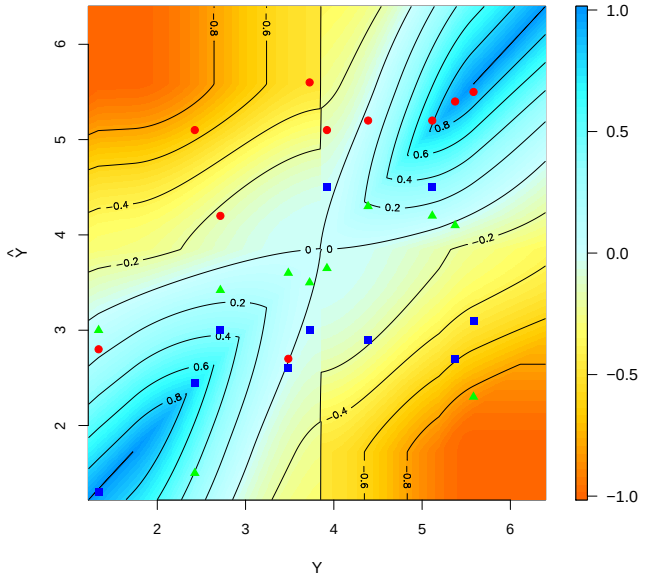

Fig. 5. A utility setting and the three artificial models predictions.

For a given example $q = \langle \mathbf{x}_k, y_k \rangle$, with y_k unknown, the optimal prediction y^* can be determined as follows:

$$y^* = \underset{z \in Y}{\operatorname{argmax}} \int f_{Y|X}(y|X = \mathbf{x}_k).U(y, z)\ dy \qquad (8)$$

Equation 8 shows how the optimal prediction for a given example can be obtained, assuming the true conditional density for the target variable is known. This equation is the extension to regression and utility-based problems of the minimization of the conditional risk [1]. Equation 8 shows how to obtain the prediction that maximizes the expected utility. To apply this mechanism in practice, we are required to provide two components: i) a utility surface that states the user preference bias and ii) an estimate of the conditional probability density function $f_{Y|X}$.

In the following we propose a method that is able to obtain an estimate for the optimal prediction considering the above conditions and Equation 8.

A. Conditional density probability function estimation through class probabilities

The first component we need is an estimate of the conditional density function. In this section we briefly describe how we obtained an estimate for $f_{Y|X}$ using a class probability estimator. The method presented here is described in Frank and Bouckaert [7] and Rau [8] and uses ordinal classification to obtain an estimate for $f_{Y|X}$.

Let us assume we have a class probability estimator $\hat{p}$ that is able to provide for each given class c and case q an estimate of the class probability $\hat{p}(c|q)$ using a training set. The main idea is to use $\hat{p}(c|q)$ to derive weights for each value of the target variable conditioned on q.

We begin by discretizing the continuous target variable values in the training set into equal width, non-overlapping bins which will be treated as classes. Let c_y represent the class that contains the target value y and let n_{c_y} be the number of examples in that class.

Given a case $q = \langle \mathbf{x}_k, y_k \rangle$, we derive a weight w_i for each y_i value in the training set as follows:

$$w_i(q) = \frac{\hat{p}(c_{y_i}|q)}{n_{c_{y_i}}} \qquad (9)$$

We then apply a weighted Gaussian kernel density estimator in conjunction with the weights obtained to yield the following estimate of the conditional probability density function:

$$f_{Y|X}(y|X = \mathbf{x}_k) = \sum_{i=1}^{n} w_i(q)\mathcal{N}(y; y_i, \sigma^2) \qquad (10)$$

Following Silverman's "rule of thumb" [9], the value of σ is set as follows in our experiments: $\sigma = 0.9An^{-1/5}$, where $A = min\{\sigma_X, \frac{IQR}{1.34}\}$. Further details regarding this method for conditional probability density estimation can be found in [7], [8].

An important question that arises is which probabilistic classifier should be used for obtaining the weights in the kernel density estimates $f_{Y|X}$ for each X. It is appropriate to select a classification approach that is closely related to the corresponding regression approach used to estimate $\hat{y}$. For example, if an algorithm for building regression trees is used to estimate $\hat{y}$, it is appropriate to use a corresponding classification tree algorithm to obtain the class probabilities that yield the weights in the kernel density estimate. When dealing with cost-sensitive classification, a mismatch between the probability estimator and the classifier used has been shown to negatively impact performance [10]. This motivated our approach of selecting the probabilistic classifier most closely related to the regression algorithm being used. Moreover, we will assume, as done by Domingos [2], that the user is able to select the regression scheme that best adapts to the problem domain that is being considered. This selected

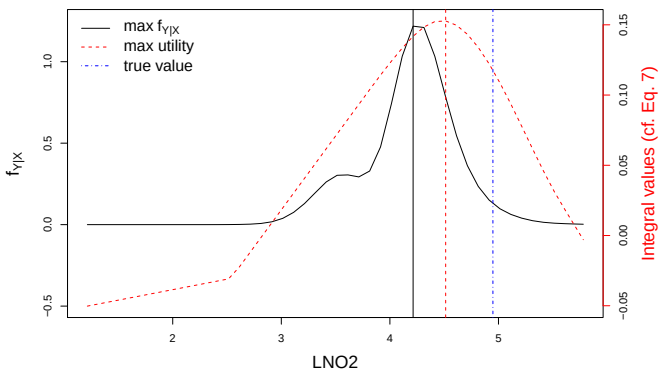

Fig. 6. Results from utility optimization and conditional density estimation for one test point with a true LNO2 value of 4.949.

scheme is then used for learning the regression model and its probabilistic classifier counterpart is used for obtaining the conditional probability density estimates.

B. Optimizing Utility

Regarding the utility optimization procedure presented in Equation 8, our proposal involves, for a given case q, the evaluation of $f_{Y|X}$ for $X = \mathbf{x}_k$ and the evaluation of the integral for all $z \in Y$. To be more precise, consider a case $q = \langle \mathbf{x}_k, y_k \rangle$ for which we want to obtain a prediction for the true target variable value y_k. Let us suppose we want to assess the effects of predicting z for the true value y_k of case q. The integral in Equation 8 allows us to obtain an estimate of the *conditional utility* of predicting z. Given that our goal is to maximize utility, we only have to determine which $z \in Y$ enables us to obtain the largest value. This z value is selected as the optimal prediction for case q because it provides the estimate that yields the highest expected utility.

Regarding the implementation of this algorithm, we use a parameter ϵ for specifying the granularity used in the approximations. Given this granularity ϵ, we evaluate the function $f_{Y|X}$ and the surface U in the domain Y using a set of points equality spaced by ϵ. Figure 6 shows an illustration for a given test case of the LNO2Emissions data set. The solid black line represents the $f_{Y|X}$ estimate obtained conditional on the considered case. The red dashed line displays the values obtained for the integral proposed in Equation 8 across the Y domain. The vertical lines in the figure show the true target variable value (dashed blue), the LNO2 value with the highest probability given the information in the probability density function conditional on the case (solid black) and the LNO2 value with the highest estimated conditional utility (dashed red). We observe that, in this case, the prediction is pushed to a value with lower conditional probability but higher expected utility.

IV. Experimental Evaluation

In this section we present the experimental evaluation conducted and discuss the results obtained. For reproducibility purposes, the data used and the code for all the experiments described is available at https://github.com/paobranco/UtilityOptimizationRegression. We have used the free open source R environment to ensure easy replication of our work.

TABLE III
USED DATA SETS AND CHARACTERISTICS. (N: NR OF CASES; $tpred$: NR OF PREDICTORS; $p.nom$: NR OF NOMINAL PREDICTORS; $p.num$: NR NUMERIC PREDICTORS;$nRare$: NR. CASES WITH $\phi(Y) > 0.8$; $\%Rare$: $100 \times nRare/N$).

Data Set	N	tpred	p.nom	p.num	nRare	% Rare
servo	167	4	2	2	34	20.4
a6	198	11	3	8	33	16.7
Abalone	4177	8	1	7	679	16.3
machineCpu	209	6	0	6	34	16.3
a3	198	11	3	8	32	16.2
a4	198	11	3	8	31	15.7
a1	198	11	3	8	28	14.1
a7	198	11	3	8	27	13.6
boston	506	13	0	13	65	12.8
a2	198	11	3	8	22	11.1
a5	198	11	3	8	21	10.6
fuelCons	1764	38	12	26	164	9.3
availPwr	1802	16	7	9	157	8.7
bank8FM	4499	9	0	9	288	6.4
Accel	1732	15	3	12	89	5.1
airfoild	1503	5	0	5	62	4.1

TABLE IV
REGRESSION ALGORITHMS AND THEIR PARAMETER VALUES, AND THE RESPECTIVE R PACKAGES.

Learner	Parameter Variants	R package
SVM	$cost = \{10, 150\}$ $gamma = \{0.01, 0.001\}$	**e1071** [11]
Random Forest	$mtry = \{5, 7\}$ $ntree = \{500, 750, 1500\}$	**randomForest** [12]

A. Evaluation of our Proposed Approach

The experiments carried out aim at testing the effectiveness of our proposed approach for optimizing the utility in the context of utility-based regression tasks. For this goal we selected 16 regression data sets. Table III shows the main characteristics of these data sets. We obtained a relevance function for each of them through the automatic method proposed by Ribeiro [5], which we discussed in Section II. The relevance functions produced assign higher relevance to rare extreme values of the target variable. Table III shows the number and percentage of rare values in each data set when considering a threshold of 0.8 on the relevance values. We also used the method proposed by Ribeiro [5] for defining different utility surfaces for each data set. For each data set we obtained four utility surfaces by varying the parameter p that specifies which types of errors should be more/less penalized. For each data set, a different utility surface was obtained for each p value in $\{0.2, 0.5, 0.8, 1\}$.

We have evaluated our proposal using 2 different learning algorithms and we considered several parameter variants for each algorithm type. The algorithms, the set of parameters tested and the respective R packages used in our experiments are described in Table IV.

We applied 10 learning approaches to each of the 16 data sets (6 Random Forest variants + 4 SVM variants) using both the learner without utility optimization and our proposed approach for optimizing the utility. Moreover, in each of the previous combinations, we used each one of the four utility surfaces defined for each data set. We set the granularity parameter ϵ to 0.1 in all data sets.

All experiments were evaluated using the NMU (Normalized Mean Utility) metric defined in Section III. The use of a normalized measure allows us to obtain results that are comparable between different data sets. The NMU values were estimated by 2 repetitions of a 10-fold cross validation process. We assessed the statistical significance of the observed paired differences through the non-parametric Wilcoxon Signed Rank test.

Table V shows the 2×10-fold CV estimate of the NMU metric on each data set and learner, evaluated using both the baseline learner by itself (labelled as Orig) and our proposed utility optimization strategy (labelled as Util). The results displayed in both tables concern only the utility surfaces obtained with p set to 0.5 and 1. Given the space constraints, the results obtained for the remaining utility surfaces and other complementary results are available at https://github.com/paobranco/UtilityOptimizationRegression. The advantage of our approach is clear, specially for the most extreme utility surface setting: we obtain improvements in the majority of data sets and learners in terms of the utility of the models observed. The observed gains are often relatively small but there is an improvement in almost all cases.

Figures 7 to 10 show the wins (left/blue) and losses (right/brown) obtained through the paired Wilcoxon signed rank test for different values of p and for each base learner. To obtain these results, our proposed method was compared against the baseline regression schemes 160 times (96 and 64 times for the RF and SVM learners, respectively). The results show the wins and losses (lighter bars) and the significant wins and losses (darker bars) for a significance level of 0.05. In all the settings of p, the number of wins of the utility optimization strategy is overwhelming for the SVM learner. Regarding the RF learner, these results are more accentuated for the most extreme value of p. In all utility surface settings tested, the number of wins/significant wins is always larger than the number of losses/significant losses. These results show that the proposed strategy is able to efficiently adapt to different conditions, clearly improving the results obtained for the utility metrics. Similar results were obtained for a significance level of 0.01 and are available at https://github.com/paobranco/UtilityOptimizationRegression.

Figures 11 and 12 show the results obtained for different values of the parameter p when averaging across learning algorithms and parameter settings. The results show that the baseline learners are more insensitive to this parameter in terms of the NMU metric. In fact, the majority of the data sets show a slight decrease in performance when the value of p is increased. Regarding the results obtained with the strategy that optimizes utility (Figure 12), we can observe that several

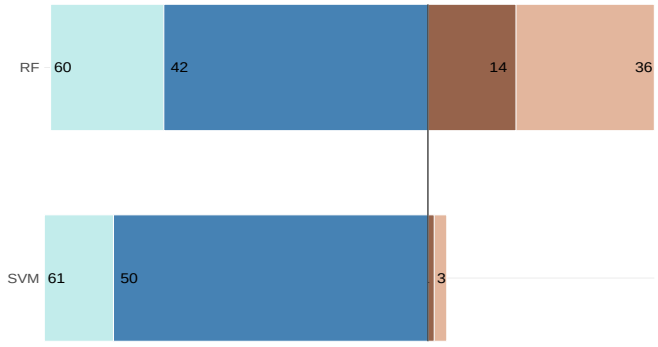

Fig. 7. Wins (left) and losses (right) of the utility optimization strategy against the baseline for a utility surface with p=0.2.

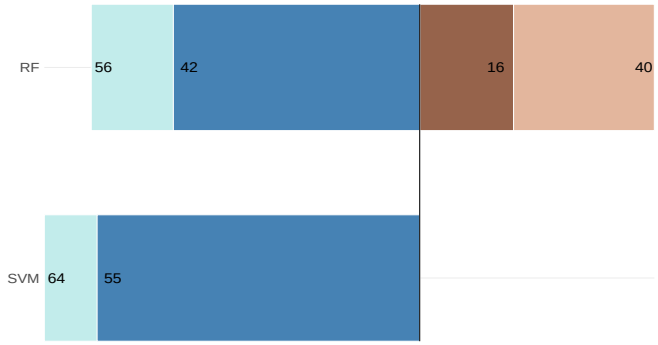

Fig. 8. Wins (left) and losses (right) of the utility optimization strategy against the baseline for a utility surface with p=0.5.

of the data sets exhibit an increase in the NMU values when p is set to the most extreme values (0.2 and 1). Only 1 out of 16 data sets presents a decrease in the NMU values when p is increased from 0.5.

Figures 13 and 14 show the impact of the proposed approach in the predictions of the *airfoild* data set for different settings of the utility surface. The lower extreme values of this data set target variable are the most relevant ones according to the automatic method used. We measured, for each example, the difference between the predictions obtained by the original learner and those obtained using the proposed approach. The average value of these shifts on the 2×10 - fold CV is represented on the vertical axis. This data set was selected as a representative of the data sets that do not always achieve a better performance with our proposed strategy. We observe a different impact on the predictions when comparing the results obtained with the two utility surfaces. Moreover, we also observe a higher divergence between the shifts applied for the SVM and RF algorithms when considering the utility surface with $p = 1$.

V. Related work

A diverse set of proposals exist for cost-sensitive learning in the context of classification tasks. These proposals can be categorized into direct and meta-learning methods [13]. The former manipulate the learning algorithm internally to

TABLE V
MEAN RESULTS OF NMU METRIC BY LEARNING ALGORITHM AND DATA SET FOR A PENALIZING FACTOR p OF 0.5 AND 1 IN THE UTILITY SURFACE.

Data sets	$p = 0.5$				$p = 1$			
	SVM		RF		SVM		RF	
	Orig	Util	Orig	Util	Orig	Util	Orig	Util
servo	0.4875	**0.5601**	**0.5718**	0.5655	0.4849	**0.5613**	**0.5727**	0.5667
a6	0.5072	**0.5207**	**0.514**	0.5069	0.4931	**0.5441**	0.5166	**0.5298**
Abalone	0.5705	**0.5893**	0.5764	**0.5884**	0.5629	**0.6037**	0.5728	**0.6035**
machineCpu	0.5617	**0.5618**	**0.5641**	0.5635	0.5614	**0.565**	0.5641	**0.5664**
a3	0.4927	**0.5051**	**0.5048**	0.4958	0.4739	**0.5391**	0.5123	**0.5305**
a4	0.514	**0.5313**	0.5284	**0.5331**	0.5041	**0.5608**	0.5308	**0.566**
a1	0.5297	**0.5394**	0.5479	**0.5531**	0.5232	**0.5682**	0.5469	**0.5707**
a7	0.4859	**0.4975**	0.481	**0.484**	0.4652	**0.5158**	0.4936	**0.5013**
boston	0.5743	**0.5775**	0.5782	**0.5814**	0.5733	**0.5798**	0.5779	**0.5831**
a2	0.518	**0.5289**	**0.5269**	0.5242	0.5085	**0.5515**	0.5236	**0.5488**
a5	0.5172	**0.5282**	0.5257	**0.5261**	0.5083	**0.5486**	0.5236	**0.5473**
fuelCons	0.6135	**0.6194**	0.6171	**0.6259**	0.6129	**0.6217**	0.6165	**0.6268**
availPwr	0.5766	**0.5774**	0.5818	**0.5821**	0.5766	**0.5777**	0.5817	**0.5819**
bank8FM	0.5692	**0.5702**	**0.5703**	0.5694	0.569	**0.5713**	0.5702	**0.5706**
Accel	0.558	**0.5611**	0.5638	**0.5655**	0.5576	**0.5626**	0.5637	**0.5657**
airfoild	0.4508	**0.4633**	0.4575	**0.4635**	0.4412	**0.4472**	**0.4534**	0.4514

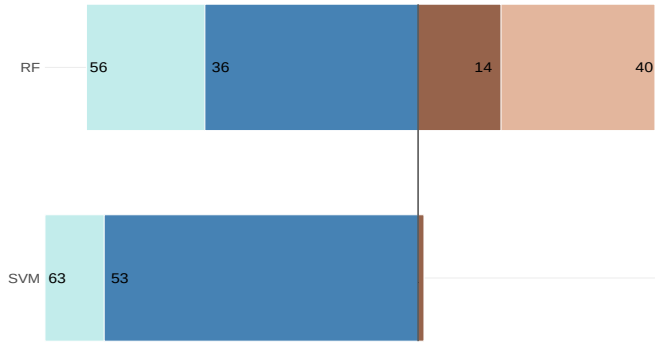

Fig. 9. Wins (left) and losses (right) of the utility optimization strategy against the baseline for a utility surface with p=0.8.

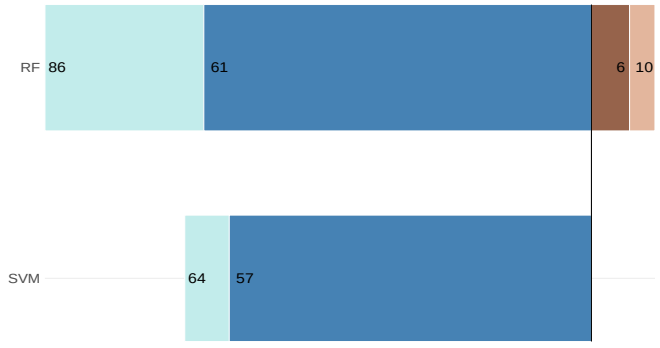

Fig. 10. Wins (left) and losses (right) of the utility optimization strategy against the baseline for a utility surface with p=1.

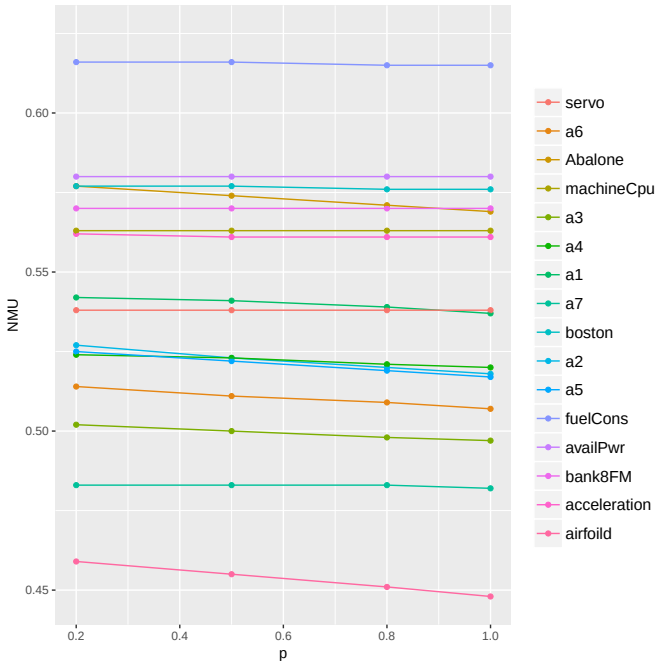

Fig. 11. NMU results obtained for the baseline learners with different values for parameter p.

make it cost-sensitive (e.g. [14]). The latter aim at creating a "wrapper" around a cost-insensitive learning algorithm in order to transform it into a cost-sensitive algorithm (e.g. [2]). As discussed earlier, these proposals are mostly focused on the cost-sensitive paradigm rather than the utility-based framework.

For regression tasks, cost-sensitive learning has been studied less. The issue of considering asymmetric loss functions that model the costs of under and over-predictions has been considered in some proposals. Orallo [15] studied the use of probabilistic reframing for addressing cost-sensitive regression problems with asymmetric losses. Zhao [16] and Bansal [17] also presented two proposals for addressing the problem of cost-sensitive regression. The proposed method tries to minimize the expected misprediction costs in a post-hoc manner

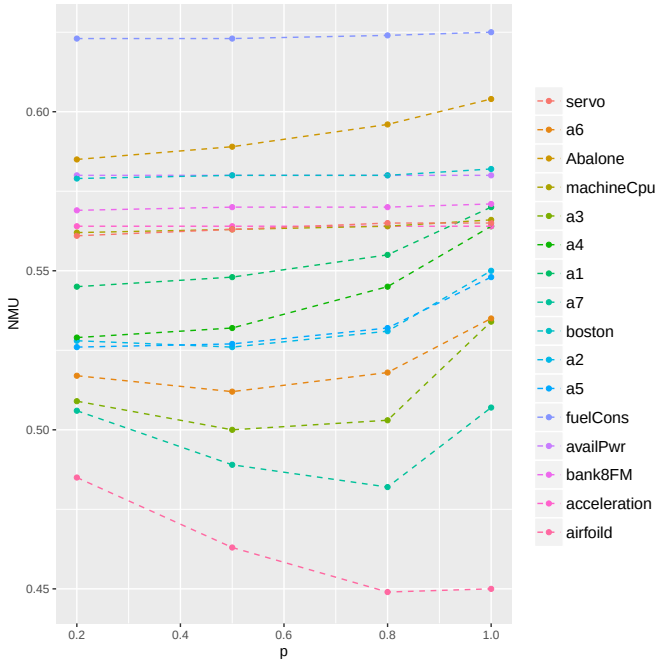

Fig. 12. NMU results obtained for the utility optimization strategy with different values for parameter p.

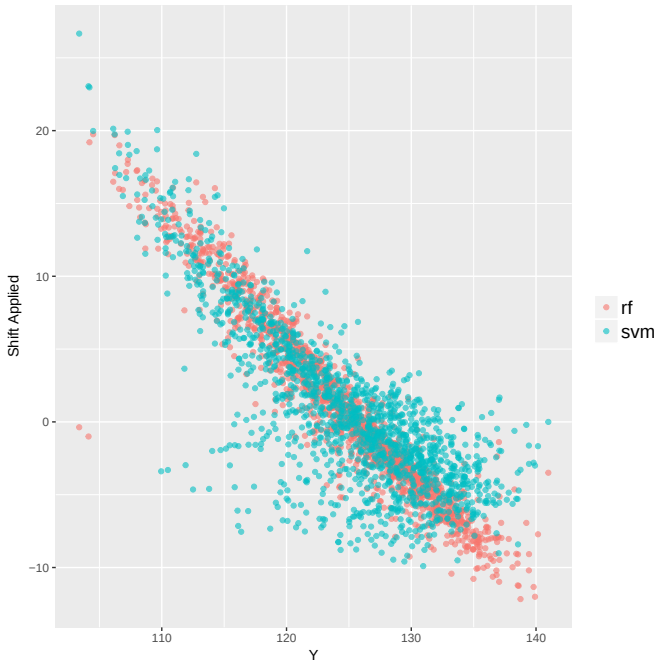

Fig. 13. Average shift applied in the *airfoild* data predictions between the original learner and the proposed algorithm for a utility surface with $p = 0.5$.

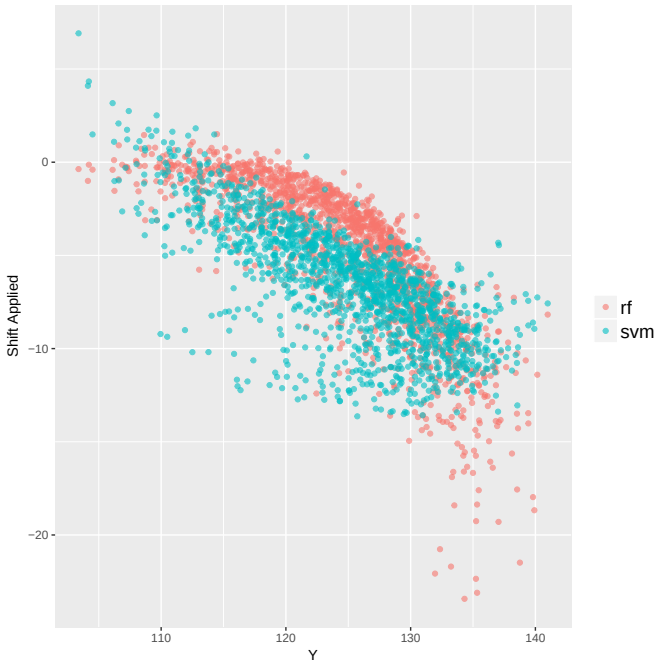

Fig. 14. Average shift applied in the *airfoild* data predictions between the original learner and the proposed algorithm for a utility surface with $p = 1$.

assuming that asymmetric costs are available for the problem.

Regarding utility-based regression, the works of Torgo and Ribeiro [6] and Ribeiro [5] are, as far as we know, the only approaches that address this issue. Ribeiro [5] proposed utility-based Rules (ubaRules), a rule ensemble system developed for obtaining models biased towards the preferences defined in a utility function. The system design follows two steps: i) obtain rule ensembles from different regression trees, and ii) select the best rules to include in the final ensemble. On several stages of the algorithm, the utility function is used.

Frequently, other problems are tackled through cost-sensitive methods. This happens, for instance, with imbalanced domains. Dealing with imbalanced domains is a challenging problem for several tasks, such as binary classification, multi-class classification, regression, time series classification/regression, data streams, and multi-target and multi-label problems among others (e.g., [18], [19]). In imbalanced domains, the user preferences are also not uniform across the target variable domain. However, typically the user does not define costs and/or benefits. The user preferences are known to be biased towards the most rare cases but they are usually not quantified, which blocks the evaluation of costs/benefits.

Many approaches have been developed for solving the problem of imbalanced domains through the cost-sensitive framework (e.g., [4]). The performance assessment of the models is achieved through metrics such as G-mean, precision and recall or AUC.

VI. Conclusions

In this paper, we formally present the problem of utility-based learning in regression tasks. To achieve this, we resort

to the definition of a relevance function and a utility surface. We show that standard regression metrics are not suitable for evaluation in this setting and appropriate metrics based on utility are required.

We formally define the process of obtaining optimal predictions for a given utility-based regression task. The strategy we propose for optimizing utility uses a utility surface and an estimation of the conditional probability density function of the target variable. The optimal prediction for a case is obtained by maximizing expected utility.

A large set of experiments were carried out with a diverse set of learning algorithms, regression data sets and utility surfaces. The results obtained highlight the advantages of our proposal across several different settings.

The key contributions of this paper are as follows: i) we define the problem of utility-based learning in regression domains; ii) we propose and test a solution for optimizing the predictions in this setting; and iii) we analyse the impact of different utility surfaces on the performance achieved.

Regarding future work, we would like to explore the performance of this method when using other learning algorithms. Moreover, it would be interesting to understand if there are specific data characteristics and learning algorithms that provide a suitable setting for our method.

Acknowledgment

This work was financed by the ERDF – European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme within project POCI-01-0145-FEDER-006961, and by National Funds through the FCT – Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) as part of project UID/EEA/50014/2013. Paula Branco was supported by a scholarship from the Fundação para a Ciência e Tecnologia (FCT), Portugal (scholarship number PD/BD/105788/2014). This work was also supported by SFB-Transregio 33 'The Dark Universe' by the Deutsche Forschungsgemeinschaft (DFG), the DFG cluster of excellence 'Origin and Structure of the Universe', and the Royal Society of New Zealand Marsden Fund.

References

[1] C. Elkan, "The foundations of cost-sensitive learning," in *IJCAI'01: Proc. of 17th Int. Joint Conf. of Artificial Intelligence*, vol. 1. Morgan Kaufmann Publishers, 2001, pp. 973–978.

[2] P. Domingos, "Metacost: A general method for making classifiers cost-sensitive," in *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1999, pp. 155–164.

[3] D. B. O'brien, M. R. Gupta, and R. M. Gray, "Cost-sensitive multi-class classification from probability estimates," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 712–719.

[4] Y. Sun, M. S. Kamel, A. K. Wong, and Y. Wang, "Cost-sensitive boosting for classification of imbalanced data," *Pattern Recognition*, vol. 40, no. 12, pp. 3358–3378, 2007.

[5] R. P. Ribeiro, "Utility-based regression," Ph.D. dissertation, Dep. Computer Science, Faculty of Sciences - University of Porto, 2011.

[6] L. Torgo and R. P. Ribeiro, "Utility-based regression," in *PKDD'07: Proc. of 11th European Conf. on Principles and Practice of Knowledge Discovery in Databases*. Springer, 2007, pp. 597–604.

[7] E. Frank and R. R. Bouckaert, "Conditional density estimation with class probability estimators," in *Asian Conference on Machine Learning*. Springer, 2009, pp. 65–81.

[8] M. M. Rau, S. Seitz, F. Brimioulle, E. Frank, O. Friedrich, D. Gruen, and B. Hoyle, "Accurate photometric redshift probability density estimation–method comparison and application," *Monthly Notices of the Royal Astronomical Society*, vol. 452, no. 4, pp. 3710–3725, 2015.

[9] B. W. Silverman, *Density estimation for statistics and data analysis*. CRC press, 1986, vol. 26.

[10] P. Domingos, "Knowledge acquisition from examples via multiple models," in *Machine Learning - International Workshop Then Conference -*. Morgan Kaufmann Publishers, INC., 1997, pp. 98–106.

[11] E. Dimitriadou, K. Hornik, F. Leisch, D. Meyer, and A. Weingessel, *e1071: Misc Functions of the Department of Statistics (e1071), TU Wien*, 2011.

[12] A. Liaw and M. Wiener, "Classification and regression by randomforest," *R News*, vol. 2, no. 3, pp. 18–22, 2002.

[13] C. X. Ling and V. S. Sheng, "Cost-sensitive learning," in *Encyclopedia of Machine Learning*. Springer, 2011, pp. 231–235.

[14] C. X. Ling, Q. Yang, J. Wang, and S. Zhang, "Decision trees with minimal costs," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 69.

[15] J. H. Orallo, "Probabilistic reframing for cost-sensitive regression," in *ACM Transactions on Knowledge Discovery from Data*, vol. 8, no. 4. Association for Computing Machinery (ACM), 2014, pp. 1–55.

[16] H. Zhao, A. P. Sinha, and G. Bansal, "An extended tuning method for cost-sensitive regression and forecasting," *Decision Support Systems*, vol. 51, no. 3, pp. 372–383, 2011.

[17] G. Bansal, A. P. Sinha, and H. Zhao, "Tuning data mining methods for cost-sensitive regression: a study in loan charge-off forecasting," *Journal of Management Information Systems*, vol. 25, no. 3, pp. 315–336, 2008.

[18] P. Branco, L. Torgo, and R. P. Ribeiro, "A survey of predictive modeling on imbalanced domains," *ACM Computing Surveys (CSUR)*, vol. 49, no. 2, p. 31, 2016.

[19] B. Krawczyk, "Learning from imbalanced data: open challenges and future directions," *Progress in Artificial Intelligence*, vol. 5, no. 4, pp. 221–232, 2016.