RUTCOR
RESEARCH
REPORT

Pseudo-Boolean Regression

Tibérius O. Bonates[a] Peter L. Hammer[b]

RRR 3-2007, January, 2007

RUTCOR
Rutgers Center for
Operations Research
Rutgers University
640 Bartholomew Road
Piscataway, New Jersey
08854-8003
Telephone: 732-445-3804
Telefax: 732-445-5472
Email: rrr@rutcor.rutgers.edu
http://rutcor.rutgers.edu/∼rrr

[a]RUTCOR – Rutgers Center for Operations Research, Rutgers University, 640 Bartholomew Road, Piscataway, NJ 08854-8003
[b]RUTCOR – Rutgers Center for Operations Research, Rutgers University, 640 Bartholomew Road, Piscataway, NJ 08854-8003

Rutcor Research Report
RRR 3-2007, January, 2007

Pseudo-Boolean Regression

Tibérius O. Bonates Peter L. Hammer

Abstract. In numerous problems of data analysis the given data consists of a set $\mathbf{X}$ of binary n-vectors $\mathbf{x}^i, i = 1, \ldots, m$, along with the real values $r(\mathbf{x}^i)$ of an unknown pseudo-Boolean "target function" $r : \mathbf{X} \to \mathbb{R}$. The problem of finding an L_1-best linear approximation of r on $\mathbf{X}$ can be solved by linear programming. We show in this study that a better approximation of r on $\mathbf{X}$ can be found as a linear combination of monomials of bounded degree – a problem solvable by an iterated integer programming technique. In the iterative process additional monomials are introduced gradually into the expression of the approximant with the aim of compensating for gaps. It is shown on a series of publicly available benchmark problems that the correlation between the values of the proposed approximant and those of r on $\mathbf{X}$ is close to 1.0 on each one of the analyzed datasets. We also present results of real-life applications of the proposed methodology.

Acknowledgements: The first author gratefully acknowledges the partial support of a graduate student support award from DIMACS, Center for Discrete Mathematics and Theoretical Computer Science.

1 Introduction

Linear regression is a frequently used tool in statistics and machine learning. Given a set of m real vectors $\mathbf{X} = (\mathbf{x}^1, \ldots, \mathbf{x}^m)$ and a so-called *target function* $r : \mathbf{X} \to \mathbb{R}$, whose expression is unknown, we want to find a function $f(\mathbf{x}) = \beta_0 + \sum_{j=1}^{n} \beta_j \mathbf{x}_j$ that approximates the values of target function r on $\mathbf{X}$ as closely as possible, with $\beta_j \in \mathbb{R},\ j = 0, \ldots, n$. The n components of $\mathbb{R}^n$ are usually called the n *independent variables* of the regression problem, while the vector $\mathbf{Y} = (r(\mathbf{x}^1), \ldots, r(\mathbf{x}^m))^\top$ is called the *dependent variable*.

A commonly used criterion to measure how closely f approximates the values of r is the *least absolute residuals* (LAR) criterion, which measures the sum of the absolute values of the residuals $r(\mathbf{x}^i) - f(\mathbf{x}^i)$, i.e. $\sum_{i=1}^{m} |r(\mathbf{x}^i) - f(\mathbf{x}^i)|$. Finding $\beta = (\beta_0, \beta_1, \ldots, \beta_n)^\top$ that minimize the LAR criterion can be done by solving the following optimization problem:

$$\text{minimize} \quad \sum_{i=1}^{m} |e_i| \quad \text{subject to:} \quad \beta_0 + \sum_{j=1}^{n} \beta_j \mathbf{x}_j^i + e_i = r(\mathbf{x}^i), \quad i = 1, \ldots, m, \tag{1}$$

where $\beta \in \mathbb{R}^{n+1}$ and $e_i \in \mathbb{R}$, for every $\mathbf{x}^i \in \mathbf{X}$. It can be seen that with a proper rewriting of its constraints, problem (1) becomes a linear program.

In many cases, a linear function in the original independent variables is not suitable to approximate r. To overcome this problem one can apply a nonlinear regression technique, or map the data points to a new space, in which a hopefully more accurate linear fit is possible. In Section 2 we briefly discuss such alternatives.

In this paper we propose a technique for dealing with regression problems on binary datasets. Since in real-life applications the dataset $\mathbf{X}$ is frequently not binary, we introduce in Section 3 a binarization technique for mapping a non-binary dataset into $\{0,1\}^N$. We shall also present empirical evidence that regression results in this space are typically superior to those in the original space. Also in Section 3 we define an extended space of Boolean conjunctions, and show that the results of various regression techniques carried out in this space are further improved as compared with the original or the binarized spaces. Section 4 describes an algorithm for iteratively constructing an optimal set of conjunctions that defines an extended space in which the LAR criterion is minimized. We present the results of computational experiments performed on standard benchmark problems available from the Web. The paper ends with a section of conclusions.

2 Related Work

Much work has been devoted to the study of linear regression. Indeed, a linear function of the independent variables frequently permits a satisfactory approximation of the original function r. In many cases, the original variables are not enough to describe the underlying phenomenon and certain functions of the original variables must be used in conjunction with the original variables in order to produce a reasonable approximation. In other cases, a

function which depends nonlinearly on the parameters β is more appropriate to approximate r. In this section we briefly discuss such alternatives.

In nonlinear regression one wants to fit to the target function r a function of the original variables that depends nonlinearly on the parameters β. For instance, if we know that r follows approximately an exponential function of variable x, then we can try to fit a function of the form $f(x) = \beta_1 e^{\beta_2 x}$ to r by adjusting the parameters β_1, and β_2 of f. In order to successfully apply nonlinear regression, one usually needs some previous knowledge about the nature of function r, as shown in the example above. In many cases, such information is not available. Moreover, depending on the function used, the optimization of a fitness criterion such as least squares or LAR may become rather complicated in the context of nonlinear regression.

Transforming the original data is an alternative to obtain more accurate approximations. It can be accomplished in some cases by making use of previous knowledge about the data or the target function. Even if such knowledge is not available, simple observation may provide valuable insights into the nature of the data and the target function. For instance, in the univariate case an exponential decay of the target function r can be easily observed in a plot of the data. By taking the logarithm of the values of r, the new points will fall around a straight line, making the data amenable to a linear fit.

The technique of *Support Vector Regression* (SVR) relies on data transformation, and has been applied with success to regression problems (see [8], [9]). The SVR algorithm implicitly maps the points in $\mathbf{X}$ to a higher-dimensional space and applies a standard regression technique on that space. This is achieved with the use of a so-called *kernel function* that allows for the regression calculations to be performed without the need to actually carry out the transforming computations. The choice of the kernel function to be used is highly dependent on the nature of the data, and different kernel functions have been developed for particular types of data. Among the most frequently used kernel functions are polynomial and radial basis functions (see [7], [8]).

In this paper, we propose a regression technique based on data transformation. Our so-called *pseudo-Boolean regression* technique extracts information from the set $\mathbf{X}$ in order to build a map $\Phi : \mathbf{X} \to \{0,1\}^N$ into a new space, whose N components consist of monomials involving the components of $\{0,1\}^n$. Problem (1) is then solved for the set $\widehat{\mathbf{X}} = \{\Phi(\mathbf{x}^i) : i = 1, \ldots, m\}$ of transformed points. This procedure can be viewed as constructing a pseudo-Boolean function $f(\mathbf{x}^i, \beta)$ on the original n variables of $\{0,1\}^n$ and fitting it to r. Note however that, although f is a nonlinear function of the original n variables, it depends linearly on the parameters β. Therefore, the pseudo-Boolean regression should be regarded as a linear regression technique.

3 Regression on Transformed Data

In this section, we shall first describe a binarization procedure for dealing with datasets containing numerical or nominal variables. In what follows we present the data transformation

which lies at the center of our approach.

In order to illustrate both the binarization and the data transformation procedures, we use a small set of five datasets. Four of the datasets (*Auto-mpg*, *Boston housing*, *Servo*, and *WPBC*) come from the UCI Machine Learning Repository [5]. The dataset *Time to failure* concerns the time to failure of a certain machinery and was extracted from [6]. All five datasets contain real-valued independent variables, and involve a real-valued target function. Table 1 presents the sizes of these datasets.

Datasets	Number of observations	Number of variables
Auto-mpg (AUTO)	398	7
Boston housing (BH)	506	13
Servo (SV)	167	12
Time to failure (TF)	24	7
WPBC (WPBC)	198	33

Table 1: Size of datasets used in experiments.

For each dataset, we recorded the results of four different regression algorithms. Three of these algorithms (Linear Least Squares Regression (LS), Multilayer Perceptron (MP), and Support Vector Regression (SVR)) are available in the Weka package [12]. We used their implementation in Weka to collect the results shown in this paper. We also present the results of linear regression using the LAR criterion, to which we refer simlpy by LAR. The LAR algorithm was implemented using the Xpress linear programming solver [3]. The results of each algorithm is presented in terms of the correlation coefficient between the resulting function f and the target function r over $\mathbf{X}$, and of its *Mean Absolute Error* (MAE), computed as the average of the absolute residuals $|f(\mathbf{x}^i) - r(\mathbf{x}^i)|$ over all $\mathbf{x}^i \in \mathbf{X}$. Tables 2 and 3 summarize the results of applying each of the regression algorithms to the original (non-binarized) datasets.

	Correlation				
Algorithm	AUTO	BH	SV	TF	WPBC
LR	0.902	0.845	0.747	0.980	0.682
LAR	0.897	0.837	0.724	0.977	0.651
MP	0.947	0.957	0.987	0.999	0.837
SVR	0.901	0.849	0.724	0.996	0.625

Table 2: Correlation of regression algorithms on original datasets.

3.1 Binarization

In most real-life applications a substantial part of the data is numerical. We describe below a simple binarization procedure for converting a numerical independent variable into one or

	Mean Absolute Error				
Algorithm	AUTO	BH	SV	TF	WPBC
LR	2.538	3.287	0.846	306.823	20.192
LAR	2.467	3.118	0.725	260.979	19.318
MP	2.224	2.169	0.388	44.617	16.764
SVR	2.447	3.083	0.725	140.756	20.815

Table 3: Mean absolute error of regression algorithms on original datasets.

more binary ones, while trying to preserve the information available in the original variable. The binarization algorithm described here is based on the one introduced in [2].

Let us denote by $x_1, \ldots, x_n$ the numerical independent variables of $\mathbf{X}$. For each variable x_j, we construct a sorted list L_j of the values that x_j takes in $\mathbf{X}$. For each two consecutive values $u < v$ in L_j let us introduce a cutoff value of $\frac{u+v}{2}$, called a *cutpoint*, whenever $v-u > \tau$, where τ is a predefined tolerance value that determines whether two actual values of x_j are "too close". We say that $\frac{u+v}{2}$ *separates* the points with $x_j > \frac{u+v}{2}$ from those with $x_j < \frac{u+v}{2}$. If $v - u \leq \tau$ then the two values u and v are so close that the introduction of a cutpoint between them is not justified: if the measurement of the values of x_j is subject to some small imprecision, then, in practice, we are not able to distinguish the values u and v. In our computational tests, the value of τ is defined as a fixed percentage p of the standard deviation of variable x_j, the same value p being used for all variables $x_1, \ldots, x_n$.

We associate to each cutpoint c (corresponding to variable x_j) an indicator variable y_c such that

$$y_c = \begin{cases} 1, & \text{if } x_j > c \\ 0, & \text{otherwise.} \end{cases}$$

With this notation we now have a binary description of each point of $\mathbf{X}$ in terms of indicator variables. The rationale behind the binarization process is that points in $\mathbf{X}$ that are significantly different with respect to the original numerical variables should have significantly distinct representations in terms of indicator variables as well. In general the set of cutpoints generated in the way described above can be very large, and frequently a small subset of the cutpoints suffices to map $\mathbf{X}$ to a binary representation where the differences between significantly distinct points are preserved to a reasonable extent. It will become clear in Section 4 that the overall performance of the pseudo-Boolean regression algorithm is directly linked to the number of indicator variables actually used. In view of this, we show how to formulate a set covering problem whose solution provides a minimum-size set of cutpoints that still preserves the original information to a predefined extent.

Let us consider the entire set $C = \{c_1, \ldots, c_{|C|}\}$ of cutpoints generated by the procedure described above, and let $I = \{1, \ldots, |C|\}$ be the index set of those cutpoints. Let z_j be a binary decision variable associated to the inclusion or not of the j-th cutpoint in the definitive set of cutpoints to be used. For each pair of data points having distinct values of r we would like to have at least one of the cutpoints that separates them (if any) taken as part of the definitive set. At the same time, for the sake of improved computational performance we

want the set of cutpoints chosen to be as small as possible. This gives rise to the following minimum set cover problem:

$$\begin{aligned} \text{(SC)} \quad \text{minimize} \quad & \sum_{k=1}^{|C|} z_k \\ \text{subject to:} \quad & \\ & \sum_{k \in D(\mathbf{x}^i, \mathbf{x}^j)} z_k \geq 1, \ \ \forall\, \mathbf{x}^i, \mathbf{x}^j \in \mathbf{X}, \ \ r(\mathbf{x}^i) \neq r(\mathbf{x}^j) \qquad (2) \\ & z_k \in \{0,1\}, \ \ j = 1, \ldots, |C|, \end{aligned}$$

where $D(\mathbf{x}^i, \mathbf{x}^j) \subseteq I$ is the index set corresponding to the cutpoints that separate points $\mathbf{x}^i$ and $\mathbf{x}^j$. Note that, in the case of distinct data points that have the same value of the target function, we are not interested in forcing them to be mapped to distinct binary points. An important variant of this problem is the one in which the right-hand side of constraints (2) is replaced by an integer greater than 1. This is equivalent to asking for a more robust separation of distinct points and is an important parameter of this procedure.

Several real-life datasets contain variables that are neither binary nor numerical, but instead assume one of a set of nominal, or categorical, values. Let $S = \{s_1, \ldots, s_{|S|}\}$ be the set of possible values that a certain nominal variable x_j can assume. In some cases, there is a total order between the possible values, for instance: *low*, *medium*, and *high*. In such a case, the binarization of x_j can be done by associating a sequence of $\lceil log|S| \rceil$ binary variables to it in such a way that each value s_i of x_j is represented in binarized form simply as the binary representation of the integer $i-1$. When no total order among the elements of S is available, then the binarization of x_j can be accomplished by associating to it a set of $|S|$ binary variables, each one of which represents a possible value of x_j. In the resulting binarized representation of x_j exactly one of the associated binary variables would have value 1.

It is clear that for very large datasets solving (SC) may become prohibitively expensive. In such cases, one can apply a heuristic, such as the one in [10], to find a relatively small set of cutpoints that are reasonably informative. Another alternative is to select a large value of τ, resulting in the generation of a small set of cutpoints that detect only the most pronounced discrepancies in the values of the original variables. In the experiments presented in this section, we chose to solve (SC) by the usual greedy heuristic [10], due to its satisfactory performance and to the fact that solving (SC) to optimality is not practically justified [4].

Table 4 shows the number of indicator variables generated for each dataset. Tables 5 and 6 presents regression results for the binarized datasets. The binarization procedure was carried out with right-hand side equal to 1 for constraints (2), and with values of τ in the range of 1% to 20 % of the standard deviation. Note that the regression results on the binarized datasets are superior than those on the original datasets (Tables 2 and 3), suggesting that the binarization process did not cause significantly loss of information, while actually making it easier for the regression algorithms to approximate the target function values.

Datasets	Number of observations	Number of variables	
		Original	Binarized
Auto-mpg (AUTO)	398	7	91
Boston housing (BH)	506	13	90
NIH severity score (NSS)	78	3	75
Servo (SV)	167	12	17
Time to failure (TF)	24	7	43
WPBC (WPBC)	198	33	76

Table 4: Sizes of datasets used in experiments: original and binarized.

	Correlation				
Algorithm	AUTO	BH	SV	TF	WPBC
LR	0.948	0.923	0.871	0.983	0.714
LAR	0.931	0.901	0.829	0.967	0.628
MP	0.991	0.986	0.996	1.000	0.993
SVR	0.950	0.921	0.829	1.000	0.679

Table 5: Correlation of regression algorithms on binarized datasets.

3.2 Conjunction-space

We assume from now on that the given dataset is binary, i.e. $\mathbf{X} \subset \{0,1\}^n$. If it is not, then the binarization procedure described in the previous section can be used. The main transformation step in our algorithm consists in creating conjunctions, or monomials, involving the n binary variables of $\mathbf{X}$ and their negations $(x_1, \ldots, x_n, \overline{x}_1, \ldots, \overline{x}_n)$ and incorporating them into an extended representation $\widehat{\mathbf{X}}$ of $\mathbf{X}$. We say that a conjunction *covers* a point in $\mathbf{X}$ if it takes value 1 (true) on that point. The *coverage* of a conjunction is simply the number of points of $\mathbf{X}$ covered by it.

Let $\mathcal{C} = \{C_1, \ldots, C_{|\mathcal{C}|}\}$ be a set of conjunctions in n binary variables. The extended dataset $\widehat{\mathbf{X}}$ associated to $\mathcal{C}$ is constructed by mapping each point $\mathbf{x}^i$ in $\mathbf{X}$ to the characteristic vector of conjunctions covering $\mathbf{x}^i$, i.e. $\big(C_1(\mathbf{x}^i), \ldots, C_{|\mathcal{C}|}(\mathbf{x}^i)\big)$, where $\mathcal{C}_j(\mathbf{x}) = 1$ if $\mathcal{C}_j$ covers $\mathbf{x}$, and $\mathcal{C}_j(\mathbf{x}) = 0$ otherwise.

	Correlation				
Algorithm	AUTO	BH	SV	TF	WPBC
LR	1.808	2.375	0.569	161.477	18.907
LAR	1.632	2.092	0.478	126.875	16.636
MP	0.822	1.374	0.408	11.515	3.221
SVR	1.537	2.022	0.479	7.511	16.140

Table 6: Mean absolute error of regression algorithms on binarized datasets.

Ideally, we would like to apply a regression algorithm on the set of all conjunctions in n variables. Since this set includes all literals, the regression results on the corresponding extended space are at least as good as those on binary variables. However, the number of such conjunctions is clearly very large, even for datasets of relatively small sizes. In the next section we present an algorithm for constructing an optimal set of conjunctions $\mathcal{C}^*$ of arbitrary degree, such that the LAR criterion is minimized on the resulting extended dataset.

4 Constructing an Optimal Conjunction-Space

As discussed in the previous section, projecting a dataset on a large conjunction-space can lead to more accurate regression results. However, it is conceivable that only a relatively small set of conjunctions may be really needed to obtain an approximation of good quality to the target function. Therefore, the computational effort of generating *all* conjunctions satisfying certain requirements and of applying a regression algorithm on a very large set of variables is not justified.

Let $\mathbb{C}$ denote the set of all conjunctions satisfied by at least one point in $\mathbf{X}$. In this section, we propose a column generation algorithm for constructing an optimal pool of conjunctions, i.e. a set of conjunctions $\mathcal{C} \subset \mathbb{C}$ that allows an approximation of the target function which is as accurate as the one that one would obtain by using $\mathbb{C}$. It is natural to expect that, in many cases, a set $\mathcal{C}$ can be found such that $|\mathcal{C}| \ll |\mathbb{C}|$.

Rewriting (1) as a linear program we obtain:

$$
\begin{aligned}
\text{(R)} \qquad \text{minimize} \quad & \sum_{i=1}^{m} e_i \\
\text{subject to:} \quad & \\
& e_i + \beta_0 + \sum_{j=1}^{n} \beta_j \mathbf{x}^i_j \geq \quad r(\mathbf{x}^i), \;\; i = 1, \ldots, m \\
& e_i - \beta_0 - \sum_{j=1}^{n} \beta_j \mathbf{x}^i_j \geq -r(\mathbf{x}^i), \;\; i = 1, \ldots, m, \\
& e_i \geq 0, \;\; i = 1, \ldots, m,
\end{aligned}
$$

where each variable e_i corresponds to the absolute value of the error $r(\mathbf{x}^i) - f(\mathbf{x}^i)$, $i = 1, \ldots, m$. Let $\mathcal{C} = \{C_1, \ldots, C_{|\mathcal{C}|}\} \subset \mathbb{C}$, and let us denote by (R($\mathcal{C}$)) the variant of problem (R) in which we seek an LAR-optimal approximation of function r in the truncated transformed

space $\{0,1\}^{|\mathcal{C}|}$:

$$
\begin{aligned}
(\mathrm{R}(\mathcal{C})) \quad & \text{minimize} \quad \sum_{i=1}^{m} e_i \\
& \text{subject to:} \\
& e_i + \beta_0 + \sum_{j=1}^{|\mathcal{C}|} \beta_j C_j(\mathbf{x}^i) \geq \quad r(\mathbf{x}^i), \;\; i = 1, \ldots, m \qquad (3) \\
& e_i - \beta_0 - \sum_{j=1}^{|\mathcal{C}|} \beta_j C_j(\mathbf{x}^i) \geq -r(\mathbf{x}^i), \;\; i = 1, \ldots, m \qquad (4) \\
& e_i \geq 0, \;\; i = 1, \ldots, m. \qquad (5)
\end{aligned}
$$

Let us associate to constraints (3) and (4) the vectors of dual variables λ and μ, respectively. Let $\mathcal{C}^0 = \{x_1, \ldots, x_n\}$ be the initial set of conjunctions we use in our algorithm, corresponding to the n binary variables. Let (e^*, β^*) be the optimal solution of $(\mathrm{R}(\mathcal{C}^0))$, and let (λ^*, μ^*) be the corresponding dual variables. In order to find a conjunction from $\mathbb{C} \setminus \mathcal{C}^0$ whose inclusion in $\mathcal{C}^0$ would improve the objective function of $(\mathrm{R}(\mathcal{C}^0))$ we should compute the so-called *reduced costs* of the conjunctions in $\mathbb{C} \setminus \mathcal{C}^0$.

By standard linear programming manipulations, we can find that the reduced cost of a conjunction $C_j \in \mathbb{C} \setminus \mathcal{C}^0$ is given by $-(\lambda^* - \mu^*)^\top a_j$, where a_j is the characteristic vector $(C_j(\mathbf{x}^1), \ldots, C_j(\mathbf{x}^m))^\top$ of the set of points covered by C_j. We are interested in finding a conjunction whose reduced cost is negative. The following pseudo-Boolean optimization problem finds the conjunction with the most negative reduced cost

$$
\begin{aligned}
(\mathrm{S}(\lambda^*, \mu^*)) \quad & \text{maximize} \quad \sum_{k=1}^{m} \left(\prod_{i:\mathbf{x}_i^k=0} \overline{p_i} \prod_{j:\mathbf{x}_j^k=1} \overline{p_j^c} \right) (\lambda_k^* - \mu_k^*) \\
& \text{subject to:} \quad p_j, p_j^c \in \{0,1\}, \; j = 1, \ldots, n,
\end{aligned}
$$

where the binary decision variable p_j corresponds to the inclusion or not of literal x_j in the resulting conjunction, binary variable p_j^c corresponds to the inclusion or not of literal $\overline{x}_j$ in the resulting conjunction, and each term $\prod_{i:\mathbf{x}_i^k=0} \overline{p_i} \prod_{j:\mathbf{x}_j^k=1} \overline{p_j^c}$ takes value 1 whenever $\mathbf{x}^k$ is covered by the resulting conjunction, and 0 otherwise.

Once a conjunction $C_j \in \mathbb{C} \setminus \mathcal{C}^0$ with negative reduced cost has been found, it is introduced into the set of conjunctions under consideration, and an augmented version of problem $(\mathrm{R}(\mathcal{C}^0))$ is solved. The algorithm proceeds with the alternate solution of an instance of problem $(\mathrm{R}(\mathcal{C}))$, for some $\mathcal{C} \subset \mathbb{C}$, and an instance of problem (S). In general, at iteration i we solve $(\mathrm{R}(\mathcal{C}^i))$, obtain the dual variables λ^* and μ^* associated to its optimal solution, and solve the associated problem $(\mathrm{S}(\lambda^*, \mu^*))$, obtaining a conjunction C_j. If the optimal objective function value of $(\mathrm{S}(\lambda^*, \mu^*))$ is strictly positive, then we define $\mathcal{C}^{i+1} := \mathcal{C}^i \cup \{C_j\}$ and go on

to the next iteration. Otherwise, the approximating function f constructed at the current iteration is optimal with respect to the LAR criterion.

Problem $(S(\lambda^*, \mu^*))$ can be reformulated as a linear integer program by introducing a binary variable corresponding to each term in the function and associated constraints. A clear description of this linearization is found in [1].

In Tables 7 and 8 we present the correlations and mean absolute errors of the four algorithms LS, LAR, MP and SVR on the original non-binarized datasets (as shown in Tables 2 and 3), along with correlations and mean absolute errors of the pseudo-Boolean algorithm applied to the same datasets. Note that the application of the pseudo-Boolean regression algorithm involves the stages of binarization and construction of an optimal pool of conjunctions as described in the previous section. The optimal sets of conjunctions generated contained less than 200 conjunctions in each of the cases.

	Correlation				
Algorithm	AUTO	BH	SV	TF	WPBC
LS	0.902	0.845	0.747	0.980	0.682
LAR	0.897	0.837	0.724	0.977	0.651
MP	0.947	0.957	0.987	0.999	0.837
SVR	0.901	0.849	0.724	0.996	0.625
PBR	0.990	0.990	0.990	1.000	0.971

Table 7: Correlation of regression algorithms on original data and of pseudo-Boolean regression algorithm.

	Mean Absolute Error				
Algorithm	AUTO	BH	SV	TF	WPBC
LS	2.538	3.287	0.846	306.823	20.192
LAR	2.467	3.118	0.725	260.979	9.318
MP	2.224	2.169	0.388	44.617	16.764
SVR	2.447	3.083	0.725	140.756	20.815
PBR	0.417	0.581	0.076	0.000	1.536

Table 8: Mean absolute error of regression algorithms on original data and of pseudo-Boolean regression algorithm.

5 Conclusions

In addition to the superior results in terms of correlation and mean absolute error, we identify two other convenient properties of the algorithm proposed here. First, the conjunctions in the resulting pseudo-Boolean approximant may suggest relations between the original variables

that may not be apparent at a first moment. Due to the binarization procedure, such relations are of a different nature than most of the relations one can derive from standard regression techniques. We remark here that the colum generation algorithm described in Section 4 can be applied in a more general context, in which the original variables (and possibly functions of those variables) can be used to approximate the r-values together with a set of conjunctions. In this setting, the algorithm would construct a set of conjunctions that best complements the given numerical variables in the task of predicting the values of r. Second, in our experiments we limited the degree of the conjunctions generated to be at most 3, and in some cases at most 2. In spite of this limitation, the resulting function approximated closely the target function values in each case.

References

[1] Bonates T.O., P. L. Hammer, A. Kogan. Maximum Patterns in Datasets. *RUTCOR Research Report 9-2006*, 2006.

[2] Boros E., P.L. Hammer, T. Ibaraki, A. Kogan. Logical Analysis of Numerical Data. *Mathematical Programming*, 79:163–190, 1997.

[3] Dash Associates. Xpress-Mosel Reference Manuals and Xpress-Optimizer Reference Manual, Release 2004G, 2004.

[4] T. Dietterich. Overfitting and undercomputing in machine learning. *ACM Computing Surveys*, 27:326–327, 1995.

[5] C.L. Blake D.J. Newman, S. Hettich and C.J. Merz. UCI repository of machine learning databases, 1998.

[6] Lin, D., M. Wiseman, D. Banjevic, A.K.S. Jardine. An approach to signal processing and condition-based maintenance for gearboxes subject to tooth failure. *Mechanical Systems and Signal Processing*, 18:993–1007, 2004.

[7] Scholkopf B.,A. J. Smola. *Learning with kernels: Support vector machines, regularization, optimization, and beyond.* MIT Press, Cambridge, MA, 2002.

[8] Shawe-Taylor J., N. Cristianini. *Kernel Methods for Pattern Analysis.* Cambridge University Press, New York, NY, 2004.

[9] A.J. Smola and B. Schoelkopf. A tutorial on support vector regression. NeuroCOLT2 Technical Report NC2-TR-1998-030, 1998.

[10] Chvátal V. A greedy heuristic for the set covering problem. *Mathematics of Operations Research*, 4:233–235, 1979.

[11] Chvátal V. *Linear Programming.* W. H. Freeman, 1983.

[12] Witten I.H., E. Frank. *Data Mining: Practical machine learning tools with Java implementations*. Morgan Kaufmann, San Francisco, 2000.