Pattern-Aided Regression Modeling and Prediction Model Analysis

Guozhu Dong, *Senior Member, IEEE* and Vahid Taslimitehrani

Abstract—This paper first introduces *pattern aided regression* (PXR) models, a new type of regression models designed to represent accurate and interpretable prediction models. This was motivated by two observations: (1) Regression modeling applications often involve complex *diverse predictor-response relationships*, which occur when the optimal regression models (of given regression model type) fitting two or more distinct logical groups of data are highly different. (2) State-of-the-art regression methods are often unable to adequately model such relationships. This paper defines PXR models using several patterns and local regression models, which respectively serve as logical and behavioral characterizations of distinct predictor-response relationships. The paper also introduces a *contrast pattern aided regression* (CPXR) method, to build accurate PXR models. In experiments, the PXR models built by CPXR are very accurate in general, often outperforming state-of-the-art regression methods by big margins. Usually using (a) around seven simple patterns and (b) linear local regression models, those PXR models are easy to interpret; in fact, their complexity is just a bit higher than that of (piecewise) linear regression models and is significantly lower than that of traditional ensemble based regression models. CPXR is especially effective for high-dimensional data. The paper also discusses how to use CPXR methodology for analyzing prediction models and correcting their prediction errors.

Index Terms—Correlation and regression analysis, model validation and analysis, error analysis, data mining, mining methods and algorithms

1 Introduction

Constructing *accurate numerical prediction models* is fundamental for many modeling and forecasting applications, including scientific and medical modeling, and economic and severe weather forecasting. However, state-of-the-art regression algorithms, including piecewise regression [28], support vector regression (SVR) [13], random forest [6], and Bayesian additive regression trees (BART) [10], often *fail to produce highly accurate* models (see [10] and Section 6). Moreover, the prediction models they produce can be *hard to interpret*, since they often are ensembles of hundreds of component structures or use many derived variables.

One possible reason why state-of-the-art regression algorithms often fail to produce very accurate prediction models is because they cannot adequately model *diverse predictor-response variable relationships*. By "diverse predictor-response relationships" we mean "the predictor-response relationships fitting distinct logical groups[1] of data are highly different". The regression models representing those relationships often emphasize highly distinct sets of variables and their prediction values often differ significantly. Section 4 uses real data to illustrate diverse predictor-response relationships. Experiments in this paper (see Section 6) and the widespread usage of piecewise regression[2] [28] both confirm that such relationships happen frequently.

To meet the challenge caused by diverse predictor-response relationships, we need (1) a new type of regression models that (a) can adequately represent complex diverse predictor-response relationships and at the same time (b) outperform popular regression model types and (c) are easy to interpret, and we need (2) new algorithms for building such regression models. The purpose of this paper is to present a new type of regression models and a regression method to meet those needs.

The first key idea in our new type of regression models is to use a pattern[3] P as a logical characterization of a logical group of data, and a local regression model f_P as a behavioral characterization of the intrinsic predictor-response relationship for that group of data. The second key idea is to use a small set of patterns and associated local regression models to define a *pattern aided regression* (PXR) model. Thus, we use the combination of a pattern and its associated local model to represent one particular predictor-response relationship, and we use different patterns in the set to represent diverse predictor-response relationships that exist in an application.

PXR models are easy to interpret, since they usually use very few patterns and they use simple (e.g., linear) regression models as local regression models. PXR models' complexity is just a bit higher than that of (piecewise) linear regression (LR) models. Experiments show that PXR models can achieve

[1] Logical groups of data include those groups defined by patterns.

[2] See Section 2 for discussion on piecewise regression.

[3] A pattern is a simple condition on several variables. See Section 3.2.

• *The authors are with the Department of Computer Science and Engineering and Kno.e.sis Center, Wright State University, Dayton, OH 45435. E-mail: {guozhu.dong, taslimitehrani.2}@wright.edu.*

*Manuscript received 28 Apr. 2014; revised 7 Dec. 2014; accepted 3 Mar. 2015. Date of publication 10 Mar. 2015; date of current version 3 Aug. 2015.
Recommended for acceptance by X. Zhu.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TKDE.2015.2411609*

Authorized licensed use limited to: Dr. D. Y. Patil Educational Complex Akurdi. Downloaded on June 29,2025 at 11:50:30 UTC from IEEE Xplore. Restrictions apply.

1041-4347 © 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

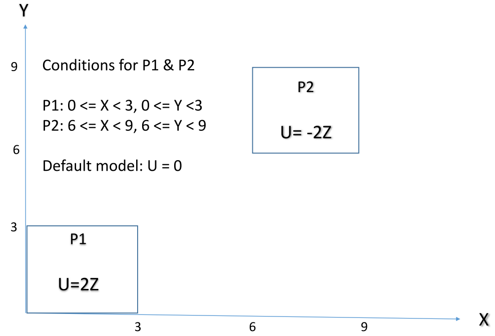

Fig. 1. Illustration of a PXR model example.

much more accurate prediction than state-of-the-art regression models with much lower model complexity.

Fig. 1 gives a pictorial description of a simple PXR model with three predictor variables, X, Y, and Z, and having U as the response variable. Only the XY plane is shown in the figure. The PXR model has two patterns, P_1, P_2, which correspond to the two square regions in the XY plane; their defining conditions and local models, as well as the default model, are given in the figure. For the XYZ vector $V = (0, 2, 5)$, which satisfies only pattern P_1, the PXR model returns the value computed by the local model of P_1, namely $U = 2Z = 10$. For the vector $V = (5, 8, 4)$, which does not satisfy any of the two patterns, the PXR model returns the value $U = 0$ using the default model. This PXR model cannot be simulated by standard linear regression, or by piecewise linear regression (PLR) using intervals over just one variable (since the piecewise linear regression cannot use intervals over x due to the difference between the models for the region of P_1 and the region above P_1, and similarly it cannot use intervals over y). For simplicity the regions of the patterns in the figure do not overlap.

Not all pattern sets can lead to desirable[4] PXR models. This paper introduces a contrast pattern aided regression method, CPXR, to compute desirable pattern sets to define high quality PXR models. CPXR's key idea is to focus on patterns that contrast the "large error (LE)" data, consisting of instances where a baseline prediction model makes large prediction errors, vs the "small error (SE)" data. This paper also introduces several quality measures and techniques to improve computational efficiency.

It turns out that using contrast patterns allows CPXR to find more accurate PXR models faster than using frequent patterns (see Section 6.7 for details). This can be explained by two factors: First, contrast patterns with high support ratios often include most of the frequent patterns that can significantly reduce the residual of the baseline model. Second, the number of frequent patterns is a lot larger than the number of contrast patterns, when the same support threshold is used. We note that the start point of "contrast pattern aided regression" is the division of given dataset D into LE (instances where large errors are made) and SE (instances where small errors are made) by the baseline model; even the use of entropy based binning with respect to these classes helps us form better contrast patterns for regression modeling than binning without access to classes.

In experiments CPXR consistently outperforms, often by big margins, state-of-the-art regression methods such as linear regression, piecewise linear regression, Bayesian additive regression trees, gradient boosting (GBM), and support vector regression. CPXR has good performance on both high and low dimensional data.

In addition to producing highly accurate models, CPXR can identify several patterns to capture most of the data instances where a given prediction model makes large prediction errors, and CPXR can build and utilize local regression models to correct those prediction errors. CPXR has potential in identifying/correcting errors in important scientific/medical/economic models, as well as in characterizing key differences between two given prediction models. CPXR's ideas regarding utilizing/selecting patterns can be useful elsewhere.

The major contributions of this paper include: (a) It articulates the diverse predictor-response relationship phenomenon. It introduces (b) a novel type of regression models (PXR), and (c) a regression method (CPXR) to build accurate PXR models, for effective modeling of diverse predictor-response relationships. (d) CPXR is also useful for analyzing prediction models and correcting their prediction errors, and for identifying multi-variable interactions. PXR/CPXR are especially effective for high-dimensional data.

In the rest of the paper, Section 2 discusses related works. Section 3 presents the PXR concepts and examples, besides the preliminaries. Section 4 uses examples to discuss diverse predictor-response relationships. Section 5 presents our CPXR algorithm, together with two useful quality measures used by that algorithm. Section 6 reports a systematic experimental evaluation. Section 7 offers concluding remarks and discusses how to use CPXR to analyze prediction models.

2 Related Work

Since regression modeling is a vast area of research, it is not realistic to discuss all related papers here; we focus on the most related representatives.

Representative model types and methods for regression include the following. In support vector regression [13], a prediction model is constructed in a manner similarly to that for support vector machines [11], except that SVR minimizes regression error instead of classification error. Often using kernel functions (and many derived variables), SVR models are hard to interpret. In Bayesian additive regression trees [10], a prediction model is an ensemble of decision trees. Often using hundreds of decision trees, BART models are hard to interpret. The gradient boosting [19] ensemble method iteratively identifies instances where the models built so far are inaccurate and then builds a new model by focusing attention on those instances. A key difference between GBM and CPXR is that GBM treats all incorrectly predicted instances in a uniform manner, while CPXR focuses on pattern-defined data groups that have accurate prediction models correcting errors of a given baseline model. Experiments show that CPXR is more accurate than SVR, GBM, and BART. As discussed elsewhere, CPXR models are fairly easy to interpret.

[4] As will be seen later, a pattern set will make a highly accurate PXR model if the local model of each pattern in the set is fairly accurate and different patterns in the set are complementary to each other, so that they combine to define an accurate regression model on all data.

Authorized licensed use limited to: Dr. D. Y. Patil Educational Complex Akurdi. Downloaded on June 29,2025 at 11:50:30 UTC from IEEE Xplore. Restrictions apply.

TABLE 1
Symbols

Symbol	Meaning
arr	average residual reduction
BART	Bayesian Additive Regression Trees
CPXR	Contrast Pattern Aided Regression
CPXR(LL)	CPXR using LR to build baseline models and LR to build local regression models
CPXR(LP)	CPXR using LR to build baseline models and PLR to build local regression models
EC	equivalence class
f	regression function/model
f_{PM}	regression function of PM
GBM	Gradient Boosting
LE	set of large error instances
LR	Linear Regression Algorithm
MG	minimal generator
mds	matching data set
P	pattern
PLR	Piecewise Linear Regression
PM	a PXR regression model
$PM(\{P_1, \ldots, P_k\}, f_d)$	$((P_1, f_{P_1}, \mathsf{arr}(P_1)), \ldots, (P_k, f_{P_k}, \mathsf{arr}(P_k)), f_d)$
PXR	Pattern Aided Regression
RMSE	root mean square error
R^2	R squared
SE	set of small error instances
SVR	Support Vector Regression
trr	total residual reduction

Piecewise linear regression [28] is the only[5] previous method (known to us) that uses easy-to-interpret models to represent diverse predictor-response relationships, although its ability is limited (see below). Typically, the range of some special predictor variable x is divided into intervals and a separate linear regression model is used for data instances whose x values are in a given interval. A simple example of PLR model is: $y = 4 + 3x$ for $x \leq 2$ and $y = 8 + x$ for $x > 2$. PLR has been fairly widely used, including for discovering change point in time associated with policy-regime shift [5] in economic modeling, and for identifying ecological thresholds [33] in ecology modeling. References [20], [4], [24] studied how to build PLR regression models.

PXR models are more general than PLR models. In the literature PLR models typically involve intervals on just one predictor variable. While several papers in the literature mentioned the possibility/desirability of PLR models using intervals on multiple variables, we did not see any papers that provided effective algorithm for building such PLR models, perhaps due to the complexity of the problem. It appears that each PLR model using intervals on k variables includes all patterns involving all possible combinations of the k variables and their intervals, and each pattern of this model uses exactly those k variables. PXR models are more general and flexible than PLR models, since they do not have those two constraints. (Other differences include: PXR models also allow instances to satisfy multiple patterns and they use weights to combine prediction by local regression models.) From a computing perspective, the CPXR algorithm provides a systematic and effective method to search for a desirable pattern set to represent high quality PXR models. Experiments show that PXR models are more accurate than PLR models.

Reference [7] studied classification model error characterization. In contrast, our CPXR is about building accurate numerical prediction models, and characterizing/correcting errors of such prediction models.

Contrast pattern mining and application have received much attention recently (see [12]). Researchers have proposed numerous contrast pattern based methods for classification [16], clustering and clustering quality evaluation [18], [26], outlier detection [9], bioinformatics [29], [30], cancer analysis [25], chemoinformatics [1], and so on. This paper is novel in its focus on contrast pattern aided construction of accurate prediction models, and in the concept of using pattern and local regression model pairs to capture distinct predictor-response relationships involving multiple multi-variable interactions.

Recently [32] adapted the CPXR approach to perform logistic regression, producing very good results on traumatic brain injury outcome prediction.

3 Preliminaries and PXR Concepts

3.1 Preliminaries on Regression

Regression analysis aims to design *regression models* to predict the numerical values of a *response* (also called *dependent*) variable, based on values of *predictor* (also called *independent* or *explanatory*) variables. Variables are also called features and attributes. A vector of values of the predictor variables is an *instance*. Regression uses a set $\{(x_i, y_i) \mid 1 \leq i \leq n\}$ of instance and response variable value pairs as *training data*.

The performance of a prediction model f is often evaluated based on its prediction residuals. The *residual* of f on a particular instance x_i is $f(x_i) - y_i$, the difference between the predicted and observed response variable values. Two often used quality measures are *root mean square error (RMSE)* and R^2 (*R Squared*): where

$$RMSE(f) = \sqrt{\frac{\sum_{i=1}^{n} (f(x_i) - y_i)^2}{n}}$$

and $R^2(f) = 1 - \frac{\sum_{i=1}^{n} (f(x_i) - y_i)^2}{\sum_{i=1}^{n} (y_i - avg_{j=1}^{n} y_j)^2}$.

3.2 Preliminaries on Patterns and Discretization

Let D be a training data set for regression. For the pattern part, we ignore the response variable. To avoid using too many symbols, we will still denote the projection of D onto the predictor variables as D.

We usually use the entropy based method [17], to partition ranges of numerical variables into disjoint intervals (bins). Let S be a data set with two classes, C_1 and C_2. Then $entropy(S) = -p_1 log_2 p_1 - p_2 log_2 p_2$, where $p_i = \frac{|C_i|}{|S|}$. Entropy measures the purity of sets. The entropy based binning method splits an interval, say $[u, w)$, of a numerical variable A into two intervals using a split value v, selected to minimize the weighted average of entropies of $D_{<v}$ and $D_{\geq v}$,

[5] The model tree regression model [35], defined as a decision tree whose leaf nodes contain regression functions, is related to PLR and PXR. While a model tree can represent a PXR model (viewing root-to-leaf paths as patterns), its pattern set is highly restricted, since all patterns must involve the variable at the root of the tree.

Authorized licensed use limited to: Dr. D. Y. Patil Educational Complex Akurdi. Downloaded on June 29,2025 at 11:50:30 UTC from IEEE Xplore. Restrictions apply.

where $D_{<v}$ contains all instances of D whose value of variable A is in $[u, v)$, and $D_{\geq v}$ contains those instances whose A value is in $[v, w)$. The new intervals may be split further until a termination criterion [17] is met.

An *item* is a *single-variable condition* of the form "$A = a$" (or written as "A:a") if A is a categorical variable, or "$v_1 \leq A < v_2$" (or written as "A:$[v_1, v_2)$"), where v_1 and v_2 are constants, if A is numerical. A *pattern* or *itemset* is a finite set of items. An instance x is said to *satisfy*, or *match*, a pattern P, denoted by $x \models P$, if x satisfies every item/condition in P. The *matching data* of P in D is given by $\mathsf{mds}(P, D) = \{x \in D \mid x \models P\}$. We will write $\mathsf{mds}(P, D)$ as $\mathsf{mds}(P)$ if D is clear from the context. The *support* of P in D is $supp(P, D) = \frac{|\mathsf{mds}(P,D)|}{|D|}$. We may generalize the above by using a subset D' of D to replace D, as in $supp(P, D')$.

3.3 Preliminaries on Contrast Patterns

Intuitively, a pattern is a contrast pattern if its supports in different classes are very different.

Definition 1 ([15]). *Given two data classes C_1 and C_2, the* support ratio *(also called* growth rate*) of a pattern P from C_1 to C_2 is*[6] *$supp\_ratio_{C_1}^{C_2}(P) = \frac{supp(P,C_2)}{supp(P,C_1)}$. Given a support ratio threshold γ, a* contrast pattern *(also called* emerging pattern*) of class C_2 is a pattern P satisfying $supp\_ratio_{C_1}^{C_2}(P) \geq \gamma$.*

In addition to γ, we also often impose a threshold, $minSup$, on support of contrast patterns in the LE class (discussed in Section 3.4). Only contrast patterns P satisfying $supp(P, LE) \geq minSup$ are of interest.

To avoid processing redundant patterns, we partition the total set of contrast patterns into equivalence classes (EC), each consisting of all patterns sharing a common matching data set. Since patterns having the same matching data can be considered as having the same behavior, it suffices to consider just one pattern per EC. Technically, a pattern P is a member of an EC of patterns defined as $EC(P) = \{Q \mid \mathsf{mds}(Q) = \mathsf{mds}(P)\}$. It can be shown that each EC can be described by a closed pattern (the longest in the EC) and a set of minimal-generator (MG) patterns (minimal with respect to $\subseteq$); so an EC contains all patterns Q satisfying "Q is a superset of some MG and Q is a subset of the closed pattern, of the EC".

Following [26], [18], CPXR picks one shortest MG (arbitrarily) of each EC of contrast patterns to represent it, and ignores other patterns of the EC, in the pattern set selection process. This helps us gain efficiency without losing useful candidates. Moreover, the MGs are the easiest to match and interpret. The selected MGs can be viewed as conceptual identifiers of all possible matching data sets (of patterns), allowing users to see what each matching data set is about.

We now illustrate using data in Table 2. Here, we write patterns in compact form (e.g. be denotes $\{A_1 = b, A_3 = e\}$). One EC is $\{b$, i, bg, bi, gi, $bgi\}$; the EC's MGs are b and i, its closed pattern is bgi; and the matching data of the patterns is $\{t1, t2, t5\}$. Patterns in this EC are contrast pattern of C_1 with support ratio of 3; their supports in C_1 and C_2 are $1 = 2/2$ and $1/3$ respectively. Another EC is $\{be$, ei, beg, bei, egi, $begi\}$.

TABLE 2
Small Data Set with Two Classes

TID	A_1	A_2	A_3	A_4	A_5	Class
t1	b	d	e	g	i	C_1
t2	b	c	e	g	i	C_1
t3	a	c	e	g	j	C_2
t4	a	c	e	h	j	C_2
t5	b	d	f	g	i	C_2

3.4 PXR Concepts

Let $D = \{(x_i, y_i) \mid 1 \leq i \leq n\}$ be a given training data set for regression. Let f be a regression model built on D, which we will call the baseline model.

In the CPXR approach, we first split D into two classes, LE and SE, consisting of instances of D where f makes large/small prediction errors respectively. CPXR then searches for a small set of contrast patterns of LE to optimize the trr measure (see Section 5.1), and uses that set to build a PXR model. Intuitively, the search aims to find a set of patterns, such that each pattern is highly useful in correcting prediction errors of f, and when combined the patterns in the set give high prediction accuracy.

Importantly, we associate with each pattern P a *local regression model* f_P built using P's matching data, $\mathsf{mds}(P)$, as training data. There is no limit on what variables f_P can use. While any regression method can be used to build f_P, one may prefer linear regression or piecewise linear regression models, since the simplicity of such models helps ensure that PXR prediction models are easy to interpret.

Using a pattern P and its associated local regression model f_P is a flexible way to represent a *predictor-response relationship*. f_P can accurately reflect how the response variable depends on the predictor variables, on data instances satisfying P. Different (P, f_P) pairs can emphasize distinct sets of predictor variables; the set for a given (P, f_P) contains those variables used in f_P and emphasized in different ways using coefficients of different magnitudes, and those variables used in P. Each (P, f_P) pair can represent a predictor-response relationship that is highly different from the predictor-response relationships on data not satisfying P. A small set of (P, f_P) pairs can represent a diverse set of predictor-response relationships.

Definition 2. *A* pattern aided regression *model is a tuple $PM = ((P_1, f_1, w_1), \ldots, (P_k, f_k, w_k), f_d)$, where k is a positive integer, $P_1, \ldots, P_k$ are patterns*[3]*, $f_1, \ldots, f_k$, f_d are regression models, and $w_1, \ldots, w_k > 0$ are weights. $\{P_1, \ldots, P_k\}$ is the* pattern set *of PM, f_i is the* local regression model *of P_i, and f_d is the* default regression model. *We define the* regression function of *PM as*

$$f_{PM}(x) = \begin{cases} \frac{\Sigma_{P_i \in \pi_x} w_i f_i(x)}{\Sigma_{P_i \in \pi_x} w_i} & \text{if } \pi_x \neq \emptyset \\ f_d(x) & \text{otherwise} \end{cases} \quad (1)$$

for instances x, where $\pi_x = \{P_i \mid 1 \leq i \leq k, x \text{ satisfies } P_i\}$.

Remark. The PXR model uses the weighted average of the predicted values of the local models of those patterns

[6] If $supp(P, C_1) = 0$ we define the ratio as a large number such as $2|C_2|$ and we call P a jumping emerging pattern.

Authorized licensed use limited to: Dr. D. Y. Patil Educational Complex Akurdi. Downloaded on June 29,2025 at 11:50:30 UTC from IEEE Xplore. Restrictions apply.

that are satisfied by x if x is satisfied by some patterns in the PXR model, and the PXR model uses the default model's prediction as the PXR model's prediction otherwise. Illustration is given in the next section. As will be discussed later, w_i can be determined based on f_i's influence on residual reduction.

3.5 Example PXR Model on TBI

For illustration, we give an example PXR model, which will also be used later, on the TBI data set.[7]

The baseline regression model[8] is the following linear regression model built from all TBI data

$$\begin{aligned} f : y = 5.71 - 0.4\, cistern + 0.71\, ctclass - 0.55\, eDH \\ - 0.8\, hypoten - 0.53\, pupil - 0.44\, tSAH \end{aligned}$$

One PXR model uses the following seven patterns[9]:

$$\begin{aligned}
&P_1 : cause : 3, cistern : 0, ctclass : 3, hypoten : 0, pupil : 0 \\
&P_2 : cause : 3, hypoxia : 0, pupil : 0, tSAH : 1 \\
&P_3 : cause : 3, cistern : 1, pupil : 1 \\
&P_4 : cause : 3, eDH : 0, hypoten : 0, glucose : [115.5, 173.3) \\
&P_5 : cause : 9, cistern : 1, hypoten : 0, pupil : 3 \\
&P_6 : cause : 3, hypoxia : 0, shift : 0, glucose : [173.3, 231) \\
&P_7 : cause : 5, ctclass : 3, hypoten : 1, age : [23, 43)
\end{aligned}$$

The local regression models for the patterns are[8]

$$\begin{aligned}
f_{P_1} : y &= 8.6 - 2.32\, eDH - 0.24\, hypoxia - 0.33\, motor \\
&\quad - 0.65\, pH - 1.03\, sodium + 5.16\, tSAH \\
f_{P_2} : y &= 4.8 - 0.12\, age - 1.32\, ctclass + 0.30\, cistern \\
&\quad + 0.26\, eDH - 0.19\, ipupil - 2.17\, pH - 0.48\, sodium \\
f_{P_3} : y &= 6.18 - 1.87\, age - 4.43\, ctclass + 1.21\, eDH \\
&\quad - 0.31\, ipupil - 3.92\, pH - 2.46\, sodium - 8.89\, tSAH \\
f_{P_4} : y &= 7.76 + 5.2\, age + 0.46\, cistern - 6.41\, ctclass \\
&\quad + 1.41\, ipupil + 3.49\, pH - 1.28\, sodium + 1.11\, tSAH \\
f_{P_5} : y &= 130 - 5.59\, age + 2.73\, ctclass + 0.98\, motor \\
&\quad + 13.66\, pH + 3.22\, sodium + 55.44\, tSAH \\
f_{P_6} : y &= -204.42 + 3\, age + 0.58\, cistern - 1.76\, eDH \\
&\quad + 1.13\, ipupil + 0.96\, pupil - 6\, pH + 1.67\, sodium \\
&\quad + 23.6\, tSAH \\
f_{P_7} : y &= 240.5 - 16\, age - 5.21\, eDH - 3.17\, ipupil \\
&\quad + 6.76\, pH - 5.44\, sodium - 5.42\, tSAH \\
f_d : y &= 12.7 - 7.3\, age + 1.1\, cistern + 1.39\, pupil \\
&\quad + 1.71\, sodium + 10.01\, tSAH.
\end{aligned}$$

The PXR model is $PM = ((P_1, f_{P_1}, w_1), \ldots, (P_7, f_{P_7}, w_7), f_d)$ where $w_1, \ldots, w_7$ (given by arr) are 19.8, 7.5, 78.5, 57.2, 94.1, 58.1, 95.9 respectively. Regarding accuracy, f's RMSE is 10.45, and its R^2 is 0.29; f_{PM}'s RMSE is 3.51, and its R^2 is 0.85. So, the PXR model f_{PM} reduces f's RMSE by 67 percent, and improves f's R^2 by 193 percent.

4 Diverse Predictor-Response Relationships

We now use the TBI example to illustrate the concepts about diverse predictor-response relationships. By examining the patterns and the local regression models given in Section 3.5, we get the following observations.

a) The local regression models often give high importance to variables[10] that got low importance in the baseline model f. For example, the local model f_{P_1} gives a high importance to $sodium$ which received very low importance in the baseline model f. Moreover, for each i satisfying $2 \leq i \leq 7$, age received high importance in f_{P_i} but very low importance in f.

b) The seven patterns are fairly different from each other regarding their conditions. While distinct patterns may share some conditions on certain individual variables, there are typically two or more such conditions unique to each pattern.

c) Different local regression models often emphasize different variables. For example, f_{P_1} uses large coefficient on $tSAH$ while f_{P_2} uses large (in absolute value) coefficient on $ctclass$.

d) The largest coefficients of the local regression models are quite different. For example, it is 2.17 in f_{P_2}, whereas it is 55.44 in f_{P_5}.

The above factors indicate that the patterns and local regression models represent diverse predictor-response variable relationships in the TBI data.

4.1 Challenges with Diverse PR-Relationships

Our experiments indicate that CPXR can discover diverse predictor-response relationships and use them in building accurate PXR models, and other regression methods often miss such relationships (indicated by the low prediction accuracy of their models). Sometimes, the latter may have happened because the underlying types of regression models do not have the necessary mechanism to represent diverse relationships. Below we discuss another possible cause, which is related to diverse predictor-response relationships.

Recall that each distinct predictor-response relationship is applicable only to its associated logical group of data. Sometimes different predictor-response relationships may *cancel each other's effect* in the whole data set, in the sense that the diverse predictor-response relationships have no impact on the predictor-response relationship on all data. To illustrate, consider the soil water content[11] data set called Soil WC [3]. We first used a set S_1 of four variables to build a

[7] The TBI data was used to train a model for predicting traumatic brain injury patients' response (during the 6 months after surgery) to brain surgery, based on data at time of admission [22].

[8] Terms with small impact on y are omitted.

[9] $cause$:3 denotes $cause = 3$; age:[23, 43) denotes $23 \leq age < 43$.

[10] We use magnitude of coefficients of variables as indicator of importance of variables.

[11] This data set was used for water content prediction for soil samples. It has around ten variables, including soil texture, namely sand, silt, and clay content, geometric mean particle-size diameter and geometric standard deviation, as well as bulk density. Water content prediction for soil is important for agriculture and forest fire management, among other applications.

Authorized licensed use limited to: Dr. D. Y. Patil Educational Complex Akurdi. Downloaded on June 29,2025 at 11:50:30 UTC from IEEE Xplore. Restrictions apply.

linear regression model f. Then we added two variables to S_1 to get S_2, and built a linear regression model f' on S_2. It turns out that the additional variables offer no advantage on prediction accuracy—the accuracies of f' and f are roughly the same. Interestingly, CPXR found a PXR regression model on S_2 with big improvement (20 percent) in RMSE reduction relative to the RMSE of the PXR model on S_1. The two variables are involved in all but one of the patterns used by the PXR model; so they are heavily involved in diverse predictor-response relationships. Since the linear regression model on S_2 is not more accurate than that on S_1, the diverse predictor-response relationships discovered by CPXR somehow cancelled or neutralized each other's effect, when viewed from the perspective of the whole data set. Moreover, the competitor methods all failed to be competitive with CPXR (see Table 5)—their models on S_2 were about 40 percent less accurate than CPXR's model on S_2. We conjecture that the phenomenon described above also occurs in the data sets where CPXR's performance is over 25 percent better than the best competitor methods (see Table 5).

5 The CPXR Algorithm

This section presents the CPXR algorithm, together with two quality measures used by the algorithm and an illustrating example.

5.1 Quality Measures on Patterns and Pattern Sets

Our CPXR algorithm needs to efficiently find a desirable pattern set, from huge search spaces of potential pattern sets, in order to build an accurate prediction model. We need quality measures on individual patterns and on pattern sets, to remove patterns that have little hope of being useful and to perform efficient incremental search of pattern sets respectively. We next define such measures, based on residual reduction.

We first define our measure on patterns. Recall that f_P is the local regression model built from $\mathsf{mds}(P)$. Let $r_x(g)$ denote a function g's residual on an instance x.

Definition 3. *The* ***average residual reduction*** (arr) *of a pattern P w.r.t. a prediction model f and a data set D is*

$$\mathsf{arr}(P) = \frac{\Sigma_{x \in \mathsf{mds}(P)} |r_x(f)| - \Sigma_{x \in \mathsf{mds}(P)} |r_x(f_P)|}{|\mathsf{mds}(P)|}. \qquad (2)$$

Clearly, $\mathsf{arr}(P)$ measures the average residual reduction by f_P on instances in $\mathsf{mds}(P)$. It will be used to filter out patterns of little value and used as weights for patterns in CPXR prediction models.

We now define our quality measure on pattern sets. It will be used for pattern set selection.

Definition 4. *The* ***total residual reduction*** (trr) *of a pattern set $PS = \{P_1, \ldots, P_k\}$ w.r.t. a prediction model f and a data set D is*

$$\mathsf{trr}(PS) = \frac{\Sigma_{x \in \mathsf{mds}(PS)} |r_x(f)| - \Sigma_{x \in \mathsf{mds}(PS)} |r_x(f_{PM})|}{\Sigma_{x \in D} |r_x(f)|} \qquad (3)$$

where $PM = ((P_1, f_{P_1}, w_1), \ldots, (P_k, f_{P_k}, w_k), f)$, $w_i = \mathsf{arr}(P_i)$, and $\mathsf{mds}(PS) = \cup_{P \in PS}\, \mathsf{mds}(P)$.

So $\mathsf{trr}(PS)$ measures the total residual reduction achieved by f_{PM}, as a fraction of f's total residual.

5.2 The Algorithm

Before presenting the CPXR algorithm, we note that CPXR can be used to construct regression models, by first using a standard regression method to build a baseline regression model f. Moreover, CPXR can also be used to analyze prediction models (see Section 7).

Given $PS = \{P_1, \ldots, P_k\}$ and f_d, let $PM(PS, f_d)$ denote $((P_1, f_{P_1}, \mathsf{arr}(P_1)), \ldots, (P_k, f_{P_k}, \mathsf{arr}(P_k)), f_d)$. This will be used several times below.

Algorithm 1. The CPXR Algorithm

```
      Input: (1) training data $D = \{(x_i, y_i) | 1 \le i \le n\}$
      (2) a prediction model $f$
      (3) a number $\rho$ to partition $D$ into $LE, SE$
      (4) a $minSup$ threshold on contrast patterns
      Output: A PXR model
1.1  Let $r_1, \ldots, r_n$ denote $f$'s residuals on $x_1, \ldots, x_n$;
1.2  Determine $\kappa$ to minimize $|\rho - \frac{\sum_{r_i > \kappa} |r_i|}{\sum |r_i|}|$;
1.3  Let $LE = \{x_i \mid r_i > \kappa\}$, $SE = D - LE$;
1.4  Discretize each numerical variable using entropy based
       binning w.r.t. the $LE$ and $SE$ classes;
1.5  Extract all contrast patterns for $minSup$ in the $LE$ class,
       and select just one pattern having shortest length from
       each equivalence class for inclusion in the $CPS$ set of
       contrast patterns, and apply other filters (see Section 5.3)
       to remove patterns of little value from $CPS$;
1.6  For each $P \in CPS$, build the local regression model $f_P$ for
       data in $\mathsf{mds}(P)$;
1.7  Let $PS = \{P_0\}$, where $P_0$ is the pattern $P$ in $CPS$ with
       highest $arr$;
1.8  repeat
          // add a pattern to $PS$
1.9     Let $P$ be the pattern in $CPS - PS$ that
          maximizes $\Delta(f_{PM(PS \cup \{P\}, f)}, f)$;
          // $\Delta(f', f)$ denotes $\frac{RMSE(f) - RMSE(f')}{RMSE(f)}$
1.10    Let $PS^o = PS$;
        if
1.11    $\Delta(f_{PM(PS \cup \{P\}, f)}, f) > 0$ then
1.12       Let $PS = PS^o \cup \{P\}$;
1.13       Call IteratImp(CPS,PS) to improve $PS$;
        end
      until $\Delta(f_{PM(PS,f)}, f) - \Delta(f_{PM(PS^o,f)}, f) < 0.01$;
1.14  Let $f_d$ be the regression model trained from
       $D - \cup_{P \in PS} \mathsf{mds}(P)$;
1.15  Return $PM(PS, f_d)$;
```

CPXR takes four inputs: a training data set D for regression, a prediction model[12] f, a ratio ρ for dividing D into large error and small error parts, and a $minSup$ threshold on contrast patterns.

By default, CPXR uses linear regression to build baseline/local regression models. Other methods can be used.

We first give a high level view of CPXR. After dividing D into LE and SE parts, CPXR mines the contrast patterns of LE and uses filters to remove patterns that are similar to

[12] One can also use f's residuals on data in D instead of f itself.

Authorized licensed use limited to: Dr. D. Y. Patil Educational Complex Akurdi. Downloaded on June 29,2025 at 11:50:30 UTC from IEEE Xplore. Restrictions apply.

others or that yield little residual reduction. Then it uses a double loop to search for a desirable pattern set with large trr: The inner loop performs repeated pattern replacements, and the outer loop adds a new pattern to the pattern set and then calls the inner loop. The inner loop terminates when the improvement of the best replacement is too small. The outer loop terminates when the improvement of the last iteration is too small.

Specifically, after computing f's residuals, CPXR uses ρ to determine a split value κ (Step 1.2), to partition D into LE and SE parts, based on the magnitude of the residuals (Step 1.3); κ is found so that $\rho \approx \frac{\sum_{x_i \in LE} |r_i|}{\sum_{x_i \in D} |r_i|}$. Step 1.4 uses the entropy based binning method to discretize all numerical variables into disjoint intervals. In Step 1.5, the contrast patterns of the LE class (contrasting the LE class against the SE class) are mined (using GcGrowth of [27], other methods can also be used); then one shortest pattern in each equivalence class is selected and added to CPS. In Step 1.6, a regression model f_P is built for $\mathsf{mds}(P)$ for each $P \in CPS$ (using the corresponding y values of instances $x \in \mathsf{mds}(P)$). In Step 1.7, the pattern P_0 in CPS with highest arr is selected to initialize PS. Steps 1.9-1.13 repeatedly add one pattern to PS (if the previous addition to PS led to more than 1 percent improvement in RMSE reduction), and iteratively improve the PS set. Finally, the algorithm determines f_d on the data not matching any $P \in PS$ and returns $PM(PS, f_d)$.

We now discuss $IteratImp(CPS, PS)$, which uses an iterative loop to find improving replacement of patterns in PS. The improvement value of replacing P_- in PS by $P_+ \in CPS - PS$ is measured by

$$imp(PS, P_-, P_+) = \mathsf{trr}(f_{PM'}, f) - \mathsf{trr}(f_{PM(PS,f)}, f),$$

where $PM' = PM(PS - \{P_-\} \cup \{P_+\}, f)$. For each P in PS, it finds P's best replacement Q_P in $CPS - PS$. Then, it finds the best pair, $P \in PS$ and $Q_P \in CPS - PS$, of replacement that maximizes the improvement value, and actually makes the replacement if the improvement is positive. This top-level loop terminates when the improvement is too small.

Algorithm 2. The $IteratImp(CPS, PS)$ Function

```
2.1 Let $impval = 1$;
2.2 repeat
2.3   for each $P \in PS$ do
2.4     Let $Q_P$ be a pattern $P'$ in $CPS - PS$ maximizing
          $imp(PS, P, P')$;
      end
2.5     Let $P_-$ be a pattern $P \in PS$ maximizing $imp(PS, P, Q_P)$;
2.6     Let $impval = imp(PS, P_-, Q_{P_-})$;
2.7     if $impval > 0$ then $PS = PS - \{P_-\} \cup \{Q_{P_-}\}$;
    until $impval < 0.001$;
```

5.3 Techniques for Efficient Computation

To speed up the computation we use several techniques, aimed at reducing the number of patterns without losing highly desirable pattern sets. We start with contrast patterns of LE whose support ratio is at least 1. (This allows us to focus on far fewer patterns than the case when one starts with frequent patterns.) Then, we remove those patterns P associated with small reduction of the residual of f, i.e., $\Sigma_{x \in \mathsf{mds}(P)} |r_x(f)| - \Sigma_{x \in \mathsf{mds}(P)} |r_x(f_P)| \leq 0.01 \Sigma_{x \in \mathsf{mds}(P)} |r_x(f)|$. As an optional step, we then remove patterns by using Jaccard similarity between pattern pairs P_1 and P_2, defined by $\frac{|\mathsf{mds}(P1) \cap \mathsf{mds}(P2)|}{|\mathsf{mds}(P1) \cup \mathsf{mds}(P2)|}$; for each pair whose the Jaccard similarity is ≥ 0.9, the pattern with smaller arr is removed. Finally, we remove patterns whose matching data set's cardinality is less than the number of predictor variables in the data set.[13]

TABLE 3
D_r: Four Predictor Variables and 38 Instances

x	u	v	w	y
160	85	174	26	350
182	91	295	28	440
208	85	278	25	391
343	85	349	31	432
152	91	164	33	403
206	70	338	30	428
213	88	216	30	412
162	80	185	28	340
146	91	356	30	511
164	88	272	25	379
106	96	356	23	432
188	80	122	30	367
49	93	324	32	559
105	91	284	28	490
140	98	335	23	486
207	78	362	30	408
238	85	353	26	380
190	93	362	30	527
228	85	101	24	238
98	91	407	30	577
261	85	245	27	318
277	80	271	28	321
177	85	186	23	280
195	88	353	32	428
232	85	335	27	480
107	88	121	31	445
156	91	160	23	268
13	88	308	26	399
123	80	308	26	339
186	85	118	23	381
249	85	305	25	472
159	91	22	24	368
166	85	144	35	298
148	98	236	27	309
177	93	175	27	257
100	93	288	34	631
125	96	278	29	624
55	93	291	31	309

We also use an incremental computation idea to speed up the computation of the improvement values concerning adding a pattern P into, or replacing a pattern P_- by a pattern P_+ in, a pattern set PS. This is done by using the matching data sets (implemented as bit-vectors) of patterns of interest to identify data instances whose $f_{PM(PS,f)}$ value will change. This helps avoid recomputing the entire $f_{PM(PS,f)}$ function for each candidate pattern (pair).

The ideas above have been proven to be effective by large number of experiments, enabling us to get high quality PXR models efficiently for data sets with large numbers of variables, instances, and patterns.

5.4 Example Illustrating CPXR

To illustrate CPXR, we consider data set D_r in Table 3.

First, we build this baseline linear regression model

$$f : y = -74.07 - 0.29\, x + 2.32\, u + 0.5\, v + 7.19\, w$$

from D_r, whose RMSE is 70.81 and its R^2 is 0.45. Next, we divide the data into LE and SE classes, using Steps 1.2 and 1.3 of Algorithm 1 with $\rho = 0.45$. The shaded 8 rows are in LE and the other 30 rows are in SE.

Now, a total of 17 contrast patterns of LE were extracted; 12 were removed in the filtering process, leaving just five PIPs (positive impact patterns), given in Table 4. Here, Cov denotes the support of patterns in the whole data set. We rounded some trr to integer percentages.

The pattern selection phase starts by selecting the pattern with the highest arr, which is P_4 here, setting $PS = \{P_4\}$.

[13] This helps control overfitting.

Authorized licensed use limited to: Dr. D. Y. Patil Educational Complex Akurdi. Downloaded on June 29,2025 at 11:50:30 UTC from IEEE Xplore. Restrictions apply.

TABLE 4
Pattern Characteristics (trr, Cov in Percentage)

ID	Pattern	arr	trr	R^2	Cov
P_1	x:$[95.5, 178)$	2,507	25	0.72	50
P_2	u:$[91, \inf)$	382	3.2	0.45	42
P_3	x:$[95.5, 178)$, u:$[91, \inf)$	2,988	19	0.71	31
P_4	v:$[214.5, 310.75)$, u:$[91, \inf)$	5,447	17	0.42	16
P_5	v:$[118.25, 214.5)$, x:$[95.5, 178)$	4,492	21	0.94	23

Then we pair P_4 with each of the other four patterns and select a pattern P_i with largest $\Delta(f_{PM(\{P_4\}\cup\{P_i\},f)}, f)$. Here, P_3 is selected as $\{P_3, P_4\}$ has the highest RMSE reduction, and PS becomes $\{P_3, P_4\}$. Next, we call *IteratImp* to find replacements for patterns in PS that can lead to more RMSE reduction. Here, the only "improving" change to PS is the replacement of P_3 by P_1, yielding $PS = \{P_1, P_4\}$. Since $\Delta(f_{PM(\{P_1,P_4\},f)}, f) - \Delta(f_{PM(\{P_3,P_4\},f)}, f) < 0.01$, the algorithm terminates. (The above also demonstrates that not all pattern sets yield PXR models with optimal accuracy.) The algorithm returns $PM = \{(P_1, f_{P_1}, \mathsf{arr}(P_1)), (P_4, f_{P_4}, \mathsf{arr}(P_4)), f_d\}$, where

$$f_{P_1} : y = -3944 + 1.51\,x + 23.58\,u + 4.3\,v + 27.82\,w$$
$$f_{P_4} : y = 401.7 - 2.47\,x + 1.18\,u + 0.33\,v + 6.43\,w$$
$$f_d : y = -168 - 0.21\,x + 3.78\,u + 0.42\,v + 6.48\,w.$$

The RMSE and R^2 of the final PXR model PM are 51.7 and 0.8 respectively, yielding 28 and 77 percent improvement respectively over those of the linear regression model f.

We note that several patterns (e.g., P_5, P_3 and P_2) with higher R^2 than P_4 were not selected. For example, although $\mathsf{arr}(f_{P_3})$ is higher than $\mathsf{arr}(f_{P_1})$, $\mathsf{trr}(\{P_3, P_4\})$ is 11 percent less than $\mathsf{trr}(\{P_1, P_4\})$; one reason for this is that P_1 and P_3 are not complementing each other nicely, since $\mathsf{mds}(P_1)$ and $\mathsf{mds}(P_3)$ are very similar. P_1 and P_4 are very dissimilar to each other in their mds and they complement each other well. This also demonstrates when a pattern set may yield a desirable PXR model.

To make prediction on an unseen instance, we need to follow Eq. (1) using arr of P_1 and P_4 as their weights.

6 Experimental Evaluation

This section reports a systematic evaluation of CPXR. We compare CPXR against several state-of-the-art regression methods, on 50 real data sets, concerning prediction accuracy, overfitting, and sensitivity to noise. The results show that CPXR is consistently better than competing methods, often by big-margins. We discuss the reasons why, and the conditions when, CPXR outperforms other methods by big margins. We also examine the impact of parameters and the baseline regression methods on CPXR's performance, and CPXR's computation time and memory usage. Finally, we discuss advantages of using contrast patterns instead of frequent patterns in constructing PXR models.

CPXR can choose regression methods for building baseline and local regression models. Below, unless noted otherwise, by CPXR we mean CPXR using linear regression for generating both the baseline and local regression models. There are two exceptions (Sections 8.2 and 8.5.3). Sometimes, we will use CPXR(LL) and CPXR(LP) to respectively denote the variants using LR and PLR for local regression models (both using LR for generating baseline regression models).

Remark. Since RMSE measures error, smaller RMSE is better, and larger RMSE reduction is better.

6.1 Data Sets, Regression Methods and Parameters

The 50 real data sets were obtained from several sources (43 from [23], the other seven from the UCI repository [8] and elsewhere [36], [34], [2], [22]). Table 5 shows some statistics of the data sets, including their names, sources, and numbers of predictor variables and of instances; it also shows how each data set is split into training and testing parts in the experiments. For some data sets from [23], we used the splits given by [10] (six-fold CV with 20 repetitions), since BART is the best competitor and we want to find out how CPXR compare against BART.

The 50 data sets are very diverse in terms of dimensionality (between 4 and 28) and in size (between 96 and 21,252 instances). We excluded simple data sets having ≤ 3 predictor variables.

The state-of-the-art regression methods used in our comparison are: linear regression, piecewise linear regression [28], support vector regression [13], Bayesian additive regression trees [10], and gradient boosting [19] (which generalizes AdaBoost). We did not include Neural Network and Random Forest here, since experiments [10] show that they are usually inferior to BART.

We now discuss the implementations and parameter settings of the regression methods. For LR, we used our implementation; no parameters are needed. For PLR, we used the implementation in R [31] (with default value for "number of breakpoints" determined by an function called seg.control). (Better implementations of PLR can be used by our CPXR to get better results. Our experiments here have demonstrated that CPXR can do better, using this PLR implementation.) For SVR, we used svm with RBF kernels in R (in the e1071 package [14]) with default parameter settings. For GBM [19], we used R's implementation with default settings. For BART, we used BayesTree in R, with the same set of parameters proposed in [10] (following the BART-cv methodology[14]). For CPXR we used fixed parameter values, $minSup = 0.02$ and $\rho = 0.45$, except when examining the impact of parameters.

6.2 Prediction Accuracy Evaluation

Table 5 shows the reduction of RMSE (relative to LR's RMSE) for PLR, SVR, BART, GBM, CPXR(LL), CPXR(LP), and CPXR(LL:Regularized)[15] on the 50 data sets; a cell's value is defined as $(RMSE(LR) - RMSE(X))/RMSE(LR))$ where X is the method of the cell, and for each method Y, $RMSE(Y)$ is the RMSE of Y on the test data; for CPXR(LL: Regularized), $RMSE(LR)$ is replaced by the RMSE of regularized linear regression. Each bold number indicates the best RMSE reduction value among all methods except CPXR(LL:Regularized) (since the other algorithms are not regularized).

[14] BART-cv uses three sets of parameter values (selected by the authors of [10]); the set giving the best result (on the training data) among the three sets is used.

[15] We set $\lambda = 0.05$ when using regularized linear regression.

Authorized licensed use limited to: Dr. D. Y. Patil Educational Complex Akurdi. Downloaded on June 29,2025 at 11:50:30 UTC from IEEE Xplore. Restrictions apply.

TABLE 5
RMSE Reduction of PLR, SVR, BART, GBM and CPXR over RMSE of LR

Dataset	#instances	#variables	PLR	SVR	BART	GBM	CPXR(LL)	CPXR(LP)	CPXR(LL:Regularized)
Abalone [23] a	4,177	8	4.5	0.00	3.18	1.36	12.33	**14.25**	18.67
Alcohol [23] a	2,467	18	12.58	10.6	20.53	11.26	24.83	**26.14**	21.71
Amenity [23] a	3,044	21	34.89	29.24	41.34	39.11	39.68	**42.19**	5.85
Attend [23] a	838	9	11.24	2.4	28.07	19.58	16.54	**30.43**	39.08
Baskball [23] a	96	4	21.93	11.23	8.02	4.81	53.66	**54.1**	40.91
Budget [23] a	1,729	10	37.48	26.81	**91.58**	84.46	76.30	80.58	76.15
Cane [23] a	3,775	9	8.51	0.00	20.15	16.42	23.93	**25.97**	21.48
Cardio [23] a	375	9	−18.42	−0.71	−0.21	−15.63	43.97	**49.09**	45.47
College [23] a	694	24	14.9	2.65	11.33	4.78	43.03	**46.73**	49.27
Concrete [36] c	1,030	9	43.76	19.41	27.47	−48.18	41.36	**48.17**	50.48
County [23] a	3,114	13	11.94	3.67	26.29	23.42	**32.33**	29.18	31.42
CPS [23] a	534	10	−3.75	−10.00	−5.00	−5.00	4.65	**4.92**	27.82
CPS95 [23] b	21,252	14	10.81	16.5	55.29	21.46	**65.57**	59.69	55.74
CPU [23] a	209	7	33.31	10.54	41.51	−59.65	57.78	**58.09**	63.04
Deer [23] a	654	13	−29.14	−28.95	−21.00	−28.34	50.07	**52.04**	47.00
Diabetes [23] a	375	15	7.43	6.43	4.42	4.71	32.32	**34.11**	46.85
Edu [23] a	1,400	5	10.78	5.75	10.66	10.18	11.13	**13.45**	18.08
Energy Eff. [34] c	768	8	64.65	18.52	42.42	**80.07**	59.47	61.14	58.68
Engel [23] b	11,986	5	6.72	2.99	5.37	5.22	11.2	**13.93**	18.79
Enroll [23] a	258	6	10.6	−6.79	0.66	−7.62	21.6	**22.73**	41.94
Fame [23] a	1,319	22	43.95	11.58	41.91	24.36	55.11	**58.1**	43.43
Fat [23] a	252	14	16.04	4.4	2.86	−0.88	**45.88**	44.89	49.65
Fishery [23] a	6,806	14	1.34	15.53	**35.73**	33.05	13.51	19.09	34.58
Houses [23] b	6,880	8	6.93	1.35	25.13	24.34	23.49	**39.48**	23.61
Insur [23] a	2,182	6	29.82	−0.26	33.68	**83.65**	40.49	39.07	39.95
Istanbul SE [2] c	536	8	23.73	5.08	22.03	16.95	22.92	**25.01**	50.00
Labor [23] a	2,953	18	67.87	43.71	**74.13**	70.49	30.56	32.88	56.08
Labor2 [23] b	5,443	17	2.86	−3.17	0.00	1.59	13.11	**14.28**	10.82
Laheart [23] a	200	16	22.08	−0.87	−0.55	3.61	52.44	**52.76**	60.34
Medicare [23] a	4,406	21	12.48	7.75	21.71	12.4	21.77	**28.79**	17.98
MPG2001 [23] a	852	10	15.02	−4.1	**37.2**	29.01	34.82	36.25	51.52
Mussels [23] a	201	4	44.83	4.39	28.68	25.39	59.15	**61.3**	60.62
Ozone [23] a	330	8	13.02	1.77	9.49	9.49	**42.68**	41.01	43.95
Pole [23] b	5,000	26	25.81	11.37	34.02	**64.47**	34.33	43.11	16.83
Price [23] a	159	15	43.72	4.16	29.54	17.73	70.65	**71.08**	64.26
Rate [23] a	144	9	−5	−37.5	−12.5	−37.5	18.18	**19.44**	19.71
Rice [23] a	171	15	33.72	−1.67	14.84	17.16	59.87	**60.05**	30.91
Rosetta [3] c	213	13	50.72	51.06	51.06	55.84	83.25	**87.14**	78.73
Servo [23] a	167	4	20.81	−33.33	**56.57**	51.52	28.18	31.1	14.58
Smsa [23] a	141	10	26.03	6.03	−33.29	−51.97	84.34	**85.9**	31.92
Soil WC [3] c	210	10	3.26	2.17	8.7	18.48	47.87	**48.11**	24.76
Spouse [23] b	11,136	21	12.9	11.83	36.56	15.59	45.27	**52.11**	34.26
Strike [23] a	625	5	−24.13	−1.18	−0.77	−0.35	30.18	**47.93**	13.07
TA [23] a	324	6	8.66	0.00	−1.22	−1.22	**36.46**	33.04	31.25
TBI [22] c d	2,159	16	35.51	13.71	33.14	14.95	67.18	**69.41**	68.22
Tecator [23] a	215	10	40.62	0.16	19.35	−14.15	63.02	**65.1**	67.64
Tree [23] a	100	8	17.68	7.92	−7.23	−10.82	59.22	**61.73**	38.99
Triazine [23] a	186	28	25.24	1.51	13.44	12.89	23.49	**25.98**	33.19
Wage [23] a	3,380	13	12.2	9.15	25.42	11.86	21.31	**38.45**	28.21
Yacht [8] c	308	7	−2.19	−5.93	−2.68	**69.65**	43.81	45.1	51.97
Average	–	–	18.41	4.94	20.18	14.6	39.89	**42.89**	39.39

[a]Training and test splits as provided by BART's authors. [b]External test data set as used by [10]. [c]Training and test splits according to 10 fold CV. [d]We thank E. W. Steyerberg for discussion on the TBI data and the regression model he previously used on the data.

CPXR(LP) achieved the highest average RMSE reduction (42.89 percent), which is approximately 24.5, 38, 22.7 and 28.29 percent higher than that of PLR, SVR, BART and GBM respectively. Moreover, CPXR(LP) achieved the highest RMSE reduction in 36 out of 50; in comparison, PLR, SVR, BART, GBM, and CPXR(LL) are the best 0, 0, 5, 4, 5 times respectively. Among PLR, SVR, BART, GBM and CPXR (LP), CPXR(LP) is the best in 41 out of the 50 data sets. CPXR(LP) achieved RMSE reduction over 80 percent several times. CPXR(LL:Regularized)'s performance is quite similar to that of CPXR(LL), although it is a bit less accurate (CPXR (LL:Regularized)'s RMSE reduction over standard LR is 38.9 percent, versus 39.89 percent for CPXR(LL)).

CPXR(LL) achieved the average RMSE reduction of 38.89 percent, which is about 2.5 percent lower than that of CPXR (LP). In general, CPXR(LL)'s performance is very close to that of CPXR(LP). Both CPXR(LP) and CPXR(LL) achieved positive RMSE reduction relative to LR in all 50 data sets. Sometimes users might prefer to use CPXR(LL) since the local regression models are somehow simpler than those

Authorized licensed use limited to: Dr. D. Y. Patil Educational Complex Akurdi. Downloaded on June 29,2025 at 11:50:30 UTC from IEEE Xplore. Restrictions apply.

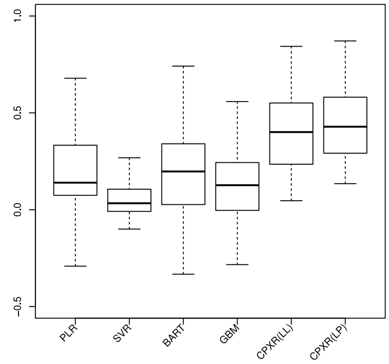

Fig. 2. Box plot of RMSE reductions.

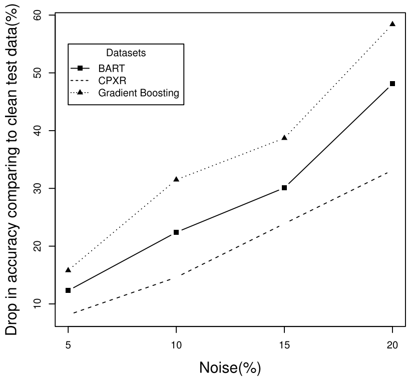

Fig. 3. Noise sensitivity of regression methods.

used by CPXR(LP). If the focus is accuracy, then CPXR(LP) should be used.

It is worth noting that, when CPXR(LP)'s RMSE reduction is the highest, it is usually much larger than that of other methods, and when it is not the highest, it is usually not much smaller than that of other methods (Labor and Servo are two exceptions). Finally, out of the 50 data sets, there are 42 where CPXR(LP)'s performance is over 25 percent (relatively) compared to LR.

Fig. 2 shows the box plots for the RMSE reductions of six methods in Table 5. Clearly, the quartiles of CPXR(LL) and CPXR(LP) are consistently higher than those of the other methods, and the first quartiles of CPXR(LL) and CPXR(LP) are higher than the median of the other four methods.

Remark. Since BART models are much more complicated than those of PLR and the PLR models have the constructs needed to model some diverse predictor-response relationships, the fact that PLR achieved almost the same accuracy as BART confirms that diverse predictor-response relationships occur often in real data. However, the limited form of PLR prevented it from being competitive with CPXR in general.

To compare CPXR and M5P (model trees) [35], we report their performance on the three shared datasets studied in [35] and in our study: CPXR's models are more accurate than M5P models in terms of RMSE by 64.8, 61 and 36.5 percent on CPU, Price (called auto-price in the M5P paper) and Servo respectively. We used RWeka's implementation of M5P.

6.3 Overfitting and Noise Sensitivity Comparison

Comparison with respect to overfitting. Overfitting is an important issue on the desirability of prediction models. In general, the goal is to build accurate models while at the same time avoiding overfitting, so that the models generalize to new data instances. This section compares CPXR and other algorithms regarding overfitting.

TABLE 6
Average Relative Drop in Accuracy

Method	Average of RMSE reduction over LR		Drop in accuracy
	Training	Test	
PLR	37.11%	18.76%	49%
SVR	7.65%	4.8%	37%
BART	41.02%	20.15%	51%
CPXR(LL)	51.4%	39.88%	22%
CPXR(LP)	53.85%	42.89%	21%

In general, a model is overfitting "if it is more complex than another model that fits equally well" [21]. Because CPXR's models are more accurate than the regression models produced by other algorithms, CPXR is the winner on this perspective. Since CPXR uses simple regression models (linear or piecewise linear) for the baseline and local regression models and the CPXR models use a very small number of patterns (around seven patterns in the experiments on real data sets), CPXR's models are quite simple and easy to understand.

We also use "relative accuracy drop," the difference between the accuracies on training data and test data, to evaluate overfittingness. While the accuracy on training data is higher than that on test data for all algorithms we considered, the relative magnitude of accuracy drop differs. The last column of Table 6 shows the relative accuracy drop, computed by $\frac{RMSE(training)-RMSE(test)}{RMSE(training)}$ for each algorithm. On average, CPXR(LP)'s accuracy drop is the smallest for the 50 data sets, and CPXR(LL)'s accuracy drop is similar to that of CPXR(LP). So CPXR is the winner on this perspective.

Recall that CPXR does not select patterns whose matching data set's cardinality is less than the number of predictor variables in the data set. This idea should have helped on controlling overfitting.

Remark. Residual reductions by CPXR models on training data and testing data are distributed in fairly similar manners. For example, for Tecator, we divided all test instances into four quarters after sorting the instances in decreasing residual (of the LR model) order, and similarly on the training instances. Residual reduction by CPXR on the testing and training data are distributed to the four quarters in similar manner: 6, 18, 30, 52 percent (testing) vs 6, 14, 30, 57 percent (training).

Comparison on noise sensitivity. Sensitivity to noise is an important issue on the quality of prediction algorithms, and on the overfittingness of models they produce. To evaluate noise sensitivity of regression algorithms, we examine the difference of their models' accuracy on clean training data and noise-added test data, on three real data sets. (With noise level ℓ, we transform the test data by adding a value $z \times y$ to each y value in the original test data, where z is a random value in $[-\ell, \ell]$.) We used 0.05, 0.10, 0.15 and 0.20 as noise levels ℓ.

Fig. 3 shows the change in RMSE for BART, CPXR and GBM models. Apparently GBM is the most sensitive to noise and CPXR is the least. For example, when we added

Authorized licensed use limited to: Dr. D. Y. Patil Educational Complex Akurdi. Downloaded on June 29,2025 at 11:50:30 UTC from IEEE Xplore. Restrictions apply.

TABLE 7
Characteristics of Data Sets Where CPXR has Different Performance

		Dataset	# of patterns	# of PIP	Cov on LE	Cov on all data	Avg R^2 improvement	Difference in coefficients
LR	High	CPS95	2443	1720	91%	89%	14%	2.1
		Smsa	40	39	87%	85%	24%	2.6
		Price	351	227	95%	79%	11%	2.7
		CPU	138	93	95%	92%	17%	3.2
		Tree	33	16	50%	63%	16%	2.1
	Low	Fat	1135	1086	29%	30%	14%	1.1
		Wage	2969	208	34%	57%	4.5%	1.4
		Attend	1402	63	29%	42%	14%	1.7
		Strike	54	48	38%	17%	59%	1.9

5 percent noise, the RMSE increased by 8, 12 and 16 percent for CPXR, BART and GBM models respectively.

6.4 Data Characteristics vs CPXR's Performance

This section analyzes two groups of data sets in order to understand what data characteristics are correlated with CPXR's performance. We selected nine real data sets, divided into two groups: the "high" group consists of five data sets where CPXR has very large RMSE reduction over LR (between 57 and 84 percent for CPXR), and the "low" group consists of four where CPXR has relatively small RMSE reduction (between 16 and 45 percent for CPXR). Table 7 gives the characteristics of these data sets, including the number of (equivalence classes) of contrast patterns after filtering using minSup, supportRatio and total residual reduction, the number of positive improvement patterns (PIPs) (after removing, from the set of patterns above, any pattern P whose associated R^2 improvement over the baseline regression model, on $\mathsf{mds}(P)$, is less than 5 percent), (collective) coverage on large error instances (by PIPs), (collective) coverage on all instances (by PIPs), average R^2 improvement of local regression models of the PIPs, and the difference (given as a ratio) between largest coefficients of the local regression models in the result computed by CPXR. The R^2 improvement by a pattern P is defined as $\frac{R^2(f|_{\mathsf{mds}(P)})-R^2(f_P)}{R^2(f|_{\mathsf{mds}(P)})}$, where $f|_{\mathsf{mds}(P)}$ denotes f restricted to $\mathsf{mds}(P)$.

We want to understand what characteristics are indicators of different performance of CPXR. From Table 7 we can see that the number of PIPs, the coverage on large error instances, and the difference in largest coefficients are the dominant factors. CPXR is a pattern based algorithm and it uses patterns (and associated local regression models) to characterize diverse predictor-response relationships. If the number of PIPs is small or the coverage on large error instances is small, then the data set does not involve significant diverse predictor-response relationships; CPXR will not give large improvement in such cases. In general, we see that data sets in the "high" group all have Cov LE $\geq 50\%$, Cov all$\geq 63\%$, and Difference in coefficients ≥ 2.1, whereas data sets in the "low" group all have Cov LE $\leq 38\%$, Cov all$\leq 57\%$, and Difference in coefficients ≤ 1.9. When data sets have high Cov LE, most of the large error instances are covered by PIPs; when Difference in coefficients is large, the baseline regression model tends to make very large prediction errors on many instances; both offer opportunity for CPXR to make large improvement in prediction accuracy.

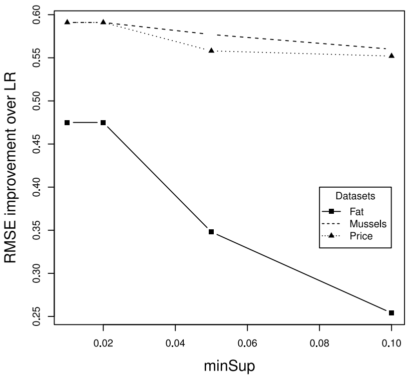

Fig. 4. minSup's impact on RMSE reduction.

Remark. . Table 7 confirms that real life data sets indeed contain diverse predictor-response relationships.

6.5 Analysis of CPXR

Impact of minSup. One of the two CPXR parameters is $minSup$, the minimum threshold on support (in LE) of contrast patterns. Fig. 4 shows how RMSE reduction increases

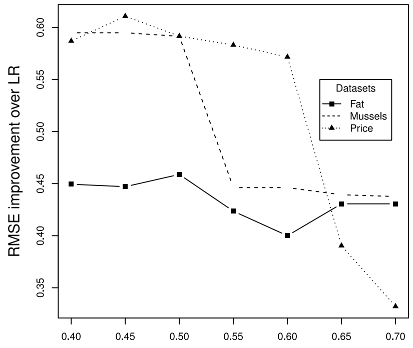

Fig. 5. ρ's impact on RMSE reduction.

TABLE 8
RMSE of CPXR vs Baseline Methods

Dataset	LR	BART	GBM	PLR
Enroll	0.09	0.06	0.1	0.09
Diabetes	0.72	0.63	1.15	0.46
CPU	10.24	10.09	15.34	11.1
Insur	2.16	2.17	2.27	2.28

Authorized licensed use limited to: Dr. D. Y. Patil Educational Complex Akurdi. Downloaded on June 29,2025 at 11:50:30 UTC from IEEE Xplore. Restrictions apply.

TABLE 9
Running Time and Memory Usage of CPXR, and Running Time of Other Algorithms

Dataset	Number of instances	Number of attributes	Number of PIPs	CPXR time^a	CPXR memory (MB)	LR time^b	PLR time^b	SVR time^b	BART time^b	GBM time^b
Fat	252	14	1,135	0.1	1.1	0.25	0.22	0.058	9.78	1.06
Mussels	201	4	38	0.015	0.85	0.29	0.15	0.036	7.56	0.68
Price	159	15	362	0.022	0.8	0.27	0.25	0.042	6.89	1.09
Spouse	11,136	21	228,165	6.1	48	0.8	0.75	39.26	1839	9.19
Pole	5,000	26	4,442,542	8.1	83	0.54	0.79	16.12	14.5	5.57

a: minues; b: seconds.

when $minSup$ drops from 0.1 to 0.01 (with ρ fixed at 0.45). Although $minSup = 0.01$ gives slightly better result than $minSup = 0.02$, having $minSup = 0.01$ can lead to significant increase in computing time and memory usage. We decided to set $minSup$ at 0.02 in our experiments (we built so many prediction models—the number of data sets times the number of folds).

Impact of ρ. The other parameter of CPXR is ρ (controlling ratio between cumulative residual of LE and of SE). Fig. 5 shows how RMSE reduction changes when ρ varies between 0.4 and 0.7 (with $minSup$ fixed at 0.02). Clearly, all ρ values between 0.45 and 0.65 are near optimal. We recommend to try multiple ρ values. Our experiments used $\rho = 0.45$ for all data sets.

Impact by baseline regression methods. In all other experiments we used linear regression to generate baseline regression models. We want to know how changing the methods for building the baseline models will affect the performance of CPXR (we kept LR here for generating local regression models). Table 8 shows the RMSE of CPXR when different regression methods were used to generate the baseline regression models. It turns out that the results of using BART to generate the baseline models are only slightly better than that of using LR, the results of using PLR to generate the baseline models are almost the same as that of using LR, and the results of using GBM are actually a bit worse than that of using LR. We conjecture that a possible reason is that the data where GBM makes large errors are overly fragmented, and as a result the contrast patterns of LE are not as useful as those for the LR and BART cases. While using BART may also fragment the large error data, the better performance of the BART models (as baseline models) may have compensated the performance of the computed PXR model.

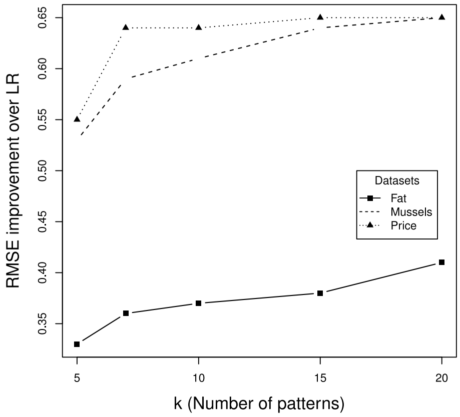

Fig. 6. Impact of number of patterns.

Impact of number of patterns in PXR models. The CPXR algorithm decides how many patterns are selected for use a PXR model. Fig. 6 shows the influence of the number k of patterns, when k is predetermined. We see that the curves are near their peak before k is 10. In our experiments on the 50 data sets, the maximum, minimum and average of k are 12, 3 and 7 respectively.

Impact of pattern filtering. Experiments on several datasets indicated that the pattern filtering part of CPXR helped shorten the computation time fairly significantly and no significant loss of prediction accuracy of the computed PXR models was observed. The speedup in computation is achieved because pattern filtering often removes more than 20 percent (for Spouse the filtering removed $128{,}240$ from a total of $520{,}000$) of the equivalence classes of contrast patterns, which help reduce the time for building local models and for searching for PXR models (typically reducing the total computation time by at least 20 percent).

Using order-based PXR and squared error. We performed experiments to determine if using order-based definition[16] for PXR, or using squared error instead of absolute error in arr and trr, can substantially affect prediction accuracy. The answer is basically no, although the first option led to slight loss of accuracy.

6.6 Computation Time and Memory Usage

We tested CPXR on a single processor machine (with a 2.26 GHz CPU, 4 GB of RAM), concerning running time and memory usage. Table 9 shows the results. Three factors are important for running time, namely the numbers of instances, of variables, and of positive improvement patterns (see Section 6.4). Clearly, CPXR built the PXR models within reasonable amount of time and memory. For 4 of the 6 data sets, the computing time was at most 0.3 minutes. For Pole the computing time is about 8 minutes; this could be attributed to the numbers of PIPs and of variables which are both fairly large.

6.7 Advantages of Using Contrast Patterns in CPXR

Experiments show that CPXR (using contrast patterns) can find more accurate PXR models faster than using frequent patterns. For illustration, consider the Spouse dataset: For minSup = 2% and minSupRatio = 1, there are 391,760 equivalence classes of contrast patterns (after applicable

[16] Among the patterns matching a given instance x, the local regression model of the pattern with the highest arr determines the predicted value for x.

Authorized licensed use limited to: Dr. D. Y. Patil Educational Complex Akurdi. Downloaded on June 29,2025 at 11:50:30 UTC from IEEE Xplore. Restrictions apply.

filtering) and CPXR used about 10 minutes to get a PXR model that achieved a 53.4 percent RMSE reduction. In contrast, for minSup $= 2\%$, there are around 1.7 M (million) equivalence classes of frequent patterns (after applicable filtering) using equi-width binning (of each numerical variable into three bins); CPXR's search process used about 101 minutes to get a PXR model that achieved a 45.9 percent RMSE reduction. For datasets with more variables, and hence many more frequent patterns, the computation time using frequent patterns will become much longer. (For example, in a dataset that we worked with (from a private source), there are 84 predictor variables and 1,278 instances; for minSup $= 1\%$ and minSupRatio $= 1$ there are 1.2 M ECs of contrast patterns, and for minSup $= 1\%$ there are 22.6M ECs of frequent patterns (using entropy based binning).)

7 Conclusion and Discussion

This paper articulated the diverse predictor-response relationship phenomenon, which says that in regression applications the predictor-response variable relationships fitting different logical groups of data are often highly different. The paper introduced a novel type of regression models, called pattern aided regression models, defined using small sets of patterns and corresponding local regression models. PXR models can naturally model diverse predictor-response relationships. The paper introduced a regression method (CPXR) for building highly accurate PXR models, which outperforms state-of-the-art regression methods, often by big margins, in experiments. PXR models are easy to interpret, and they achieve highly accurate prediction modeling with low model complexity; the above indicate that the pattern and local regression model construct of PXR models is very powerful, enabling PXR models "to achieve more with less". PXR/CPXR is especially effective for high-dimensional applications involving multiple multi-variable interactions. The paper also discussed how to use the PXR/CPXR methodology (a) to analyze individual prediction models and to correct their prediction errors, and (b) to compare multiple prediction models.

Finally, we discuss how to use CPXR to analyze prediction models. Given a prediction model f that one wants to analyze together with a training data set D, we can analyze the pairs $(P_1, f_{P_1}), \ldots, (P_k, f_{P_k})$ of patterns and local regression models found by CPXR to analyze f. Indeed, each P_i characterizes a subset of data where f makes large prediction errors and f_{P_i} can correct those errors. The weights associated with P_i indicates how much gain in accuracy f_{P_i} can achieve. Moreover, each (P_i, f_{P_i}) pair shows an interaction among some set of variables; that set consists of the variables used in P_i and those associated with large coefficients in f_{P_i}. Combining the constants used in the patterns and the variable coefficients in local regression models, these sets can indicate how different predictor-response relationships differ, and how they differ from the baseline regression model. The CPXR methodology can also be used to identify key differences between two prediction models.

Acknowledgments

This work was supported in part by the National Science Foundation under grant IIS-1044634. The authors also thank the anonymous reviewers for their very constructive comments. Guozhu Dong is the corresponding author.

References

[1] J. Auer and J. Bajorath, "Emerging chemical patterns: A new methodology for molecular classification and compound selection," *J. Chem. Inf. Model.*, vol. 46, no. 6, pp. 2502–2514, 2006.

[2] O. Akbilgic, H. Bozdogan, and M. E. Balaban, "A novel hybrid RBF neural networks model as a forecaster," *Statist. Comput.*, vol. 24, pp. 365–375, 2013.

[3] B. G. Alavijeh and H. Millan, "Point pedotransfer functions for estimating soil water retention curve," *Int. Agrophys.*, vol. 24, pp. 243–251, 2010.

[4] M. Arumugam and S. D. Scott, "EMPRR: A high-dimensional EM-based peicewise regression algorithm," in *Proc. Int. Conf. Mach. Learn. Appl.*, 2004, pp. 264–271.

[5] J. Baim, "Estimation of a change point in multiple regression models," *Rev. Economics Statist.*, vol. 79, no. 4, pp. 551–563, 1997.

[6] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[7] S. D. Bay and M. J. Pazzani, "Characterizing model errors and differences," in *Proc. 7th Int. Conf. Mach. Learn.*, 2000, pp. 49–56.

[8] K. Bache and M. Lichman. (2013). UCI machine learning repository [Online]. Available: http://archive.ics.uci.edu/ml

[9] L. Chen and G. Dong, "Masquerader detection using OCLEP: One class classification using length statistics of emerging patterns," in *Proc. Int. Inf. Process. Evolving Netw. Workshop*, 2006, p. 5.

[10] H. A. Chipman, G. I. Edward, and M. E. Robert, "BART: Bayesian additive regression trees," *Ann. Appl. Statist.*, vol. 4, no. 1, pp. 266–298, 2010.

[11] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[12] G. Dong and J. Bailey, Eds., *Contrast Data Mining: Concepts, Algorithms, and Applications* (Data Mining and Knowledge Discovery Series). Boca Raton, FL, USA: CRC, 2012.

[13] H. Drucker, C. J. C. Burges, L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines," in *Proc. Adv. Neural Inf. Process. Syst.*, 1997, vol. 9, pp. 155–161.

[14] E. Dimitriadou, K. Hornik, F. Leisch, D. Meyer, A. Weingessel, and M. Leisch. (2006). The e1071 package. Dept. Statist., Vienna Univ. Technol., TU Wien, Vienna, Austria. [Online]. Available: http://cran.r-project. org/web/packages/e1071/index.html

[15] G. Dong and J. Li, "Efficient mining of emerging patterns: Discovering trends and differences," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 1999, pp. 43–52.

[16] G. Dong, X. Zhang, L. Wong, and J. Li, "CAEP: Classification by aggregating emerging patterns," in *Proc. 2nd Int. Conf. Discovery Sci.*, 1999, pp. 30–42.

[17] U. Fayyad and K. Irani, "Multi-interval discretization of continuous-valued attributes for classification learning," in *Proc. Int. Joint Conf. Uncertainty Artif. Intell.*, 1993, pp. 1022–1029.

[18] N. Fore and G. Dong, "CPC: A contrast pattern based clustering algorithm," in *Contrast Data Mining: Concepts, Algorithms, and Applications*. Boca Raton, FL, USA: CRC Press, 2012.

[19] J. H. Friedman, "Stochastic gradient boosting," *Comput. Statist. Data Anal.*, vol. 38, no. 4, pp. 367–378, 2002.

[20] G. Ferrari-Trecate and M. Muselli, "A new learning method for piecewise linear regression," in *Proc. Artif. Neural Netw.*, 2002, pp. 444–449.

[21] D. M. Hawkins, "The problem of overfitting," *J. Chem. Inf. Comput. Sci.*, vol. 44, no. 1, pp. 1–12, 2004.

[22] C. W. Hukkelhoven, E. W. Steyerberg, J. D. Habbema, E. Farace, A. Marmarou, G. D. Murray, L. F. Marshall, and A. I. Maas, "Predicting outcome after traumatic brain injury: Development and validation of a prognostic score based on admission characteristics," *J. Neurotrauma*, vol. 22, no. 10, pp. 1025–1039, 2005.

[23] H. Kim, W. Y. Loh, Y. S. Shih, and P. Chaudhuri, "Visualizable and interpretable regression models with good prediction power," *IIE Trans.*, vol. 39, no. 6, pp. 565–579, 2007.

[24] J. Luo and A. Brodsky, "An optimal regression algorithm for piecewise functions expressed as object-oriented programs," in *Proc. Int. Conf. Mach. Learn. Appl.*, 2010, pp. 937–942.

Authorized licensed use limited to: Dr. D. Y. Patil Educational Complex Akurdi. Downloaded on June 29,2025 at 11:50:30 UTC from IEEE Xplore. Restrictions apply.

[25] J. Li and L. Wong, “Identifying good diagnostic gene groups from gene expression profiles using the concept of emerging patterns,” *Bioinformatics*, vol. 18, no. 10, pp. 1406–1407, 2002.

[26] Q. Liu and G. Dong, “CPCQ: Contrast pattern based clustering quality index for categorical data,” *Pattern Recognit.*, vol. 45, no. 4, pp. 1739–1748, 2012.

[27] H. Li, J. Li, L. Wong, M. Feng, and Y. P. Tan, “Relative risk and odds ratio: A data mining perspective,” in *Proc. ACM Symp. Principles Database Syst.*, 2005, pp. 368–377.

[28] V. E. McGee and W. T. Carleton, “Piecewise regression,” *J. Amer. Statist. Assoc.*, vol. 65, no. 331, pp. 1109–1124, 1997.

[29] S. Mao and G. Dong, “Discovery of highly differentiative gene groups from microarray gene expression data using the gene club approach,” *J. Bioinformat. Computat. Biol.*, vol. 3, no. 6, pp. 1263–1280, 2005.

[30] S. Mao and G. Dong, “Towards mining optimal emerging patterns amidst 1000s of genes,” in *Contrast Data Mining: Concepts, Algorithms, and Applications*. Boca Raton, FL, USA: CRC Press, 2012.

[31] R Core Team, “R: A language and environment for statistical computing,” R Found. Statist. Comput., Vienna, Austria. Version 2.14.1, Dec. 2011.

[32] V. Taslimitehrani and G. Dong, “A new clinical prediction method using contrast pattern aided logistic regression with application on traumatic brain injury,” in *Proc. IEEE Int. Conf. Bioinformat. Bioeng.*, Nov. 2014, pp. 283–290.

[33] J. D. Toms and M. L. Lesperance, “Piecewise regression: A tool for identifying ecological thresholds,” *Ecology*, vol. 84, no. 8, pp. 2034–2041, 2003.

[34] A. Tsanas and A. Xifara, “Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools,” *Energy Buildings*, vol. 49, pp. 560–567, 2012.

[35] Y. Wang and I. H. Witten, “Induction of model trees for predicting continuous classes,” in *Proc. 9th Eur. Conf. Mach. Learn.*, 1997, pp. 128–137.

[36] I.-C. Yeh, “Modeling of strength of high-performance concrete using artificial neural networks,” *Cement Concrete Res.*, vol. 28, no. 12, pp. 1797–1808, 1998.

Guozhu Dong received the PhD degree in computer science from the University of Southern California in 1988. He is currently a full professor at Wright State University. His main research interests are data science, data mining, and bioinformatics. He has published more than 150 articles and two books, and he holds four US patents. He received the Best Paper Award from the 2005 IEEE ICDM and the 2014 PAKDD. He has served on more than a hundred program committees of international conferences, including serving as a program committee chairs. He is a senior member of the IEEE and ACM.

Vahid Taslimitehrani received the MS degree in applied mathematics from the Iran University of Science and Technology in 2007. He is currently working toward the PhD degree at Wright State University. His current research focuses on prediction and classification techniques. He was summer intern at Explorys and Mayo. He won Best Student Paper Award at the 2014 IEEE BIBE.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.

Authorized licensed use limited to: Dr. D. Y. Patil Educational Complex Akurdi. Downloaded on June 29,2025 at 11:50:30 UTC from IEEE Xplore. Restrictions apply.