УДК 519.23

'SPLIT AND PEEL' RULE INDUCTION METHOD

© [1]Treebushny D., [1]Kotkov V., [2]Chikalov I.

[1]Institute of Mathematical Machines and System Problems
NAS Ukraine,
Prospekt Glushkova, 42, Kiev, Ukraine, 03680 GSP
e-mail: *{dima, kotkovv}@env.com.ua*

[2]Intel Corporation,
Nizhny Novgorod Lab,
30 Turgeneva st., Nizhny Novgorod, Russia, 603024
e-mail: *igor.chikalov@intel.com*

Abstract. Patient Rule Induction Method (PRIM) [2] is a rule learning procedure that seeks to locate bumps: regions in the feature space where an output variable has substantially higher values than its mean value in entire input domain. Though accepted by many practical researches the original PRIM may perform poorly on datasets containing multiple bumps. The paper proposes an addition to classical PRIM: a splitting procedure that replaces peeling to process a multimodal bump. Performance of the new method is compared with the classical algorithm on an artificial dataset simulating fault analysis problem.

Introduction

Patient rule induction method (PRIM) was proposed by Friedman and Fisher as an algorithm of optimization of expected function value. Several problems of optimization, classification, and clustering can be formulated in such a form. PRIM generates interpretable solutions – associative rules describing hypercubes in an input space. A distinctive feature of PRIM is patience – unlike other rule induction algorithms (CART [1], RIPPER [3], CN2 [4]) PRIM comes to a solution through multiple iterations. This improves precision as misdirected iterations are compensated on later stages, makes the solution more stable to small changes in data and increases a search breadth – more input variables have a chance to participate in the solution.

We applied PRIM to the analysis of root causes of yield loss in semiconductor manufacturing. While performing the experiments we discovered a property of PRIM that complicates work with multiple bumps in data. To overcome this we implemented box splitting procedure that separates bumps.

The rest of paper is organized as follows. Chapter 1 gives basic notions, describes essential details of PRIM and describes a problematic situation with multiple bumps. Chapter 2 describes the box splitting procedure and all modifications that are necessary to incorporate it in PRIM. Chapter 3 experimentally compares the modified algorithm with original PRIM on a synthetic data set modeling failure analysis problem.

1. Bump hunting

1.1. **Problem statement.** Let $\mathbf{x} = (x_1, x_2, ..., x_p)$ be input variables (real valued or categorical) and X_j be a set of possible values (domain) of x_j for $j = 1, \ldots, p$. We will call $X = X_1 \times X_2 \times \ldots \times X_p$ *input space*. Let y be a real valued output variable and $D = \{d^i = (\mathbf{x}^i, y^i), i = 1, ..., n\}$ be a random sample taken from an unknown probability

distribution $p(y, \mathbf{x})$. For given D the goal is to find such a sub-region $R \subset S$ that the mean expected output value in R is substantially higher than the mean output value in the whole input space S. We will focus on the problem of bump hunting i.e. generating constraints on input variables that caused output value to be high. It imposes two restrictions on R: its description must be interpretable by an expert and it should be representative, i.e. contain enough samples from D.

Let us call *elementary constraint* on a variable x_j any subset $s_j \subset X_j$, such as

$$s_j = \begin{cases} [a_j,\, b_j], & \text{if } x_j \text{ is numeric;} \\ \{s_{j1},\, \ldots,\, s_{jm}\}, & \text{if } x_j \text{ is categorical.} \end{cases}$$

A *box* $B = s_1 \times s_2 \times \ldots \times s_p$ is a combination of elementary constraints on all input variables. We will state that a variable x_j *participates* in the box B if $s_j \neq X_j$. For interpretability purpose R must be a box or a union of small number of boxes, i.e. $R = \bigcup_{k=1}^{K} B_k$.

Two important characteristics of a box are *output mean* and *support*. For a box B we estimate the support as $\beta_B = \frac{1}{N} |\{x^i \in B\}|$ and the output mean as $\bar{y}_B = \frac{1}{N\beta_B} \sum_{d^i \in B} y^i$. For given β_0 the problem is to find a box $B_1 = \arg\max_{b \in B, \beta_b \geq b_0} \bar{y}_b$. To find multiple bumps one should remove from D samples covered by B_1 (they are considered as "explained") and repeat the process until the mean output value of the current box becomes lower than some threshold.

1.2. **Patient rule induction method.** PRIM iteratively builds a set of boxes according to the following algorithm [2]:

1. build a single box;
2. perform box post-processing in order to simplify its description;
3. remove all data samples covered by the current box;
4. perform 1-3 until the specified number of boxes is reached or mean value of the current box is lower than a specified threshold.

A key step of the algorithm is *top-down peeling* and *bottom-up pasting* procedures, which build a single box. Top down peeling starts from a box that covers all data. At each step a small subbox b within the current box B is removed. The subbox b is chosen from a class of eligible subboxes $C(b)$ such as it maximizes some criterion $I(b)$ i.e. $b^* = \arg\max_{b \in C(b)}(I(b))$.

The set $C(b)$ contains several subboxes for each input variable. A real valued input x_j provides two subboxes: $b_{j+} = \{\mathbf{x} | x_j < x_{j(\alpha)}\}$ and $b_{j-} = \{\mathbf{x} | x_j > x_{j(1-\alpha)}\}$, where $x_{j(\alpha)}$ is α-quantile of distribution of samples $\{\mathbf{x}^i \in B\}$ by x_j. Parameter α is called *peeling fraction*; it regulates the algorithm patience and is typically set to $0.05 \div 0.10$. A categorical input x_j contributes to $C(b)$ a subbox $b_{jm} = \{\mathbf{x} | x_j = s_{jm}\}$ for each value s_{jm} encountered in B.

Three criteria I(b) differing in patience degree are considered:

1. $I(b) = \bar{y}_{B-b} - \bar{y}_B$: directly targets increase in output mean in B, the most greedy
2. $I(b) = \bar{y}_B - \bar{y}_b$: minimizes output mean of peeled subbox, i.e. rejects the “worst“ part of data, most patient;

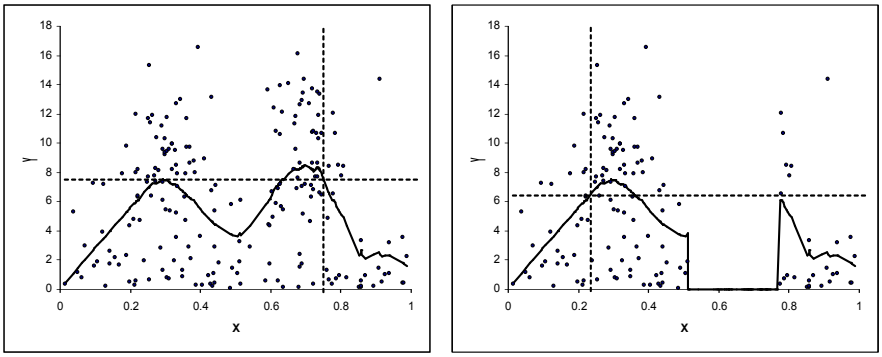

Fig. 1. (a) Scatter plot of an example data set. Dots are data samples, the solid line is a average of y by window of size $\beta_0 N$. Dashed vertical line is the box bound after peeling, grayed rectangle is the reported box after pasting. (b) The second box built after removing data contained in the first box is biased too.

3. $I(b) = \bar{y}_{B-b} - \bar{y}_b$: a sum of two previous criteria, maximizes difference between the output mean of the peeled and remained subboxes.

We used criteria 2 because it had shown best results in experiments.

Top-down peeling iteratively cuts the box until its support falls lower than a specified threshold or none of eligible subboxes increases the output mean. Bottom-up pasting is applied just after top-down peeling. It is an inverse procedure that enlarges at each step the box B by adding a subbox b^* that maximizes output mean. The class of subboxes eligible for pasting, is defined analogously to those used for peeling. A numeric variable x_j participating in B provides two subboxes that extend upper and lower condition on x_j respectively in order to cover extra $\alpha\beta_B$ samples. A categorical variable participating in B provides a subbox for each of its value not represented in B. Bottom-up pasting is over when the target mean cannot be increased by adding subboxes to B.

1.3. **Multiple bumps problem.** In case of multiple bumps PRIM can "fall between two stools". Let us demonstrate it by an example.

For the sake of simplicity assume there is a single real valued input variable x and a real valued output y. Figure 1 shows the scatter plot of a data sample and of y on x and the running average with centered window of $\beta_0 N$ samples which is used to provide spatial references. At the beginning, PRIM alternately peels outer slopes of the two peaks until reaches top of the left peak. Then it continues to peel the left face of the cube until the support threshold is reached. When peeling is over, pasting adds a part of the cut outer slope of the right cube and stops when the added cube is lower than the resulted cube mean. The box center does not coincide with the peak, thus the box corresponds to a non-optimal solution.

The problem remains after the first box is removed. "Leftovers"from the first box misdirect the algorithm in the same way and cause cutting off the outer slope of the second bump (Figure 1b).

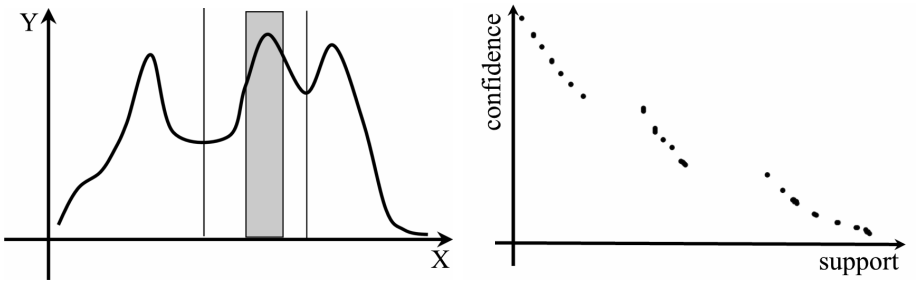

Fig. 2. Combination of peeling and splitting: (a) smoothed output curve: split points are marked with vertical lines. (b) peeling trajectory.

Let us describe the problem in general. The top-down peeling procedure can be viewed as steepest ascent: each iteration produces a step that is estimated to provide the greatest local increase to the objective function. In case of complex objective function criteria each step is seldom the optimal one in terms of leading to the ultimate solution. If each step has its own irregular bias, it is likely to be compensated by increasing steps number. This is not the case for multimodal distribution that causes a regular bias for many consecutive steps until the leading mode is localized. This introduces an error in the solution that can't be compensated by subsequent peeling steps and bottom-up pasting.

2. Modification of PRIM algorithm

The goal is to modify top-down peeling procedure in order to detect that the processed box contains multiple bumps, then to split it into two subboxes and chose one of them to continue peeling. The three topics discussed below are whether to apply splitting or peeling at the current step, how to choose the split point and which of the two halves to use further.

2.1. **Splitting criteria and choice of split point.** Algorithm searches for a split point that separates modes of conditional distribution $p(y|\mathbf{x})$. For each real valued variable x_j participating in cube B the algorithm splits elementary constraint $s_j = [a_j, b_j]$ into bins containing equal number of samples. Then the bin which has minimal mean output value is chosen to be a *splitting bin*. Decreasing of a bin size improves resolution, but increases the variance of the estimate, so we assume the bin size to be equal to a peeling fraction. The decision whether to perform splitting or peeling at the current step is taken by comparing the mean output value at the splitting point with subboxes eligible for peeling. If the mean output in the splitting bin is lower than the mean output in all eligible subboxes then splitting is performed. In other words splitting is performed when valleys between modes of conditional distribution of y over some variable become lower than the mean output value at the cube edges. It leads to a smooth peeling trajectory as shown in Figure 2.

The splitting bin is removed from the data set because it is actually a good candidate for peeling. Note that only minor changes are required in the original peeling procedure: modification only extends only a set of eligible subboxes.

2.2. **Choice of subbox to continue peeling.** As PRIM builds boxes iteratively, a rejected good candidate will be most likely located at a subsequent pass. Thus the primary target for the choice criterion is resistance to outliers. We have used a simple criterion that considers guaranteed optimization result: such a box is chosen that delivers maximal mean output value over contiguous bins covering at least $\beta_0 N$ samples.

2.3. **Modification of top down peeling procedure.** To integrate proposed changes in PRIM, the top-down peeling procedure should be changed in the following way.

1. For each real input variable x_j the algorithm splits an interval $[a_j,\, b_j]$, and constructs subboxes $b_{ji} \equiv \{\mathbf{x} \in B | t_{j(i-1)} \le x_j \le t_{ji}\}$, $a_j = t_{j0} < t_{j1} < \ldots < t_{jn_j} = b_j$ so that $\beta_{b_{jk}} \approx \alpha$ (sometimes exact equity cannot be reached due to a finite sample size and coinciding values). All subboxes b_{ji} join $C(b)$. The set of eligible subboxes provided by categorical variables remains unchanged.
2. If either the leftmost or the rightmost subbox b_{jk} is chosen for removal ($k = 0$ or $k = n$), the original peeling procedure is performed. If b_{jk} is in the middle of the interval ($0 < k < n$), define $B_l = \{\mathbf{x} | \, x_j \in [t_{j0},\, t_{j(k-1)}]\}$ and $B_r = \{\mathbf{x} | \, x_j \in [t_{jk},\, t_{jn}]\}$, $Q_l = \max\limits_{1<i<k-l-1} \bar{y}_{bji \bigcup \ldots \bigcup bj(i+l)}$ and $Q_r = \max\limits_{k<i<n-l} \bar{y}_{bji \bigcup \ldots \bigcup bj(i+l)}$. If $Q_l > Q_r$ make B_l the current box; otherwise make B_r the current box.

3. Test Results

The dataset that we have used for testing simulates semiconductors manufacturing data. A data sample corresponds to a lot: several units that are processed together at each operation. It contains five numeric variables $N_1, N_2, \ldots N_5$ describing quantitative characteristic (date, physical characteristics of process) and a categorical variable C_6 with 5 levels describing qualitative characteristics (material type, vendor, machine). A numeric response variable characterizes yield loss – a number of failed units in a lot.

A sample is drawn from a mixture of distibutions: a *base sample* characterizes a normal operation mode and three bumps characterize different failures. The base sample contains 44000 samples drawn from $5D$ Gaussian distribution by $N_1, \ldots, N_5$ with random mean vectors and random covariance matrix. Values for C6 are independently drawn from a multinomial distribution with a predefined level probability. Each bump sample contains 2000 samples drawn from $1D$ Gaussian distribution on variables participating in bump and uniform distribution on other variables. C_6 participates in one bump – its level probabilities have been changed for that case. Four categorical variables $C_7, \ldots, C_{10}$ were added to the data set that have different number of levels (from 2 to 10) and are not correlated with the response.

The response variable is drawn from a beta distribution with different parameters for the base sample and bumps. Table 1 contains distribution parameters for the base sample and all bumps.

Each algorithm is requested to report 3 boxes of support 0.03. Peeling fraction is set to be 0.01. Results are shown in the Table 2.

One can see that unlike the original algorithm the modified algorithm correctly reported all three bumps.

Table 1. Variable distribution in the test sample

Variables	Base sample	Bump 1	Bump 2	Bump 3
N_1	Mixture of	$Unif(0,1)$	$N(0.5, 0.06)$	$N(0.8, 0.06)$
N_2	50 $5D$ Gaussians,	$N(0.2, 0.06)$	$N(0.5, 0.06)$	$N(0.8, 0.06)$
N_3	with random mean	$N(0.2, 0.06)$	$Unif(0,1)$	$N(0.8, 0.06)$
N_4	vectors and	$N(0.2, 0.06)$	$Unif(0,1)$	$Unif(0,1)$
N_5	covariation matrix.	$Unif(0,1)$	$N(0.5, 0.06)$	$Unif(0,1)$
C_6	Mult(0.15, 0.2, 0.25, 0.2, 0.2)	Mult(0.2, 0.2, 0.2, 0.2, 0.2)	Mult(.004, .5, .004, .004, .488)	Mult (0.2, 0.2, 0.2, 0.2, 0.2)
Response	$beta(0.1, 10)$	$beta(1, 10)$	$beta(1, 10)$	$beta(1.5, 10)$

Table 2. Reported boxes.

PRIM	Optimized PRIM
$N_1 \in (0.079, 0.84]$, $N_2 \in (0.14, 0.87]$, $N_3 \in (0.08, 0.93]$, $N_4 \in (0.12, 0.94]$, $C_6 = 1$	$N_1 \in (0.01, 1.01]$, $N_2 \in (0.04, 0.34]$, $N_3 \in (0.04, 0.33]$, $N_4 \in (0.11, 0.27]$
$N_1 \in (0.38, 0.65]$, $N_2 \in (0.41, 0.69]$, $N_3 \in (0.03, 0.98]$, $N_4 \in (-0.09, 1.22]$, $N5 \in (0.38, 0.67]$, $C6 \in (2, 5)$	$N_1 \in (0.28, 0.62]$, $N_2 \in (0.4, 0.61]$, $N_5 \in (0.39, 0.68]$, $C_6 \in (2, 5)$
$N_1 \in (0.09, 0.89]$, $N_2 \in (0.15, 0.86]$, $N_3 \in (0.03, 0.92]$, $N_5 \in (-0.21, 0.90]$, $C_6 = 5$	$N_1 \in (0.65, 0.96]$, $N_2 \in (0.65, 0.88]$, $N_3 \in (0.66, 1.06]$

Conclusion

We proposed a modification of PRIM algorithm that overcomes the problem dealing with multiple bumps. Experimental results show that the modified algorithm correctly performs separating of multiple bumps and does not suffer from leftovers when building subsequent boxes.

Acknowledgements

This work was done in the frame of the partner contract P216 "Descriptive Supervised Optimization in High Dimensional Mixed Type Data" funded by Intel Corporation through Science and Technology Center of Ukraine (STCU).

References

1. *Breiman L., Friedman J., Olshen R. and Stone C.* Classification and Regression Trees. CityWadsworth, CityplaceBelmont, StateMA, 1984.
2. *Friedman J., Fisher N.* Bump-hunting in high-dimensional data // Statistics and Computing, V. 9, 1999, P. 123-143.
3. *Cohen W.* Fast Effective rule induction //Proceedings of the Twelfth International Conference on Machine Learning (ML95), Tahoe City, CA, USA.
4. *Clark P., Niblett. T.* The CN2 Induction Algorithm. //Machine Learning, V. 3(4), 1989, P. 261-283.

Статья поступила в редакцию 25.04.2008