Paper 156-2011

Bump Hunting With SAS®: A Macro Approach To Employing PRIM

Jennifer Sniadecki, Allos Therapeutics, Westminster, CO, USA

ABSTRACT

Many learning-based data mining methods can be used to classify particular subgroups of interest. One such method, the patient rule induction method (PRIM), is used to identify subgroups with extreme values of the target variable (i.e., outcome or dependent variable). This paper examines the PRIM algorithm and its uses, provides an example with data, and discusses SAS code that can be used to generate the PRIM algorithm.

INTRODUCTION

The use of data mining to detect commonalities in a subsample of the data (e.g., subgroup discovery, subgroup detection, responder identification, etc.) is ubiquitous in both scientific and business disciplines. When the detection is reliant upon a target variable, then the construction of the 'rules' used to define the subgroup of interest can be generated from any one or combination of supervised learning techniques. The patient rule induction method (PRIM) is one such technique that can be used when the objective is to identify a subsample of the data that shares maximum or minimum values of the target variable. The applications are plentiful, such as detecting types of customers likely to return, identifying environmental phenomena that are associated with weather anomalies, or distinguishing patient characteristics associated with the tolerance to a new drug.

PRIM is a bump hunting algorithm that was originally published in 1999 by Friedman and Fisher. The term *bump hunting* refers to the task of searching for local regions of data that exhibit an especially high value for the target function. PRIM is a nonparametric method that is capable of identifying *bumps* of the input space that either maximize or minimize the mean of the target variable. It is well suited for problems where identifying small, localized regions of the data is of interest and for high-dimensional data problems where the number of measured features *p* is much larger than the sample size *n* (i.e., wide data set or p>>n problems).

PRIM uses a top-down approach to process the data based on a single target variable. The procedure is comprised of two stages. The first stage is the peeling stage, which initially begins with all observations in the data set. The strategy is then to slowly and sequentially remove the observations based on maximizing or minimizing the mean of the target variable. Because the peeling stage is shortsighted in that it is based solely on the data available at each step, it is possible to further optimize the mean of the target variable by adding back some of the previously removed observations. This second stage, known as bottom-up pasting, is essentially the inverse of the peeling stage. The result is a set of rules that define small box-shaped regions of the data. This is one characteristic of PRIM that differs from other data mining methods used for prediction. Where other methods define the entire input space, PRIM's box-shaped regions identify only the extremes of the input space where the average target values are much larger (or smaller) than the overall average target value. The details of these two stages are provided below.

THE ALGORITHM

Below is the box induction algorithm for maximizing the mean of the target variable (Input = α, β_0). The remainder of this paper will focus on the goal of maximization.

Top-Down Peeling

(1) Begin with all (remaining) data as current box B.

(2) Define a complete class $C(b)$ of eligible subboxes for removal:

(2a) For numeric variables (x_j) remove 100(α)% observations. There will be two subboxes (b_{j-} and b_{j+}):

$$b_{j-} = \{\mathbf{x} \mid x_j < x_{j(\alpha)}\} \text{ and } b_{j+} = \{\mathbf{x} \mid x_j > x_{j(1-\alpha)}\}.$$

(2b) For categorical variables (x_j) there will be as many subboxes as categories S_j for variable x_j:

$$b_{jm} = \{\mathbf{x} \mid x_j = s_{jm}\}, s_{jm} \in S_j.$$

(3) Choose the optimal subbox b^*:

$$b^* = \arg\max_{b \in C(b)} \text{ave}[y_i \mid \mathbf{x}_i \in B - b].$$

(4) Update the current box B, $B \leftarrow B - b^*$.

(5) Repeat steps (1-4) until the support β_B of the current box B is below the threshold β_0.

Bottom-Up Pasting

(6) Begin with the peeling solution of the current box B.

(7) Define a complete class $C(b)$ of eligible subboxes for addition:

(7a) For numeric variables, extend the upper and lower boundaries of B by adding αN_B observations, where N_B = number of observations in B. Boxes are defined analogously as in (2a).

(7b) For categorical variables, values s_{jm} not represented in the current box B define subboxes b eligible for pasting.

(8) Choose the optimal subbox b^* :

$$b^* = \arg \max_{b \in C(b)} \text{ave}[y_i \mid \mathbf{x}_i \in B \cup b].$$

(9) Update the current box B, $B \leftarrow B \cup b^*$.

(10) Repeat steps (6-9) until the output mean $\overline{Y}_{B+b^*}$ begins to decrease.

This procedure is eloquently coined as a *patient* method due to the slow, stepwise mechanism by which it processes the data. Unlike some algorithms that can partition the data quickly (e.g., CART), PRIM removes a small fraction of observations at each step. For numeric variables, a maximum of only 100(α)% observations are available for removal at each iteration. The treatment of categorical variables may not always be as slow because all observations that share the same category are removed together. However, the idea of the algorithm being patient for categorical variables is maintained by restricting peeling to only one category at each iteration.

BASIC ILLUSTRATION

To further examine the peeling mechanism and demonstrate the box generation, consider the following example with binary target variable *Y* coded as 0/1 along with two generic numeric input variables (i.e., predictor or independent variables) *X1* and *X2*. It can be seen from Table 1 that the overall mean of the initial box is 0.500.

Record	*Y*	*X1*	*X2*
1	0	32.0	36.0
2	0	21.0	35.0
3	0	25.0	32.9
4	0	24.0	37.0
5	0	28.3	27.0
6	1	25.0	30.0
7	1	22.0	35.0
8	1	29.4	32.1
9	1	28.4	31.8
10	1	28.1	30.0

Table 1. 2-D Example With A Binary Response

Suppose the goal is to find the subgroup with the maximal average target value by setting the input parameters to α = 0.20 and β_0 = 0.30. By these thresholds, the search is then for the rules that comprise a box that has a target mean greater than 0.500, that contains no less than 30% of the initial data, and that is created by removing at most only 20% of the total observations from each iteration. Because both predictors are numeric, eligible subboxes for removal will include the lowest and highest values based on the α and 1 - α quantiles, respectively. To examine this, let's consider Table 2, which contains the ordered values for input variable *X1*.

Y	0	1	0	0	1	1	0	1	1	0
Input Variable (*X1*)	21.0	22.0	24.0	25.0	25.0	28.1	28.3	28.4	29.4	32.0
Order Variable	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}

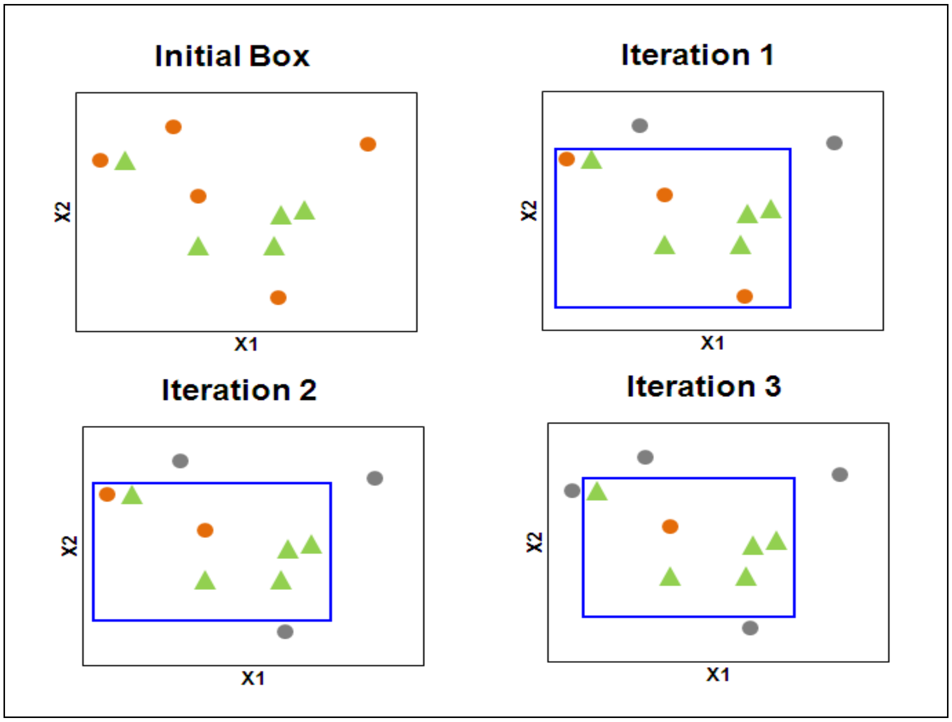
For the 20th percentile: n=10, p=0.20, g=0, j=2.

Table 2. Ordered Values of Input Variable *X1*

It is important to note that there are five definitions in SAS for computing percentiles. Used here is the default in the PROC UNIVARIATE procedure (PCTLDEF=5), which is based on the empirical distribution function with averaging; if n=number of non-missing values and x_1, x_2, ... , x_n represent the ordered values of the variable, then the (100p)th percentile is:

$$\begin{cases} 0.5(x_j + x_{j+1}) & \text{if } g = 0 \\ x_{j+1} & \text{if } g > 0 \end{cases}$$, where *j* is the integer part of *np* and *g* is the fractional part of *np*.

For the first iteration of peeling, eligible subboxes for removal for *X1* include both $b_{1-} = \{\mathbf{x} \mid X1 < 23.0\}$ and $b_{1+} = \{\mathbf{x} \mid X1 > 28.9\}$. The values of 23.0 and 28.9 correspond to the 20th and 80th percentiles, respectively. Similarly, the two eligible subboxes for removal for *X2* include $b_{2-} = \{\mathbf{x} \mid X2 < 30.0\}$ and $b_{2+} = \{\mathbf{x} \mid X2 > 35.5\}$. The optimal subbox b^* will be the one that yields the highest average value of *Y* among the four subboxes. Since removing b_{1-} will result in $\bar{Y}_{B-b_{1-}} = 0.500$, removing b_{1+} will result in $\bar{Y}_{B-b_{1+}} = 0.500$, removing b_{2-} will result in $\bar{Y}_{B-b_{2-}} = 0.556$ and removing b_{2+} will result in $\bar{Y}_{B-b_{2+}} = 0.625$, the optimal subbox is b_{2+}, and the current box will be updated by removing records 1 and 4 from Table 1. This is reflected below in the first iteration of Figure 1.



● Represents responses of 0. ▲ Represents responses of 1. Blue borders represent the peeling sequence. Darkened records are not considered in future iterations. Iteration 3 is the final solution.

Figure 1. Box Generation

After three iterations, the final solution represents 60% of the original data and has an overall target mean of 0.833. The rules defining this box are $22.0 \leq X1 \leq 29.4$ and $30.0 \leq X2 \leq 35.0$. Note that the peeling stops because no subboxes exist within the blue-bordered region that will yield an improvement in the mean of *Y* at the $\alpha = 0.20$ level; since only six observations remain after iteration 3, the next iteration is restricted to look for an improvement within the blue-bordered box by removing only one observation. Removing either extreme *X1* value will decrease the mean to 0.800. Likewise, removing the upper extreme value of *X2* results in a decrease of the mean to 0.800, and since two values share the minimum value of *X2*, removing a single observation here is not feasible. Thus, the peeling stage has ceased. Although this is a very basic example used for illustration, these same principles can easily be extended to higher dimensions.

PEELING AND PASTING DETAILS

The workhorse of the algorithm is the peeling stage. Because this method's tenet is patience, it should not be surprising that the peeling stage comprises the majority of the processing time. Factors that influence this rate are the number of input variables, their types (numeric or categorical), their characteristics (range, number of unique values or categories, missing values, etc.), and the input parameters (α, β_0). At first glance, it might seem that the smaller the value of α the better since each peeling iteration then becomes less important to the overall solution. However if α is too small, the procedure can become too sensitive to noise and thus allow unnecessary variables to be selected. The preference is left to the user, but the peeling value is typically chosen to be in the range of $0.05 \leq \alpha \leq 0.10$. The choice of β_0 dictates when the peeling stage stops. A small value can narrow in on a very small region of the data but runs the risk of overfitting while a larger value may fail to identify a unique region of the data.

During the generation of the final box, a sequence of many boxes is identified, each with their own box support, mean, and rules. This was seen above by the blue-bordered boxes in Figure 1. Since the process is incremental, the user has the freedom to identify any box from this sequence as their final box. To help with this decision, the display of the box mean vs. the box support, known as a *trajectory*, can be used. The *Prim* SAS macro produces a trajectory represented by Figure 3 below. An alternative box to the final box will not have the largest mean or smallest support, but with the loss of these traits, there may be a gain in the interpretation or generalizability of the box.

The pasting stage can contribute further gains in the identification of the area of the maximal or minimal target mean. For categorical variables, each distinct category that is not represented in the current box *B* defines subboxes *b* eligible for pasting. For numeric variables, subboxes are generated by extending the upper and lower boundaries of *B* for each distinct variable. The width of the boundary extension is determined by αN_B where N_B is the number of observations in the current box *B*. The macro parameter `paste_alpha` in the *Prim* macro allows the user the flexibility to select a different level for α during the pasting stage than what is used in the peeling stage. Therefore in the *Prim* macro, the width of the boundary extension is determined by $\alpha_{paste}N_B$ where α_{paste} refers to the value assigned to `paste_alpha`.

CROSS-VALIDATION AND MISSING VALUES

Cross-validation can be used to mitigate the effects of overfitting. Overfitting occurs when a predictive model is too specific or complex based on the observations used to construct it. If the model is applied to the data that was used to construct it, it will yield high predictive performance; however, when applied to a new set of data, the predictive performance will be poor. This results from the noise of the data being incorporated in the model and/or a lack of representative data points. A way to prevent overfitting is by splitting the data into a training sample and a testing sample. This is a basic cross-validation technique known as the holdout method. In the case of PRIM, the training sample is used to construct the boxes from the peeling and pasting sequences. Target variable means are constructed from the testing sample based on the rules dictated from the sequence of boxes in the training sample. The box with the largest test mean is then chosen as the optimal box.

Of course few data sets are free from missing values. Luckily PRIM's treatment of missing values is straightforward: missing values are treated as a suitable value of x_j. This means that for categorical inputs, missing values are simply another category, and for numeric inputs, the two subboxes generated in the peeling stage (b_{j-} and b_{j+}), are extended to three subboxes (b_{j-}, b_{j+} and b_{j0}) where $b_{j0} = \{\mathbf{x} \mid x_j = missing\}$.

BUT DO WE HAVE TO '*THINK*'?

In a nutshell, the answer is yes! We have already seen a couple of instances involving user control, but to fully understand it, we first need to further explore the interpretation of the output.

The main goal of this procedure is to produce a set of defined rules for each box. And as we have already seen, the rules are dictated by the range of values for each variable in the input space. From the example above, the rules defining the final box were [22.0, 29.4] for *X1* and [30.0, 35.0] for *X2*. Let's suppose a third variable, *X3*, that ranged from [a, b] was introduced. Also suppose that the rules defining the final box retained some subset of the input spaces for both *X1* and *X2* while retaining the entire range of *X3*, [a, b]. Then for simplicity sake, *X3* would not be considered as part of the box definition. Although there is no explicit feature selection technique inherit in PRIM, it does provide a sense of importance to the input variables on a gross level by their presence or absence in the box definition. If there is interest to further reduce or assess the relevance of the number of input variables that define a box (e.g., for improved interpretation, increased generalizability, reduction of collinear variables, etc.), the user can employ their own selection technique. The Friedman and Fisher paper provides an example using a backward stepwise selection process to assess the relevance of each input variable. The metric used is a decrease in the box mean when each variable is removed from the box definition. This process requires judgment from the user to best determine the ultimate box definition.

Until now we have only focused on the generation and definition of single boxes B_k, where k=1,…,K. But, after the first box B_1 is determined from the peeling and pasting stages, the data defining this box are removed and the process continues on the remaining data until the target means within the boxes reach the overall mean or the support becomes too small. The union of the set of boxes defines our final region *R* or *bump* where

$$R = \bigcup_{k=1}^{K} B_k .$$

However, there is opportunity for user judgment here also. An alternative definition is valid as long as the boxes defining the final bump are treated as an ordered set. For example, if the selection of the final bump is based on the region of data whose mean is greater than a threshold $\overline{Y}_0$ then

$$R = \bigcup_{\bar{y}_k > \overline{Y}_0} B_k .$$

Thus we have seen that there is control in the decision of the input parameters (*α*, β_0), in the ultimate selection of each box from the peeling/pasting sequence, in the assessment of the variables that define each box, and in the determination of the final set of boxes. This method provides the flexibility to incorporate user insight and expertise in the development of the solution.

THE ROAD MAP

Figure 2 provides a visual representation of the major steps that comprise the *Prim* macro when the objective is to maximize the mean of the target variable.

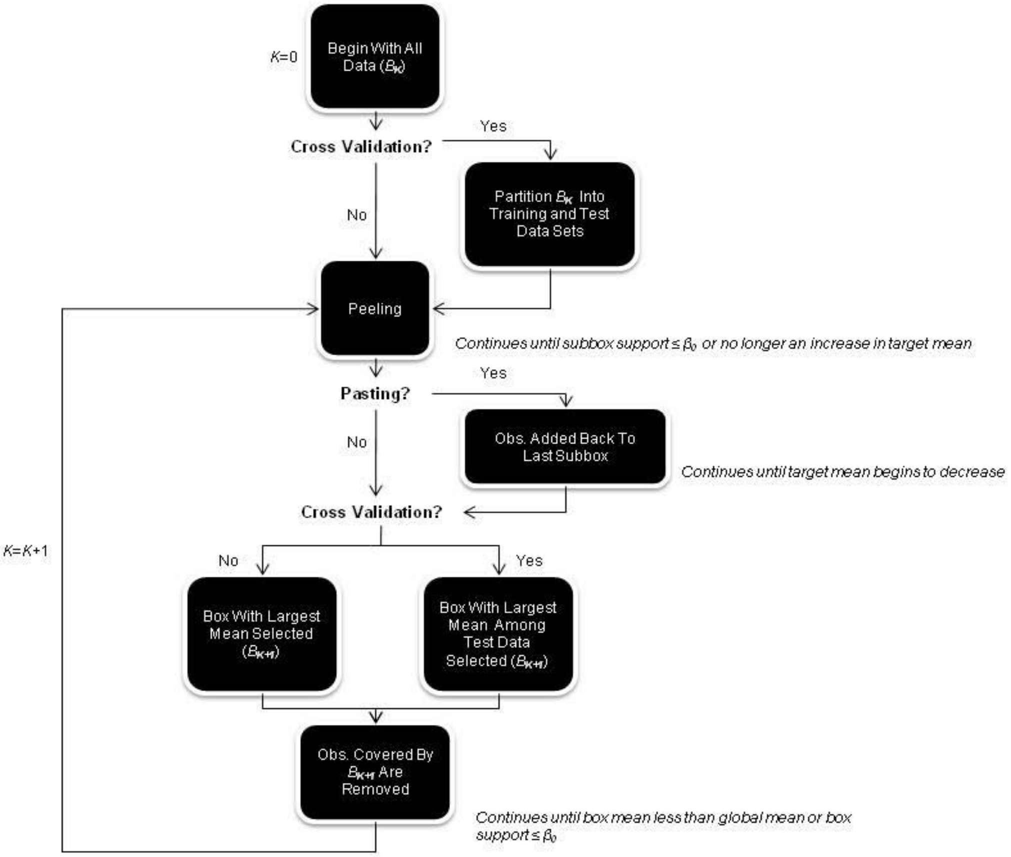

K=0
Begin With All Data (B_K)
Cross Validation?
Yes
No
Partition B_K Into Training and Test Data Sets
Peeling
Continues until subbox support ≤ β₀ or no longer an increase in target mean
Pasting?
Yes
No
Obs. Added Back To Last Subbox
Continues until target mean begins to decrease
Cross Validation?
K=K+1
No
Yes
Box With Largest Mean Selected (B_K+1)
Box With Largest Mean Among Test Data Selected (B_K+1)
Obs. Covered By B_K+1 Are Removed
Continues until box mean less than global mean or box support ≤ β₀

Figure 2. Flowchart Of The *Prim* SAS Macro

UTILIZATION WITH SAS

To see how the *Prim* macro works, let's look at an example using a prediction data set of forest fires from Portugal. The data set is available at: http://archive.ics.uci.edu/ml/datasets/Forest+Fires from the UCI Machine Learning Repository. It contains 517 records and 13 variables described in Table 3. For this example, suppose there will be an increased effort over the next year to reduce the total land mass destroyed by forest fires in this region. In order to have an improved fire prevention plan including resource availability (e.g., location of firefighters and preventive equipment), there is a need to better understand what, if any, characteristics are common to those fires that resulted in largest total burned area (*area*). For the purposes of this example, the Fire Weather Index variables (*FFMC, DMC, DC, ISI*) will be excluded, while the geographic numeric variables (*X* and *Y*), the temporal categorical variables (*month* and *day*), and the meteorological numeric variables (*temp, RH, wind, rain*) will be retained.

Variable	Description
X	x-axis coordinate (from 1 to 9)
Y	y-axis coordinate (from 1 to 9)
month	Month of the year (January to December)
day	Day of the week (Monday to Sunday)
FFMC	Fuel Moisture Code
DMC	Duff Moisture Code
DC	Drought Code
ISI	Initial Spread Index
temp	Outside temperature (in ˚C)
RH	Outside relative humidity (in %)
wind	Outside wind speed (in km/h)
rain	Outside rain (in mm/m^2)
area	Total burned area of the forest (in ha)

Table 3. Forest Fires Data Set Variable Description

The only requirements to run the *Prim* macro are Base SAS and SAS/GRAPH®. Table 4 provides a description of all of the parameters that are available to be passed into the *Prim* macro. Let's submit the following call:

```
%prim(ds=fires, goal=max, target=area, support=0.05)
```

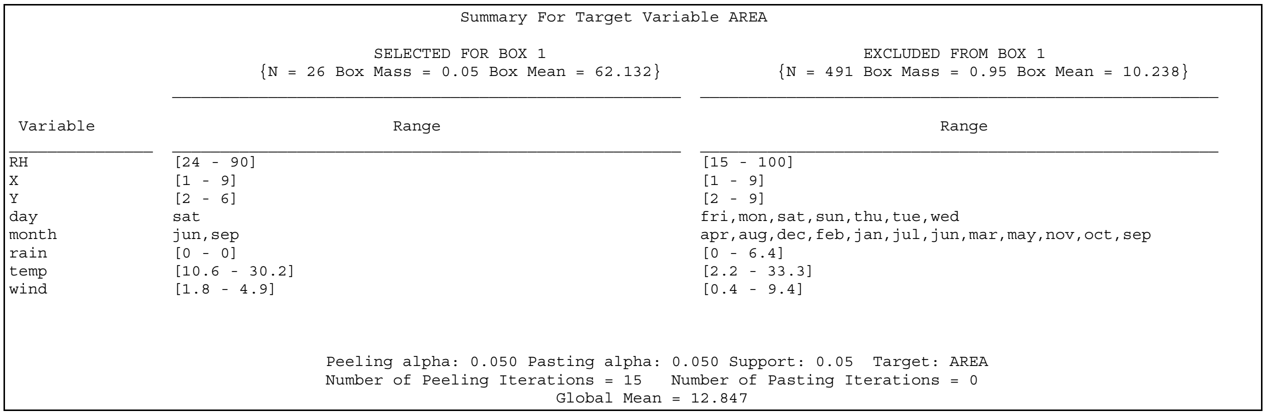
and apply the default settings along with specifying that we will be maximizing the mean of the variable *area*. Note that by default, the values for α and α_{paste} are the same at the 0.05 level, and in addition to the output (Output 1 and 2), a trajectory graph will generated for every box while cross-validation is turned off.

Parameter	Default	Description (Valid Values)
ds		Source data set. <Is Required>.
peel_alpha	0.05	Peeling alpha-quantile value. (0.00, 1.00). <Is Required>.
paste_alpha	0.05	Pasting alpha-quantile value. [0.00, 1.00). A value of 0.0 turns off pasting. <Is Required>.
goal		Is goal minimum or maximum of target function. (MIN or MAX). <Is Required>.
target		The target or dependent variable name. <Is Required>.
support		Minimum value for the proportion of data in box. (0.00, 1.00). <Is Required>.
graph	Y	Option to display a trajectory graph of box mean vs. support (Y or N).
cv	N	Option to perform the holdout method of cross-validation (Y or N).
log_path		Optional destination of where to redirect log. Recommended so that log window does not need to be cleared during processing. An example is C:\log.
prop	0.67	Proportion of observations selected for the training data set and remainder are used for the test data set (0.00, 1.00). <Is Required if cv=Y>.
seed	1	Random number seed used for splitting training and test data sets. [0, 2147483646] and must be an integer. <Is Required if cv=Y>.

Table 4. *Prim* SAS Macro Parameter Descriptions

In the original data set of 517 observations, the average total burned area is 12.8 hectares. The first box identified by the macro describes the most destructive fires in terms of land mass. As seen in Output 1, the average total burned area of this first box is 62.1 hectares and describes 5% of the total data set. The rule set for box 1 is given to the left side of the output: 24% ≤ *RH* ≤ 90%, *Y* ≤ 6, *day* = Sat, *month* = {Jun, Sept}, *rain* = 0.0 mm/m^2, 10.6˚C ≤ *temp* ≤

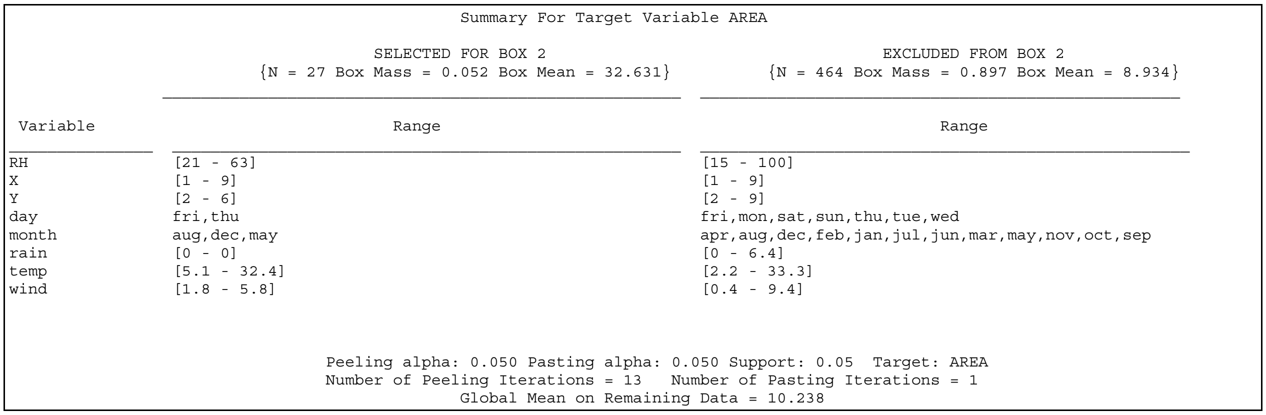
30.2˚C, and 1.8km/h ≤ *wind* ≤ 4.9km/h. After the first box has been identified, the data points contained in it are removed. From right side of Output 1, we can see that the average total burned area is 10.2 hectares in the remaining 95% of the data. The next box (Output 2) identifies a subset of this remaining data where the average total burned area is 32.6 hectares with a support of 5.2%. The rules describing this box are: 21% ≤ *RH* ≤ 63%, *Y* ≤ 6, *day* = {Thu, Fri}, *month* = {May, Aug, Dec}, *rain* = 0.0 mm/m^2, 5.1˚C ≤ *temp* ≤ 32.4˚C, and 1.8km/h ≤ *wind* ≤ 5.8km/h. This process continues producing six boxes that together comprise 35% of the total data. The remaining 65% of the data have an average total burned area of 5.3 hectares (not shown).



```
                                        Summary For Target Variable AREA

                               SELECTED FOR BOX 1                                    EXCLUDED FROM BOX 1
                     {N = 26 Box Mass = 0.05 Box Mean = 62.132}         {N = 491 Box Mass = 0.95 Box Mean = 10.238}
                 ____________________________________________________  _____________________________________________________

 Variable                                 Range                                                   Range
______________   ____________________________________________________  _____________________________________________________
RH               [24 - 90]                                             [15 - 100]
X                [1 - 9]                                               [1 - 9]
Y                [2 - 6]                                               [2 - 9]
day              sat                                                   fri,mon,sat,sun,thu,tue,wed
month            jun,sep                                               apr,aug,dec,feb,jan,jul,jun,mar,may,nov,oct,sep
rain             [0 - 0]                                               [0 - 6.4]
temp             [10.6 - 30.2]                                         [2.2 - 33.3]
wind             [1.8 - 4.9]                                           [0.4 - 9.4]



                          Peeling alpha: 0.050 Pasting alpha: 0.050 Support: 0.05  Target: AREA
                          Number of Peeling Iterations = 15   Number of Pasting Iterations = 0
                                                  Global Mean = 12.847
```

Output 1. First Box Induced By The *Prim* SAS Macro

Depending on the objective and the data, the first box may be sufficient for the descriptive goal. However, more likely it will be advantageous to combine the results all of the boxes or possibly just the first few boxes from the set of total boxes uncovered. We have seen that box 1 covers only 5% of the data, but it uncovers the ranges of the variables that contribute to a total burned area of over 62 hectares. This is nearly five times the average total burned area in this data set. If the preventative measures are to be focused on these unusually large fires, then the rule set defining box 1 may be sufficient. However, if 5% of the data is deemed too small for appropriate actions to be taken and 32 hectares is still considered a large enough burned area to meet our objective, then region describing the first two boxes, $R \leftarrow B_1 \cup B_2$, can be used. This region collectively covers 10.3% of the data. Here, the interpretation is that any records falling into rule set 1 or into rule set 2 contribute to an average total burned area of at least 32.6 hectares.



```
                                        Summary For Target Variable AREA

                               SELECTED FOR BOX 2                                      EXCLUDED FROM BOX 2
                     {N = 27 Box Mass = 0.052 Box Mean = 32.631}         {N = 464 Box Mass = 0.897 Box Mean = 8.934}
                ______________________________________________________  __________________________________________________

 Variable                                 Range                                                   Range
______________   ____________________________________________________  _____________________________________________________
RH               [21 - 63]                                             [15 - 100]
X                [1 - 9]                                               [1 - 9]
Y                [2 - 6]                                               [2 - 9]
day              fri,thu                                               fri,mon,sat,sun,thu,tue,wed
month            aug,dec,may                                           apr,aug,dec,feb,jan,jul,jun,mar,may,nov,oct,sep
rain             [0 - 0]                                               [0 - 6.4]
temp             [5.1 - 32.4]                                          [2.2 - 33.3]
wind             [1.8 - 5.8]                                           [0.4 - 9.4]



                          Peeling alpha: 0.050 Pasting alpha: 0.050 Support: 0.05  Target: AREA
                          Number of Peeling Iterations = 13   Number of Pasting Iterations = 1
                                        Global Mean on Remaining Data = 10.238
```

Output 2. Second Box Induced By The *Prim* SAS Macro

The trajectory graph provided captures the path of mean versus support for every box. Figure 3 represents this display for box 2. The peeling steps are marked by "+" and the pasting steps by "o". Since each step is a sequence in the generation of the final box, the user can choose to select an alternative to the final box. This graph can be used as an aid to assess the tradeoff between an increase in box mean with a decrease in box support.

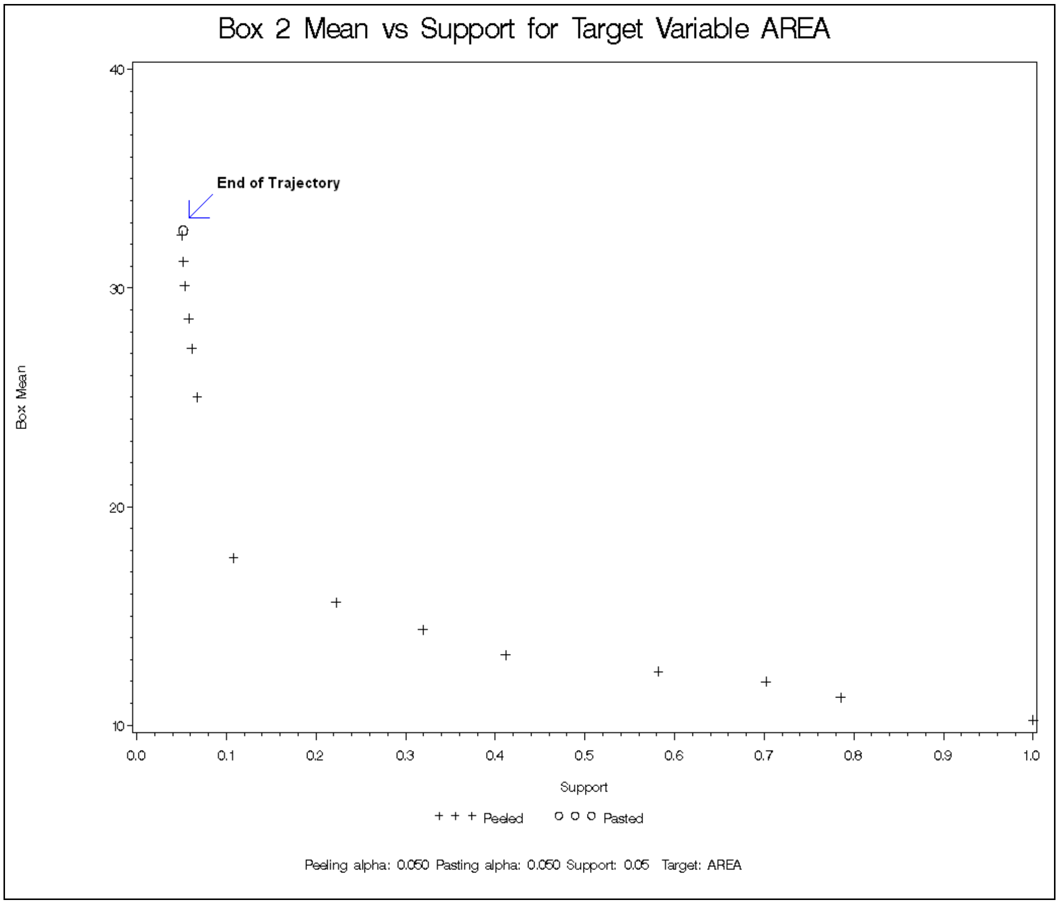


Figure 3. Trajectory For Box 2

A FEW NOTES ON THE *PRIM* MACRO

If cross-validation is selected, the representation is very similar to what is presented above. The output displays the target mean, the sample size, and the box support for both the training and test data sets. The trajectory graph displays the box mean and support of the test data based on the rules determined from the training data.

All data sets generated and processed by the macro are located in the *Work* data library. At the macro's termination, these data sets remain available in the *Work* data library. At the top of the next submission they are deleted. This feature provides effortless identification of the rules defining each box in the box generation sequence, thereby facilitating the selection process of an alternative box. The specific data sets are named as *_boxk_peel* and *_boxk_paste* where *k = the incremental box number*. Corresponding to Figure 3, the data set *_box2_peel* contains the rules defining each of the 13 peeling iterations represented by "+" and *_box2_paste* contains the rule defining the 1 pasting iteration represented by "o".

The *Prim* macro processes all variables in the data set; therefore, any extraneous variables not appropriate for searching (e.g., record number) should be removed from the source data set. All variables are treated as numeric or categorical based on their formats. During the peeling stage, if more than one subbox shares the same mean value, the subbox with the smallest support will be selected as the optimal subbox *b** for removal to promote patience. The peeling stage ceases at the first occurrence of either the support β_B of the current box below the threshold β_0 or when there is no longer an increase in box mean. Missing values are handled as described above where missing values are simply another category for categorical inputs, and for numeric inputs, three subboxes (b_{j-}, b_{j+} and b_{j0}) are considered.

Lastly, the *Prim* macro chimes a short series of beeps when it has concluded. This allows the user to return to other work after submitting the macro and receive notification via their PC speakers when macro has finished.

CONCLUSION

The *Prim* macro provides the SAS data analyst a way to explore a data mining method without having to use Enterprise Miner^{TM}. This macro is based on the patient rule induction method, which generates rules for a series of box-shaped regions of the data based on maximizing or minimizing the mean of the target variable. It is appropriate for use with binary and continuous target variables and a good resource for high-dimensional data problems. This paper provided an introduction to this method and discussion of the macro and its output. For a more detailed examination of PRIM, including applications and in-depth comparisons to other methods, please see the references provided below.

REFERENCES

- Cortez, P. & Morais, A. (2007). "A Data Mining Approach to Predict Forest Fires using Meteorological Data." In J. Neves, M. F. Santos and J. Machado Eds., New Trends in Artificial Intelligence, Proceedings of the 13th EPIA 2007 - Portuguese Conference on Artificial Intelligence. pp. 512-523. Guimaraes, Portugal. APPIA, ISBN-13 978-989-95618-0-9. Available at: http://www.dsi.uminho.pt/~pcortez/fires.pdf.
- Frank, A. & Asuncion, A. (2010). UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science. Available at: http://archive.ics.uci.edu/ml.
- Friedman, J.H. & Fisher, N.I. (1999). "Bump hunting in high-dimensional data." *Statistics and Computing (9)*. pp. 123-143.
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction (2nd ed).* New York: Springer-Verlag.
- SAS Institute Inc. (2010). Base SAS® 9.2 Procedures Guide: Statistical Procedures, Third Edition. Cary, NC: SAS Institute Inc.
- Unknown Author. "Rule induction by bump hunting". *Utrecht University*. Available at: http://www.cs.uu.nl/docs/vakken/adm/bump.pdf.

SELECTED REFERENCES WITH APPLICATIONS USING PRIM

- Kehl, V. & Ulm, K. (2006). "Responder Identification In Clinical Trials With Censored Data". Computational Statistics & Data Analysis (50). pp. 1338-1355. Amsterdam: Elsevier Science Publishers.
- Liu, X., Minin, V., Huang, Y., Seligson, D. & Horvath. S. (2004). "Statistical Methods for Analyzing Tissue Microarray Data". *Journal of Biopharmaceutical Statistics (14)*. pp. 671-685.
- Ulm, K., Kriner, M., Eberle, S., Reck, M. & Hessler, S. (2006). Statistical Methods to Identify Predictive Factors. In J. Crowley & D. Ankerst (Eds.), *Handbook of Statistics in Clinical Oncology (2nd ed)*. pp. 335-344. Boca Raton, FL: Taylor & Francis Group, LLC.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Additionally, for a copy of the macro discussed in this paper, please contact the author at:

Name: Jennifer Sniadecki
Enterprise: Allos Therapeutics
Address: 11080 CirclePoint Road Suite 200
City, State ZIP: Westminster, CO 80020
E-mail: jsniadecki@allos.com or lambertjen@gmail.com