Flexible patient rule induction method for optimizing process variables in discrete type

Il-Gyo Chong a, Chi-Hyuck Jun b,*

a Semiconductor R&D Center, Samsung Electronics Co., Ltd. San 16 Bannwol-dong, Hwasung 445-701, Republic of Korea
b Department of Industrial and Management Engineering, Pohang University of Science and Technology, San 31 Hyoja, Pohang 790-784, Republic of Korea

Abstract

This paper deals with process optimization, which establishes the optimal settings of process variables to achieve a better quality. To this end, the patient rule induction method (PRIM), widely used in various application areas, could be adopted. However, the PRIM may fail to provide successful solutions when some process variables are in discrete types. Thus, we propose a new PRIM-like method specially to deal with ordinal discrete variables. For an illustrative purpose, the proposed method is applied to a real steel-making process. Also, performance of the proposed method is compared with the original PRIM through an extensive simulation using artificial data sets.

Keywords: Data mining; Ordinal data; Process optimization; Rule induction

1. Introduction

Determining the optimal process settings is important to the process engineers who are eager to achieve high quality of products. For this purpose, two approaches are available: (i) functional approach, which models the relationship between quality and process variables and then optimizes the estimated model to determine the suitable settings of process variables and (ii) non-functional approach, which derives the suitable ranges of process variables, via data mining techniques, directly from the historical data without constructing a functional model.

The functional approach is often risky, since the resulting solution could be quite far from the optimum when a poor functional model is optimized (see Xu & Albin, 2003). Also, constructing a good model with historical data is sometimes difficult, particularly when there is no apparent relationship between quality and process variables and some noisy information is involved in the data.

The non-functional approach could give a safer and more informative solution even when the noisy historical data is used because (i) its solution is a part of the historical data, hence the solution is guaranteed by past knowledge and (ii) the solution is provided in the form of intervals, hence the controlling could be easier. A fuzzy logic such as in Iqbal, He, Li, and Dar (2007) or Zarandi, Turksen, and Kasbi (2007) can be used.

As a new non-functional approach, we introduce and modify the patient rule induction method (PRIM), originally proposed by Friedman and Fisher (1999). The PRIM discovers a combination of intervals, called as a box, for process variables at which settings higher quality values were observed.

The PRIM starts with a box large enough to contain all observations of process variables. Each observation is classified as good or bad depending on its quality value. The one face of the box is peeled off in order to remove bad observations as many as possible under the constraint that the number of observations peeled away must be less than the 100\% of the observations currently contained in the box. This peeling procedure is sequentially...
conducted until the number of remaining observations falls below the 100β percent of all observations. In the above, the PRIM has two parameters: a peeling parameter α and a stopping parameter β, both ranging from zero to one. The detailed issues are found in Friedman and Fisher (1999).

In spite of its recent advent, several successful applications have been reported in various areas, such as geology (Friedman & Fisher, 1999; Griffin, Fisher, Friedman, O’Reilly, & Ryan, 2002), marketing (Friedman & Fisher, 1999), management (Silver et al., 2001), finance (Becker & Fahrmeir, 2001), medicine (Kehl & Ulm, 2006), bioinformatics (Cole, Galic, & Zack, 2003; Liu, Minin, Huang, Seligson, & Horvath, 2004), and process optimization (Chong & Jun, 2005a).

Success of the PRIM is crucially due to its ‘patient strategy’ having a small value of the peeling parameter α (Chong & Jun, 2005a). The detailed issues are found in Friedman and Fisher (1999), management (Silver et al., 2001), finance (Becker & Fahrmeir, 2001), medicine (Kehl & Ulm, 2006), bioinformatics (Cole, Galic, & Zack, 2003; Liu, Minin, Huang, Seligson, & Horvath, 2004), and process optimization (Chong & Jun, 2005a).

2. Problem formulation

This section first describes the historical data that are assumed to be available and defines the box of process variables we want to obtain as a solution. Then, we formulate an optimization problem for obtaining the optimal settings of process variables.

2.1. Historical data

Suppose that the following historical data having N observations are available:

\[ \{y_i, x_i\}, i = 1, 2, \ldots, N, \]

where \(y_i\) and \(x_i = (x_{i1}, x_{i2}, \ldots, x_{ip})\) are respectively the quality and \(p\) process variables for the \(i\)th observation. Here, process variables are assumed to be ordinal discrete type, where a process variable \(x_j, (j = 1, 2, \ldots, p)\) takes one of \(m(j)\) values as follows:

\[ x_j \in S_j = \{d_{j1}, d_{j2}, \ldots, d_{jm(j)}\}, \]

where \(d_{j1} < d_{j2} < \cdots < d_{jm(j)}\).

We do not deal with the case that some variables are in discrete type and the others are in continuous type. This case however can be handled by the combined use of the existing PRIM and the proposed one.

The quality variable \(y\) indicates the quality of products. It may either be discrete (e.g. \(y = 1\) if quality is high and 0, otherwise) or continuous (e.g. \(y\) is a hardness of a steel sheet with a target value.). In the latter case, \(y\) will be transformed so that a larger value indicates a higher quality. For example, in the nominal-the-best case, a simple transformation would be

\[ y_i \leftarrow -(y_i - T)^2 \quad (i = 1, 2, \ldots, N), \]

where \(T\) denotes the target of the quality variable.

2.2. Box and related statistics

We define a p-dimensional box \(B\) as the intersection of sub-ranges of process variables such that

\[ B = [l_1, u_1] \times [l_2, u_2] \times \cdots \times [l_p, u_p], \]

where \([l_j, u_j]\) is a sub-range of the process variable \(x_j, l_j \leq S_j, u_j \leq u_p, (j = 1, 2, \ldots, p).\)

When the \(i\)th observation of process variables is enclosed within a box \(B\) (i.e. \(l_j \leq x_{ij} \leq u_1, l_2 \leq x_{i2} \leq u_2, \ldots, l_p \leq x_{ip} \leq u_p\) are satisfied), it is denoted by \(x_i \in B\).

There are two important statistics regarding the box \(B\). The first one is the support, which announces the proportion of observations contained in the box, given by

\[ \beta_B = \frac{1}{N} \sum_{i=1}^{N} 1(x_i \in B), \]

where the function \(1(\cdot)\) takes one when the argument is true and zero, otherwise. Obviously, it ranges from zero to one.

The second statistic is the box objective to be maximized, which is given by

\[ \text{Obj}_B = \text{Ave}(y | x_i \in B), \]

where \(\text{Ave}(\cdot)\) takes the average of \(y\) for the observations satisfying the arguments.

2.3. Optimization problem

We may formulate our problem as an optimization problem given by below that maximizes the box objective to determine the optimal box \(B\) or \(l_j\) and \(u_p, (j = 1, 2, \ldots, p)\) under a constraint on the support of the box \(B\):

\[ \max_{x_B} \text{Obj}_B \]

subject to \(\beta_B \geq \beta\),

where \(\beta\) indicates a desired minimum value of support.
Essentially, we may find the best box satisfying (7) through an exhaustive search by evaluating all possible boxes constructed from combinations of distinct values of process variables. However, the exhaustive search may be impossible, even when the number of variables is moderate because the total number of boxes to compare is \( \prod_{j=1}^{m} \frac{m_j}{m_j(m_j+1)} \). When there are 10 process variables each having 10 distinct values (i.e. \( p = 10, m(j) = 10 \)), the exhaustive search should compare about \( 2.53 \times 10^{17} \) boxes.

3. Flexible patient rule induction method (f-PRIM)

3.1. Basic idea

The proposed algorithm starts with a large box including all observations, and then iteratively reduces the box by peeling the lower or upper side of the \( j \)th process variable \( x_j \). When deciding the peeling fraction of the box and the peeling order of process variables, we adopt the concept of the patient strategy from the PRIM (Friedman & Fisher, 1999).

The patient strategy is to peel the box so that the fraction of observations excluded from the reduced box is less than \( x \) at each peeling; usually, a ‘small’ value in range of 0.05 \( \leq x \leq 0.1 \) is suggested. Since the use of a small value requires more running time, it is named as ‘patient strategy.’ Since only a small number of observations are removed at each peeling, false decisions in earlier peelings could be mitigated by later peelings.

In the case of discrete-type process variables, it is sometimes impossible to peel less than \( x \) fraction of observations at each peeling, since there might be more than \( x \) fraction of observations with the same value. Thus, the proposed method allows peeling more than \( x \) fraction of observations off, and uses the weighted sum of increase of box objective and decrease of support when evaluating the peeling. Since the patience can be controlled flexibly with a weight parameter, we name the proposed method as f-PRIM. The detailed explanation is given in Section 3.2.

3.2. Algorithm

Step 1 (Data preparation). For the purpose of avoiding over-fitting, we randomly split historical data \( \{(y_i, x_i), i = 1, 2, \ldots, N\} \) into a learning set of \( \{(y_i, x_i)^L, i = 1, 2, \ldots, N_L\} \) and a test set of \( \{(y_i, x_i)^T, i = 1, 2, \ldots, N_T\} \) where \( N_L + N_T = N \). A recommended ratio of \( N_L \) to \( N_T \) is 3:2 or 2:1. Use the learning set for Steps 2–5.

Step 2 (Initialization). Define the initial box \( B_1 \) as

\[
B_1 = [l_{11}, u_{11}] \times [l_{12}, u_{12}] \times \cdots \times [l_{1p}, u_{1p}],
\]

where \( l_{ij} = d_{ij} \) and \( u_{ij} = d_{mij} \), \( (j = 1, 2, \ldots, p) \), so that all observations are included within the box \( B_1 \). Naturally, the support of \( B_1 \) is one. Set \( k = 1 \) and the current box as \( B_1 \).

Step 3 (Generation of candidates). From the current box, generate \( 2 \times p \) candidate boxes, \( \{C_{11}, C_{12}, C_{13}, \ldots, C_{pp}\} \) by peeling one side of the current \( p \)-dimensional box. The \( C_{ij} \) \( (j = 1, 2, \ldots, p) \) is the box that results from peeling the lower side of the \( j \)th process variable of the current box \( B_k \) such that

\[
C_{ij} = [l_{k1}, u_{k1}] \times \cdots \times [l_{kj}, u_{kj}] \times \cdots \times [l_{kp}, u_{kp}],
\]

where

\[
l_{kj}^{\rightarrow} = \left\{ \begin{array}{ll}
\min\{x|l_{kj} < x \leq u_{kj} \text{ and } x \in S_j\} & \text{if } l_{kj} = x_{j(a)} \\
\min\{x|x_{j(a)} < x \leq u_{kj} \text{ and } x \in S_j\} & \text{if } l_{kj} < x_{j(a)}.
\end{array} \right.
\]

Here, \( x_{j(a)} \) is the \( a \)-quantile of \( x_j \) values, which are enclosed within the current box \( B_k \) and \( \min(x \mid \cdot) \) takes the minimum of \( x \) values satisfying the argument. Similarly, the \( C_{ij} \) \( (j = 1, 2, \ldots, p) \) is created by peeling the upper side of the \( j \)th process variable of the current box \( B_k \) such that

\[
C_{ij} = [l_{k1}, u_{k1}] \times \cdots \times [l_{kj}, u_{kj}^{\leftarrow}] \times \cdots \times [l_{kp}, u_{kp}],
\]

where

\[
u_{kj}^{\leftarrow} = \left\{ \begin{array}{ll}
\max\{x|l_{kj} \leq x < u_{kj} \text{ and } x \in S_j\} & \text{if } x_{j(1-a)} = u_{kj} \\
\max\{x|l_{kj} \leq x \leq x_{j(1-a)} \text{ and } x \in S_j\} & \text{if } x_{j(1-a)} < u_{kj}
\end{array} \right.
\]

and \( \max(x \mid \cdot) \) takes the maximum of \( x \) values satisfying the argument. Now, calculate the support and the box objective of \( 2 \times p \) candidate boxes using (5) and (6).

Step 4 (Selecting of the best candidate). First, construct an active set \( A \) with the candidate boxes having larger box objectives and smaller supports than the current box as

\[
A = \{C|\text{Obj}_C > \text{Obj}_{B_k}, \beta_C < \beta_{B_k}, \text{ and } C \in \{C_{11}, C_{12}, \ldots, C_{pp}\}\}.
\]

Second, transform the box objective and support of a candidate box involved in the active set \( A \) into the following two measures, respectively:

\[
d_{\text{box objective}} = \left( \frac{\text{Obj}_A - \text{Obj}_{B_k}}{\text{Obj}_A} \right)^2
\]

and

\[
d_{\text{support}} = \left( \frac{\beta_A}{\beta_{B_k}} \right)^2.
\]

In (14), \( \text{Obj}_M \) is the maximum value of the box, which may depend on the problem. For example, if quality variable is coded as one or zero, then \( \text{Obj}_M \) is one; if quality variable is defined as in (3), then \( \text{Obj}_M \) is zero. Both \( d_{\text{box objective}} \) and \( d_{\text{support}} \) range from zero to one.

Now, increase \( k \) by one (i.e. \( k \leftarrow k + 1 \)). Then, replace the current box with the candidate box having the largest weighted sum of \( d_{\text{box objective}} \) and \( d_{\text{support}} \) such as
\[ B_k = \arg \max_{A} \{ \lambda \cdot d_{\text{box objective}} + (1 - \lambda) \cdot d_{\text{support}} \}, 0 \leq \lambda \leq 1, \]

(16)

where \( \lambda \) is a weight to be determined.

**Step 5 (Stopping condition).** If the support of \( B_k \) is greater than the stopping parameter \( \beta \) (e.g. \( \beta = 0.05 \)), then go to Step 3 (We are allowed to do more peeling). Otherwise, go to Step 6 with a sequence of boxes \( B_1, B_2, \ldots, B_k \).

**Step 6 (Selection of the best box).** Recalculate the support and box objective of \( B_1, B_2, \ldots, B_k \), using the test set prepared in Step 1, and then choose the box with the largest \( \text{Obj}_B \) among boxes whose support is greater than \( \beta \).

### 3.3. How to determine the weight parameter \( \lambda \)

The weight \( \lambda \) in (16) may influence the value of box objective and support, which ranges from zero to one. As \( \lambda \) approaches one, the proposed algorithm becomes greedy in increasing the box objective at each step. As \( \lambda \) approaches zero, the proposed algorithm works slowly in

![Fig. 1. Dot-plots of the selected process variables.](image-url)
increasing the box objective. The proper value of \( \lambda \) can be determined by a cross-validation method in Hastie, Tibshirani, and Friedman (2001).

4. Case study

The process optimization problem encountered in a magnetic steel process was used to illustrate the proposed algorithm. The quality of magnetic steel sheets depends on the amount of core loss (W/kg). Magnetic steel sheets with lower core loss are sold at a higher price. To identify an operating process condition that produces low core loss sheets, we used historical data of one quality variable and 146 process variables with 1432 observations. Among the 146 process variables, we selected six important process variables through some variable selection methods in Chong and Jun (2005b). To apply the proposed method, we coded the quality variable with one if the core loss is less than 1.07 and zero, otherwise. Hence, the coded quality variable takes the value of one on 281 observations (19.6%) among 1432 observations.

Fig. 1 shows dot-plots of the selected important process variables, which are standardized to have mean zero and unit variance for the conditional reason. The 1432 observations of \( x_1, x_2, x_3, x_4, x_5 \) and \( x_6 \) consist of 48, 17, 7, 21 and 31 distinct ordinal values, respectively.

Firstly, the historical data set was randomly split into a learning set (60%) and a test set (40%). As mentioned before, the learning set was only used for generating a sequence of boxes and the test set was used for evaluating the boxes. Table 1 shows an example (\( \alpha = 0.1, \lambda = 0.1 \)) of how boxes were generated sequentially and how the best box was chosen using a learning and test set, respectively. As seen in Table 1, we started with a six-dimensional initial box \( B_1 \), which contains all observations of learning set. The box objective of \( B_1 \) is 0.210 (i.e. only 21% of learning data are linked to high quality). Then, \( B_2 \) was generated from \( B_1 \) by peeling the lower side of \( x_3 \) and \( B_3 \) from \( B_2 \) by peeling the upper side of \( x_3 \). In this way, a total of 37 boxes were generated until the support falls just below 0.05. Then, we recalculated box objectives and supports of 37 boxes using test set. Finally, we subjectively chose the 30th box, which has the maximum \( \text{Obj}_B \) subject to \( \beta_B \geq 0.1 \), as the final box when \( \alpha = 0.1 \) and \( \lambda = 0.1 \) are used, since we believe the final box should contain at least 10% of observations.

Secondly, to determine the proper values for \( \lambda \) and \( \alpha \), we repeated the whole procedure with different combinations of \( \lambda \) and \( \alpha \) and the box objectives based on test set are shown in Table 2. When \( \lambda = 0.4 \) and \( \alpha = 0.05 \), the box having the largest box objective was obtained and is defined as \(-0.741 \leq x_1 \leq 1.167, -2.234 \leq x_2 \leq 0.657, -1.463 \leq x_3 \leq 0.704, -4.602 \leq x_4 \leq 0.338, -0.726 \leq x_5 \leq -0.378 \), and \(-2.710 \leq x_6 \leq -1.507 \). This box includes 145 observations (10.1%) among total 1432 observations, and box objective is 0.615, which means that 61.5% of products would be sold at premium if the process is controlled within the box. Currently, only 19.6% of products are sold at a higher price.

For comparison, we also apply PRIM to the same learning and test set. Box objectives and supports, calculated by test set, are shown in Table 3. The best box was obtained when a peeling parameter \( \alpha = 0.2 \). This box includes 171 observations among 1432 observations (12%), and box objective is 0.522, which is smaller than that of the proposed method. As predicted, PRIM failed to discover a small box when \( \alpha = 0.1 \). It just found a large box whose support is 0.49.

5. Simulations

5.1. Data generation

In order to validate the proposed \( f \)-PRIM more objectively and to compare with PRIM, we conduct simulations with artificial data sets. To simulate a real process introduced in Section 4 when generating artificial data sets, we consider six process variables, each consisting of five ordinal discrete values, 1, 2, 3, 4, and 5. We assume that the best box defined in (17) tends to produce a better quality.

\[
R = \{2 \leq x_j \leq 4, j = 1, 2, \ldots, 6\}. \tag{17}
\]

We also assume that the quality variable \( y \) only takes one (good) or zero (bad). Now, for one simulation run we generate a data set having 1500 observations \( \{(y_i, x_{i1}, x_{i2}, x_{i3}, x_{i4}, x_{i5}, x_{i6}) \mid i = 1, 2, \ldots, 1500\} \) among which, only 75 (5%) observations are made be in the box \( R \) through Steps 1–5.

**Step 1.** To generate observations in the box \( R \), make a discrete probability distribution of each variable \( x_j, (j = 1, 2, \ldots, 6) \) such as

\[
P(X = x) = \frac{u_1}{u_1 + u_2 + u_3}, \quad \frac{u_2}{u_1 + u_2 + u_3}, \quad \frac{u_3}{u_1 + u_2 + u_3}
\]

where \( u_1, u_2 \) and \( u_3 \) are random values taken independently from the uniform distribution ranging from zero to one.

<table>
<thead>
<tr>
<th>Learning set</th>
<th>Test set</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Box</strong></td>
<td>( \beta_B )</td>
</tr>
<tr>
<td>( B_1 )</td>
<td>0.210</td>
</tr>
<tr>
<td>( B_2 )</td>
<td>0.210</td>
</tr>
<tr>
<td>( B_3 )</td>
<td>0.213</td>
</tr>
<tr>
<td>( \ldots )</td>
<td>( \ldots )</td>
</tr>
<tr>
<td>( B_{29} )</td>
<td>0.632</td>
</tr>
<tr>
<td>( B_{30} )</td>
<td>0.680</td>
</tr>
<tr>
<td>( B_{31} )</td>
<td>0.769</td>
</tr>
<tr>
<td>( B_{32} )</td>
<td>0.774</td>
</tr>
<tr>
<td>( B_{33} )</td>
<td>0.807</td>
</tr>
<tr>
<td>( B_{34} )</td>
<td>0.849</td>
</tr>
<tr>
<td>( B_{35} )</td>
<td>0.878</td>
</tr>
<tr>
<td>( B_{36} )</td>
<td>0.907</td>
</tr>
<tr>
<td>( B_{37} )</td>
<td>0.929</td>
</tr>
</tbody>
</table>
Step 2. Generate 75 observations of \( x = (x_1, x_2, x_3, x_4, x_5, x_6) \), where each variable is generated independently from its own distribution made in Step 1.

Step 3. To generate observations not in the box \( R \), similarly, create again a discrete probability distribution of each variable \( x_j, (j = 1, 2, \ldots, 6) \),

\( (x, P(X = x)) \) for \( x = 1, 2, 3, 4, 5 \) and generate observations of \( x = (x_1, x_2, x_3, x_4, x_5, x_6) \) from the distributions until 1425 observations not in the box \( R \) are obtained.

Step 4. Generate observations of \( y \) corresponding to observations of \( x \) as follows:

If \( x \in R \), then \( y = 1 \) with probability \( 1 - \sigma \) and \( y = 0 \), otherwise.

If \( x \notin R \), then \( y = 0 \) with probability \( 1 - \sigma \) and \( y = 1 \), otherwise.

Step 5. Shuffle 1500 observations \( \{(y_i, x_{i1}, x_{i2}, x_{i3}, x_{i4}, x_{i5}, x_{i6}), i = 1, 2, \ldots, 1500\} \).

In Step 4, \( \sigma \) is a simulation factor representing noise, included in the quality variable. Four levels of \( \sigma \) were considered, such as 0, 0.1, 0.2, and 0.3. We conducted 5000 repetitions of the above simulation run for each of these four levels of noise.

5.2. Performance measure

Let \( B_p \) and \( B_i \) be the final boxes derived from PRIM and \( f \)-PRIM, respectively. To evaluate which of \( B_p \) and \( B_i \) is closer to \( R \), we compare the number of observations contained in \( R \) of (17) that are also included in \( B_p \) (\( B_i \)) and the number of observations residing outside \( R \) that are also excluded from \( B_p \) (\( B_i \)). To this end, we adopt the confusion matrix as shown in Table 4.

After calculating the number of \( a, b, c, \) and \( d \) in Table 4, we first calculate the sensitivity; the proportion of observations in \( R \) detected by \( B_p \) (\( B_i \)), by

\[
\text{Sensitivity} = \frac{a}{a + b}
\]  

and calculate the specificity; the proportion of observations not in \( R \) undetected by \( B_p \) (\( B_i \)), by

\[
\text{Specificity} = \frac{d}{c + d}.
\]

Then, we define \( G \) measure, the geometric mean of sensitivity and specificity, (Chong & Jun, 2005b; Kubat, Holte, & Matwin, 1998) as

\[
G = (\text{Sensitivity} \times \text{Specificity})^{1/2}.
\]

The value of \( G \) ranges between zero and one. The value close to one implies that most observations are classified correctly.

One would like to use the accuracy defined in (21) to evaluate performance.

\[
\text{Accuracy} = \frac{(a + d)}{(a + b + c + d)}.
\]

However, this measure is not suitable here due to an imbalance of observations in \( R \) and not in \( R \). If an algorithm always provides a large box including all observations, then its accuracy would be 95%, which is certainly a high value.

5.3. Results

In each simulation run, we split a data set into learning (900 observations) and test set (600 observations) randomly, and then applied PRIM and \( f \)-PRIM to the same learning set, respectively. In both algorithms, a stopping parameter \( \beta \) was set to 0.05. When applying PRIM, we tried 0.1, 0.2, 0.3, 0.4, and 0.5 for \( \sigma \), hence, five sequences of boxes were obtained. Similarly, in \( f \)-PRIM we made eleven sequences of boxes by trying different values for \( \lambda \) from zero to one by 0.1 increment with \( \lambda \) fixed to 0.1. The test set was used to recalculate box objectives and supports of all boxes involved in the sequences. In both algorithms, we chose the best one as the optimal, which has the maximum

Table 2
Box objectives according to different combinations of \( \lambda \) and \( x \)

<table>
<thead>
<tr>
<th>( x )</th>
<th>( \lambda )</th>
<th>0</th>
<th>0.1</th>
<th>0.2</th>
<th>0.3</th>
<th>0.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.05</td>
<td></td>
<td>0.486</td>
<td>0.486</td>
<td>0.517</td>
<td>0.597</td>
<td>0.615</td>
</tr>
<tr>
<td>0.1</td>
<td></td>
<td>0.508</td>
<td>0.508</td>
<td>0.465</td>
<td>0.507</td>
<td>0.588</td>
</tr>
<tr>
<td>0.15</td>
<td></td>
<td>0.458</td>
<td>0.508</td>
<td>0.508</td>
<td>0.610</td>
<td>0.586</td>
</tr>
</tbody>
</table>

Table 3
Statistics of the boxes found by the PRIM

<table>
<thead>
<tr>
<th></th>
<th>( x = 0.1 )</th>
<th>( x = 0.2 )</th>
<th>( x = 0.3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Box objective</td>
<td>0.244</td>
<td>0.522</td>
<td>0.493</td>
</tr>
<tr>
<td>Support</td>
<td>0.49</td>
<td>0.12</td>
<td>0.12</td>
</tr>
</tbody>
</table>

Table 4
Confusion matrix of simulated observations

<table>
<thead>
<tr>
<th></th>
<th>Actual Obs. inside ( R )</th>
<th>Actual Obs. outside ( R )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>a: the number of observations inside ( R ) classified correctly</td>
<td>b: the number of observations inside ( R ) classified incorrectly</td>
</tr>
<tr>
<td></td>
<td>c: the number of observations outside ( R ) classified correctly</td>
<td>d: the number of observations outside ( R ) classified correctly</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Obs. inside ( R )</th>
<th>Obs. outside ( R )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Obs. inside ( R )</td>
<td>Obs. outside ( R )</td>
</tr>
<tr>
<td></td>
<td>Obs. inside ( R )</td>
<td>Obs. outside ( R )</td>
</tr>
<tr>
<td></td>
<td>Obs. inside ( R )</td>
<td>Obs. inside ( R )</td>
</tr>
</tbody>
</table>
box objective among boxes whose support is greater than 0.05. At each run, we calculate $G$ measure as a criterion. Table 5 shows the percentage frequency of $\alpha$ chosen in PRIM during the total 20,000 ($=4 \times 5000$) runs. As seen in Table 5, the most frequent value for $\alpha$ was 0.4, which is too large for a patient strategy. This example indicates that PRIM may fail to exploit its patient strategy.

Table 6 summarizes the performance of PRIM and $f$-PRIM, using the average $G$ over 5000 runs along the noise. The bold figures denote the superior ones. $f$-PRIM outperforms PRIM especially when a larger noise is involved in data and shows a higher performance than PRIM by 12.2% ($=0.953-0.850)/0.850$) on average.

For additional information, we compare the box objective, which is the proportion of ones within the final box and shows a higher performance than PRIM by 12.2% ($=0.953-0.850)/0.850$) on average.

For additional information, we compare the box objective, which is the proportion of ones within the final box and shows a higher performance than PRIM by 12.2% ($=0.953-0.850)/0.850$) on average.

6. Conclusions

We propose a new algorithm, motivated by PRIM, which directly determines the optimal conditions of process variables from historical data. The proposed method is specially developed to deal with ordinal discrete variables unlike the original PRIM. For illustrative purposes, we applied the proposed method to a real steel-making process and found a solution expecting to increase the ratio of premium products by 2.6 times. Through an extensive simulation using artificial data sets, we also show that the proposed method performs better than the PRIM particularly when a larger noise is involved in data. In conclusion, we think that the proposed method is a good alternative method to process optimization when process variables under interest are in discrete type and much noisy information is contained in the data.

Acknowledgement

This work was partially supported by Korea Science and Engineering Foundation through National Core Research Center for Systems Bio-Dynamics at POSTECH.

References

CD-ROM.