The Irrelevant Values Problem of Decision Tree for Improving a Glass Sputtering Process

Ding-An Chiang[1]*, Cheng-Tzu Wang[2], Yi-Hsin Wang[3] and Chun-Chi Chen[1]

[1]*Department of Computer Science and Information Engineering, Tamkang University, Tamsui, Taiwan 251, R.O.C.*
[2]*Department of Computer Science, National Taipei University of Education, Taipei, Taiwan 106, R.O.C.*
[3]*Department of Information Management, Chang Gung Institute of Technology, Taiwan, R.O.C*

Abstract

In this paper, we use decision tree to establish a yield improvement model for glass sputtering process; however, the tree may have irrelevant values problem. In other words, when the tree is represented by a set of rules, not only comprehensibility of the resultant rules will be detracted but also critical factors of the manufacturing process cannot be effectively identified. From the performance issue and practical issue, we have to remove irrelevant conditions from the rules; otherwise, a domain expert is needed to review the decision tree. In this paper, we use a very simple example to demonstrate this point of view. Moreover, to identify and remove irrelevant conditions from the rules, we also revise Chiang's previous algorithm such that the modified algorithm can deal not only discrete data but also quantitative data.

***Key Words*:** Data Mining, Decision Tree, The Irrelevant Values Problem, Glass Sputtering Process, Yield Analysis

1. Introduction

In the modern production environment of the opto-electronics industry, due to the automation of machines and complicated production processes, any negligence may lead to product defects. These complicated processes involve mechanical equipment, configuration of processing parameters, and the production environment. These factors are ultimately critical to product quality. How to effectively manufacture high-quality optoelectronic products will be a great challenge for the opto-electronics industry. In other words, under intensive competitions, how to reinforce the process control ability and produce stable and high quality products will be one of the key factors for optoelectronics operators to lead other competitors. Therefore, in recent years, many researchers have adopted different data mining techniques to improve the yield of manufacturing process [1–6]. Besse & Legall [1] presented change detection methods to pin-point the defective stage within a manufacturing process when the existence of a failure was only known at the end of the process relying on the MCMC method. Chien et al. [2] included k-means clustering and a decision tree to infer possible causes of faults and manufacturing process variations from the semiconductor manufacturing data. Gardner & Bieker [3] investigated several data mining algorithms including Decision Trees, Bayesian Networks, Neural Networks, and Genetic Algorithms for solving semiconductor manufacturing problems. Wang et al. [4] proposed a hybrid method composed of partitioning clustering and hierarchical clustering to identify composite defect patterns on WBM. Last et al. [5] present a novel process optimization methodology based on the Data Mining approach. The operating model relating the process quality (output variables) to controllable (manipulated) variables is con-

*Corresponding author. E-mail: chiang@cs.tku.edu.tw structed from past operational data using single-target and multi-target Information Network (IN) algorithms for induction of oblivious decision-trees. Peng et al. [6] proposed a semiconductor manufacturing data mining framework in which Self Organizing Map (SOM) and Decision Tree were employed to extract empirical rules for controlling WIP levels given various product mixes and production conditions.

The decision tree is one of the key data mining techniques that it has been used to yield improvement of different manufacturing processes. In the previous literatures about decision tree, a regression tree is usually used with the lot-based data as the data source and the lot yield is considered as the target variable [1,2,7]. However, in recent years, the production strategy for small-quantity and diverse products is prevalent in the optoelectronics industry. In such production environment, yield variation is changeable and the amount of data is limited; consequently, analyzing with regression tree, a smaller lot quantity may lead to the deviation of the overall analysis result due to the difference in lot quantity. To avoid this problem, we transform each lot-based record by its quantity into glass-based records and use classification tree to create a yield predication model to improve the yield of glass sputtering process. Moreover, the problem of using decision tree to build a yield improvement model is that the irrelevant values problem may occur in the tree. In yield improvement model such problem will cause that critical factors to the glass sputtering process may not be effectively identified. Therefore, for the performance issue and practical issue, we have to remove irrelevant conditions from the rules.

Until now, a number of different algorithms have been proposed to solve the irrelevant values problem [8−11]. For example, Fayyad has proposed two algorithms. GID3 and GID3*, to solve the irrelevant values problem of the decision tree constructed using ID3. However, the problem of these algorithms is that some branches in the GID3 and GID3* tree may be longer than that in ID3 tree. In other words, comprehensibility of the tree will be detracted. As indicated by [12], another way to simplify decision tree structure is to translate the tree structure into a set of rules. Accordingly, Chiang provided a method to solve the irrelevant values problem of the decision tree without the problem of GID3 and GID3* tree [13]. It eliminates irrelevant values in the process of converting the decision tree to a set of rules by the information on the tree with respect to discrete values. However, the data of glass sputtering process always contains quantitative data; therefore, we have to modify Chiang's previous work such that the revised algorithm can deal not only discrete data but also quantitative data. The modified algorithm is presented in section 3.

In this paper, when decision tree is used to build a model to improve the glass sputtering process, we have defined a threshold value in advance to find out the positive and negative rules. When the yield of a rule is higher than the threshold value, it is set as a positive rule; on the contrary, if it is lower than the threshold value, it is set as a negative rule. In this study, we use the positive rules to adjust the process parameters into optimal range to improve the yield of sputtering process, and the negative rules to monitor the manufacturing process to avoid to causing any yield losses. Since we do not integrate threshold value into the decision tree, the original results of the decision tree cannot be used directly to find out the positive rules and negative rules. Therefore, we have to convert the original classification result of each leaf node to a new class by the predefined threshold value before of using our modified algorithm to remove irrelevant conditions from the rules. In this paper, we will use a simple example to show this converting process and demonstrate that our approach can get more concise and comprehensible rules than the rules which are obtained from the original tree directly.

2. Background Knowledge

2.1 Introduction of Glass Sputtering Processes

Tainan Science-based Industrial Park is the main cluster of Taiwan optoelectronic companies. Despite its proactive engagement in creating a cluster of optoelectronic plants, it also expects to induce a cluster effect, by introducing manufacturers of glass, backlit panel, polarizing film, fluorescent lamp, driver IC, and other important components, to enhance the overall performance of the optoelectronic supply chain. This study will focus on a glass coating plant in this science-based industrial park and analyze the production data of glass sputtering.

The glass sputtering coater adopted in this study is composed of 13 production modules, as shown in Figure 1. In this case study, target materials are placed in M6 and M10 modules. M6 provides Ti and ITO target materials, so Ti and ITO coating can be processed, respectively. M10 provides only SiO2 target material. This module does not have the processing function. The sub-

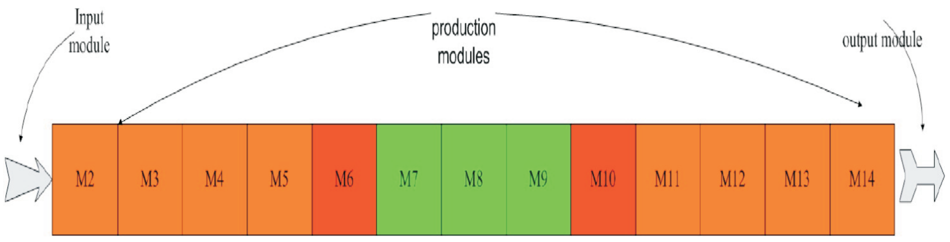

Figure 1. Illustration of the sputtering coater adopted in this study.

strates are processed from left to right through the M6 and M10 modules to coat a 2-layer thin film. M2, M3, M4, M5, M7, M8, M9, M11, M12, M13, and M14 are buffers and do not have the processing function. In the entire operation, they are in either venting or pumping status.

2.2 The Irrelevant Values Problem of Decision Tree

Decision trees can be categorized by data processing functions into classification tree and regression tree. A classification tree is applied to discrete variables, while a regression tree is applied to quantitative variables. Regression tree was first brought up by Breiman [14] in the introduction of CART. Usually, in CART analysis, data are categorized into quantitative and discrete data. Quantitative data can be applied to prediction, while discrete data can be applied to classification. In other words, CART is able to simultaneously process quantitative and discrete data. This is similar to C4.5, it is an extension of the ID3 algorithm developed by Quinlan to improve ID3 inability of processing quantitative values [11,15]. In the article, we use CART tree to analyze the collected data. Although different tree-induction algorithms have been proposed by different authors, without losing generality, we only consider ID3-like algorithm in this paper.

A decision tree is built up by selecting the best test attribute as the root of the decision tree. Then, the same procedure is operated on each branch to induce the remaining levels of the decision tree until all examples in a leaf belong to the same class. However, when an attribute is selected for branching out a node, both sub trees create a branch for some values of that appearing in the training data. Since some values of that attribute may not be relevant to the classification, the resultant rules of the decision tree may have irrelevant conditions, which demands irrelevant information to be supplied. Actually, according to the semantics of the irrelevant value, when a branch value is an irrelevant value of a rule; this value can be deleted or replaced by any value from the same domain value without affecting the correctness of the rule. Moreover, for the decision tree, not all attributes will be the nodes of some branches in the decision tree; therefore, when the corresponding values of attributes are missing in a branch, we say these attributes are also irrelevant attributes with respect to the branch. We explain this observation by the following definition.

Let A = {A1, ..., An} be a set of attributes, C = {C1, ..., Cs} be a set of classes and the branch Br of the decision tree can be represented as the form Br[A1] $\wedge$ ... $\wedge$ Br[An] $\rightarrow$ Ck, where Br[Ai] is a set of branch values out of an attribute Ai in the branch Br, i = 1 … n and $1 \le k \le s$.

Definition 1. Let Br[A1] $\wedge$ ... $\wedge$ Br[Aj-1] $\rightarrow$ Ck be a branch of the decision tree. Then, the rules with respect to attributes A1, ..., An implied by Br are:
{Br[A1] $\wedge$ ... $\wedge$ Br[Aj-1] $\wedge$aj$\wedge$ ... $\wedge$ans $\rightarrow$ Ck | ajr, ..., ans $\in$ domain(Aj, ..., An)},
where domain(Aj, ..., An) = domain(Aj) $\times$ … $\times$ domain (An).

Since some rules may be duplicated after eliminating irrelevant values, the number of the resultant rules may be less than that of leaves of the tree. The rule without irrelevant conditions is useful in many applications. For example, the patient can examine item B first to make sure whether or not this patient need to take item A. This process can reduce some burdens (expense, convenience or harmful) to the patient.

3. An Algorithm to Identify the Irrelevant Values

We introduce some important definitions and theorems of Chiang's work with respect to discrete data in section 3.1 [13]. The modified algorithm with respect to the discrete data and quantitative data is given in section 3.2.

3.1 Chiang's Previous Work Review

In this section, we introduce a definition and some theorems of Chiang's work with respect to discrete data. For easy explanation how to identify irrelevant values of a branch in a decision tree, Chiang defines the following definition.

Definition 2. Let Br and Br′ be two different branches of a decision tree, where Br = Br[A1] ∧ ... ∧ Br[Aj] → Ck1. Then Br is in conflict with Br′ with respect to attributes A1 … Aj if and only if Br[A1] ∧ ... ∧ Br[Aj] → Ck2 is a part of rule implied by Br′ and Ck1 ≠ Ck2.

To enable users to focus on only relevant conditions of the rules, Chiang provided the following theorems to solve the irrelevant values problem for a decision tree with discrete data. These theorems eliminate irrelevant values in the process of converting the decision tree to the rules according to information on the decision tree. These theorems are proven in [13], readers are suggested to refer to this paper, if further explanation is needed.

Theorem 1. Let Br[A1] ∧ ... ∧ Br[Aj] ∧ Br[Aj′] ∧ ... ∧ Br[An1] → Ck1 and Br′[A1] ∧ ... ∧ Br′[Aj] ∧ ... ∧ Br′[An2] → Ck2 be two branches through a non-leaf node P in the tree, where the branching attribute with respect to P is Aj. Let A = {Aj′, ..., An1} and A1 be the same attributes in these two branches, where $A_1 \subseteq A$. Then, Br is in conflict with Br′ with respect to A if and only if Br[A1] = Br′ [A1] and Ck1 ≠ Ck2.

Theorem 2. Let Br[A1] ∧ ... ∧ Br[Aj-1] ∧ Br[Aj] ∧ Br[Aj+1] ∧ ... ∧ Br[An] → Ck be a branch through a non-leaf node P in a decision tree, and the branching attribute with respect to P be Aj. For all branches through P of the decision tree, if Br is not in conflict with these branches with respect to attributes Aj+1 ... An, then Br[Aj] is an irrelevant value in Br.

Theorem 3. Let Br be a branch through a non-leaf node P of the decision tree. When the branch value Br[P] has been identified by theorem 2, all other branches through P are useless for the following process to identify the irrelevant values of Br.

To identify all the irrelevant values of a branch, Chiang's algorithm need to check all the branches in the decision tree only once. Moreover, since the algorithm do not have to consider the rules implied by each branch in the decision tree by theorem 3, the computation time of identifying whether two branches are in conflict with each other can be reduced greatly. Actually, without losing generality, since the number of common nodes of two branches is always small, the time complexity of identifying whether two branches are in conflict with each other can be seen as a constant. Therefore, the time complexity of identifying all irrelevant values of a branch by these theorems is reduced to O(m) at worst case, where m is the number of branches of the tree.

3.2 A Revised Algorithm

For many applications, attributes may contain quantitative data; to deal with quantitative values, the theorem 1 is revised into theorem 4 in this section. The modified algorithm can deal with not only discrete values but also quantitative values.

Theorem 4. Let Br[A1] ∧ ... ∧ Br[Aj] ∧ Br[Aj′] ∧ ... ∧ Br[An1] → Ck1 and Br′[A1] ∧ ... ∧ Br′[Aj] ∧ ... ∧ Br′[An2] → Ck2 be two branches through a non-leaf node P of the tree, where the branching attribute with respect to P is Aj. Let A = {Aj', ..., An1}, A1 be the same attributes in these two branches and a1 be a branch's value of Br[A1], where A1 ⊆ A. Then, Br is in conflict with Br' with respect to A if and only if

(1) when A1 ≠ ∅, ∃ a1, a1 ∈ Br[A1], a1 ∈ Br'[A1] and Ck1 ≠ Ck2, or
(2) when A1 = ∅, Ck1 ≠ Ck2.

Proof. Let A = {Aj', ..., An1} and A1 be the same attributes in these two branches

(1) Let A1 ≠ ∅, a1 be the branch's values, and a1 ∈ Br[A1]. When ∀ a1, a1 ∉ Br'[A1], it implies that these two branches will never be in conflict with each other with respect to A1 by theorem 1. Therefore, we need only to consider the case ∃ a1, a1 ∈ Br[A1], a1 ∈ Br'[A1] and Ck1 ≠ Ck2. When a1 ∈ Br'[A1], a1 → Ck2 must be a part of rule implied by Br'. Therefore, Br must be in conflict with Br' with respect to A if and only if ∃ a1, a1 ∈ Br[A1], a1 ∈ Br'[A1] and Ck1 ≠ Ck2.
(2) Let A1 = ∅ and Ck1 ≠ Ck2. When A1 = ∅, it implies that ∀ a, a → Ck2 must be a part of rule implied by Br', where a ∈ Br[A]. Since Ck1 ≠ Ck2, Br must be in conflict with Br' with respect to A.

According to theorem 4, when Ck1 = Ck2, branches, Br and Br' are never in conflict with each other; therefore, to identify all the irrelevant values of a branch Br, we need only to consider those branches, Br', whose leaves are different from Ck1. The corresponding algorithm is shown as Figure 2.

4. Experiments

In order to verify the proposed methodology, this study adopts the decision tree in IBM DB2 Intelligent Miner for Data to analyze the sputtering process data. In this research, the studied plant is an optoelectronic component plant in Tainan Science-based Industrial Park.

4.1 Mining Goal

The first step of this research is to set the mining goal. Yield is an important index to represent the optoelectronic companies' value. A high yield indicates high competitiveness, high production ability, and high quality. It directly affects production cost, so it is a basic tool for measuring production performance. Therefore, the mining goal of this study is to build a yield improvement model. We will investigate glass products with multilayer coatings and aim to improve the yield of first thin-film sputtering process. Although defects of sputtering process are caused by process control or human errors, in this study, the yield improvement goal is set to improve defects which are caused by process control.

In the model, we use decision trees to analyze process data to derive a set of valuable positive and negative decision rules. Since we hope that the yield values of the positive rules are as high as possible, the threshold value is defined as 95% in this study. Although the ranges of all the process control variables are identified by the different classification trees, in this paper, only a portion of results are covered due to the Non-Disclosure Agreement.

4.2 Data Preprocessing Step

With the assistance of experts in this field, as shown in Table 1, related factors of glass sputter coating were categorized into process machine data and glass lot data.

```
Input: A decision tree
Output: A set of rules without irrelevant conditions;
Let Br = {$Br_1, \ldots, Br_m$}; /* the branches of the decision tree */
For each branch Br' in Br Do
{
    Let Br' = $Br'[A_1] \wedge \ldots \wedge Br'[A_k] \rightarrow C_l$
    For j = k-1 downto 1 Do
    {
        Let be $A_j$ a node in Br';
        For all branches Br through $A_j$ and not through $A_{j+1}$ Do /*By theorem 2 and 3*/
        {
            Let Br = $Br[A_1] \wedge \ldots \wedge Br[A_j] \wedge \ldots \wedge Br'[A_n] \rightarrow C_{l'}$
            If $C_l \neq C_{l'}$ Then
                check whether Br is in conflict with Br' by theorem 4;
        }
        If Br' is not in conflict with any Br Then /*By theorem 2*/
            $Br'[A_j]$ is an irrelevant value and removed from Br';
    }
    Represented Br' by a rule;
}
```

Figure 2. Converting a decision tree to a set of rules without irrelevant conditions.

Since the studied plant mainly focuses on small-quantity and diverse products, the collected data with respect to a specific product is only composed of 1718 pieces of glass in 24 lots. The process machines data indicates the parameters on the glass sputter coating for each lot. The glass lot data records the related factors of lot yield after the glass sputter coating. Parameters that may affect glass sputtering are listed in Tables 2 and 3, where in Table 2, the yield of the i th lot is defined as follows:

$$Y_i = PO_i / PI_i \quad (1)$$

where,
Yi: The yield of i th lot
PIi: The input quantity of the i th lot
POi: The output quantity of the i th lot

In the data preprocessing step, we will process missing values, noisy data, and inconsistent data first. In addition, data directly collected from the system, as shown in Table 2 and Table 3, are not applicable to the decision tree analysis. Thus, we have to join Table 2 and Table 3 into one table in the data preprocessing step. The studied plant mainly focuses on small-quantity and diverse products; therefore, to avoid the problem that a smaller lot quantity may lead to the deviation of the overall analysis result, instead of using regression tree, we use classification tree to establish the yield improvement model in this study. Since target variable's data type of the classification tree has to be discrete and the data type of 'Yield' attribute is numerical type, we have to transform lot-based records into glass-based records by the input quantity and yield of each lot in the data preprocessing step. We

Table 1. Parameters of glass sputtering

Field type	Field name
Glass lot data	Lot number
	Input quantity
	Yield
	Input time
	Output time
	Work Time
Process machine data	Target 1 KWH
	Target 2 KWH
	Target 3 KWH
	Target 4 KWH
	Power 1 (±3%) kw (The 1st target gun of M6)
	Power 2 (±3%) kw (The 2nd target gun of M6)
	Power 3 (±3%) kw (The 1st target gun of M10)
	Power 4 (±3%) kw (The 2nd target gun of M10)
	Gasflow M4 (The gas flow voltage of M4 chamber)
	Gasflow M6 (The gas flow voltage of M6 chamber)
	Gasflow M8 (The gas flow voltage of M8 chamber)
	Gasflow M10 (The gas flow voltage of M10 chamber)
	Gasflow M12 (The gas flow voltage of M12 chamber)
	Speed (±1%) m/m (Trolley speed)

Table 2. Process production management information (1)

Lot No	Input time	Output time	Power 1	Power 2	Power 3	Power 4	Input quantity	Yield
510005	2005/10/21 06:26:30	2005/10/21 10:50:16	1.5	1.55	3.27	3.3	79	97.47%
510006	2005/10/20 19:23:21	2005/10/20 23:48:59	1.5	1.55	3.27	3	76	89.47%
510007	2005/10/20 21:22:01	2005/10/21 06:14:05	1.5	1.55	3.27	3.3	79	88.61%
510039	2005/10/31 16:43:26	2005/10/31 20:53:46	1.5	1.57	3.28	3.28	73	79.45%
510041	2005/10/31 15:31:47	2005/10/31 20:54:50	1.5	1.57	3.28	3.28	71	63.38%
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Table 3. Process production management information (2)

Lot No	Gas flow M4	Gas flow M6	Gas flow M8	Target 1 KWH	Target 2 KWH	Target 3 KWH	Target 4 KWH
510005	40	100	60/55	1366.3	1034.9	1757	1976.5
510006	40	100	60/55	1283.7	928.9	1567.8	1525
510007	40	100	60/55	1316.3	1014.9	1657	1876.5
510039	32	80	48/60	1558.4	1286.4	2163	2982.6
510041	32	80	48/60	1598.4	1346.4	2203	3012.6
⋮	⋮	⋮	⋮	⋮	⋮	⋮	

use the following example to explain how to transform lot-based records into glass-based records.

Example 1. Since the input quantity of lot '510005' has 79 pieces of glass, as shown in Table 2, this lot-based record is expended to 79 glass-based records, glass number from '55000501' to '55000579', in Table 4. The process parameters originally recorded by this lot-based record are also expanded into the corresponding glass-based records at the same time. That is, except the values of attribute 'yield flag' and 'Glass No', all other attributes' values of these 79 glass-based records are same with each other. Since the yield of this lot is 0.975, by formula (1), the yield flag's values of 77 glass-based records, glass number from '55000501' to '55000577', are marked by 'Y' and that of two other records, glass number '55000578' and '55000579', are marked by 'N', where the value 'Y' and 'N' indicate the glass has practically passed and not passed the process, respectively. In this table, the attribute 'yield flag' is considered as the target variable of our model. Through the above transformation, lot-based records are converted into glass-based records and the target variable's data type is changed from numerical to discrete. Therefore, instead of using regression tree, we can use classification tree to predict the yield and create a classification model for yield improvement.

4.3 Interpreting the Mining Results

Considering the derived decision tree presented in Figure 3, it shows the time factor with respect to the sputter coating of each glass. This tree can be represented by the following rules.

Rules:

(1) If 'Lot production completed' = 'Same Day'
and 'work time' < 333 seconds
Then 'yield flag' = 'Y' (96%)

(2) If 'Lot production completed' = 'Next Day'
Then 'yield flag'= 'Y' (75%)

(3) If 'Lot production completed' = 'Same Day'
and 'work time' >= 333 seconds
Then 'yield flag'= 'Y'(83.6%)

This first rule indicates that when the sputter coating process of a lot is completed on the same day and the

Table 4. Transformed data

Glass No	Lot No	Work time	Lot production completed	Input speed	Power 1	Power 2	Power 3	Power 4	Gas flow M4	Gas flow M6	Gas flow M8	Target 1 KWH	Target 2 KWH	Target 3 KWH	Target 4 KWH	Yield flag
51000501	510005	263	Same Day	3.34	1.5	1.55	3.27	3.3	40	100	60/55	1366.3	1034.9	1757	1976.5	Y
51000502	510005	263	Same Day	3.34	1.5	1.55	3.27	3.3	40	100	60/55	1366.3	1034.9	1757	1976.5	Y
51000503	510005	263	Same Day	3.34	1.5	1.55	3.27	3.3	40	100	60/55	1366.3	1034.9	1757	1976.5	Y
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
51000543	510005	263	Same Day	3.34	1.5	1.55	3.27	3.3	40	100	60/55	1366.3	1034.9	1757	1976.5	Y
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
51000577	510005	263	Same Day	3.34	1.5	1.55	3.27	3.3	40	100	60/55	1366.3	1034.9	1757	1976.5	Y
51000578	510005	263	Same Day	3.34	1.5	1.55	3.27	3.3	40	100	60/55	1366.3	1034.9	1757	1976.5	N
51000579	510005	263	Same Day	3.34	1.5	1.55	3.27	3.3	40	100	60/55	1366.3	1034.9	1757	1976.5	N
51000601	510006	263.4	Same Day	3.47	1.5	1.55	3.27	3	40	100	60/55	1283.7	928.9	1567.8	1525	Y
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

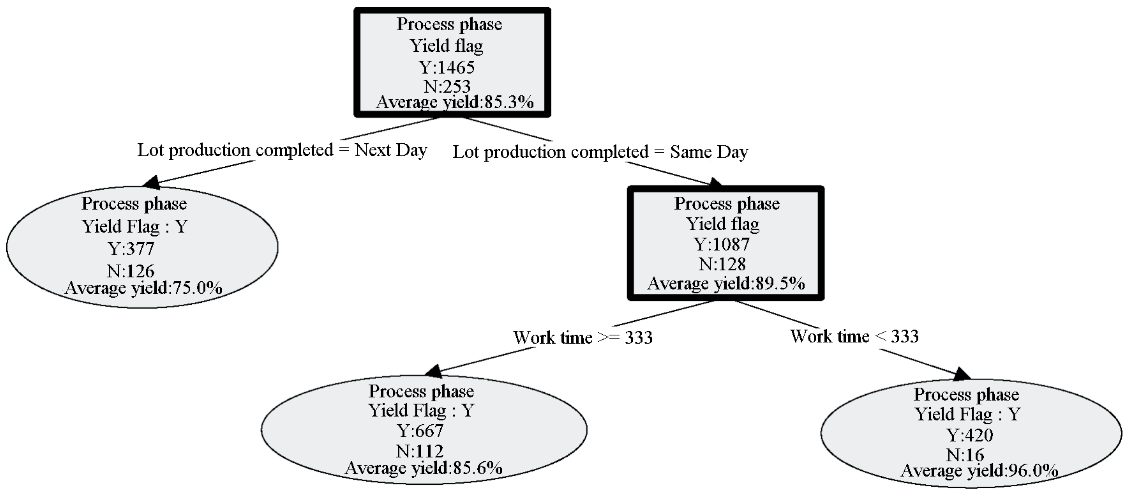

Figure 3. Classification tree yield analysis — work time.

sputtering time of each glass is less than 333 seconds, the predicted yield can reach to 96.0%. Moreover, since 6 glass substrates are manually placed on the trolley in the sputter coating process, this rule actually indicates that when the sputter coating process of a lot is completed on the same day and the sputtering time of each trolley is less than 1998 (333 × 6) seconds, the predicted yield can reach to 96.0%. The second rule indicates that if a lot is processed across 2 days, the yield of sputtering process is only 75%. In fact, if the sputtering process is delayed until the next day, contamination may occur. The last rule indicates that even the sputter coating of the lot is completed on the same day, if the sputtering time of each glass is more than 333 seconds; the yield of sputtering process is 83.6%.

4.4 The Effect of Irrelevant Values Problem in Deploying the Mining Results

Now, we explain how to deploy our mining results into our yield improvement system. Since the predicated yield of the first rule is higher than the predefined threshold value, this rule is a positive rule. On the contrary, the other two rules are negative rules. To improve the manufacturing process yield, we set the new range of each process parameter based on the positive decision rules so as to seek for the optimal yield process. Therefore, according to the positive rule, the studied plant orders that the sputter coating process of a lot has to be completed on the same day and tunes process parameters such that the sputtering time of each glass has to be done in 333 seconds. In other words, the sputtering time of each trolley has to be done no more than 1998 (333 × 6) seconds. Moreover, in our model, we use the negative rules to monitor the sputter coating process. When any condition meets one of the negative rules, the corresponding rule will be triggered to notify the engineer to adjust the corresponding control parameters in the process. Therefore, we hope that the negative rules are as short as possible so the rules can response the abnormal conditions as quickly as possible. For example, when time is over 24.00 and glasses are still on the trolley to coat, the second rule will be triggered to notify the engineer that the sputtering process a lot is processed across 2 days.

Now, let us consider the third rule in the above section. This negative rule is not useful because that the condition ‘Lot production completed = Same Day’ is in the rule. That is, when this rule is triggered to notify the engineer to adjust the corresponding control parameters in the process, the sputter coating process of this lot is already completed. Actually, in the following discussion, we could find out that the condition ‘Lot production completed = Same Day’ is an irrelevant condition in this negative rule. When the condition ‘Lot production completed = Same Day’ is removed from the rule, this rule is reduced to If ‘work time >= 333 seconds Then yield flag = Y (83.6%)’. It becomes a very useful negative rule. That is why when a decision tree is used to establish a yield improvement model, it is important that a domain expert is needed to review the decision tree carefully. In other words, not only for the performance issue but also for practical issue, the irrelevant conditions have to be identified in the tree or when the tree is represented by a set of rules, the irrelevant conditions have to be removed from the resultant rules. In the follows, we introduce how to use the modified algorithm to identify and remove the irrelevant conditions from the rules.

Since the classes of leaf nodes in the tree, as shown in Figure 3, are ‘Y’, we cannot identify any irrelevant condition in the rules by the modified algorithm. However, as shown in Figure 3, different leaf nodes have different yield values, which may present different meaning in our research; therefore, to find out and remove irrelevant conditions from the resultant rules, we have to convert the original class’s label of each leaf node to a new class’s label by the predefined threshold value and yield information of each leaf node. In this paper, when the yield value of a leaf node is greater than the predefined threshold value, the new value of this leaf node is set as ‘Po’; otherwise, it is set as ‘Ne’, where ‘Po’ and ‘Ne’ represent ‘Positive rule’ and ‘Negative rule’, respectively. Consequently, the original tree, as shown in Figure 3, can be converted to the new tree, which is shown in Figure 4.

Now, let us trace how the proposed algorithm works in the tree of Figure 4. Since the only irrelevant values condition is occurred in the branch Br2 of this tree, we only check whether the condition ‘Lot production completed = Same Day’ is an irrelevant condition in Br2. By theorem 3, we do not need to consider the branch Br3 in this computation. Since the leaf nodes’ values of branches Br1 and Br2 are same with each other, these two branches are never in conflict with each other by theorem 4. Therefore, we can conclude that the condition ‘Lot production completed’ = ‘Same Day’ is an irrelevant condition in Br2 by theorem2. Through our modified algorithm, this new tree can be represented by one positive decision rule and two negative decision rules as follows:

Positive rule:
    If ‘Lot production completed’ = ‘Same Day’
        and ‘work time’ < 333 seconds
      Then ‘new yield flag’= ‘Po’
Negative rules:
    If ‘Lot production completed’ = ‘Next Day’
      Then ‘new yield flag’ = ‘Ne’
    If ‘work time’ >= 333 seconds
      Then ‘new yield flag’= ‘Ne’

According to the above rules, we find out that the irrelevant condition ‘Lot production completed’= ‘Same Day’ has been removed from the second negative rule. Since only one condition need to be checked in this new negative rule, it is more sensitive than the original rule. Based on this new rule, when we find out that the sputtering time of each glass is more than 333 seconds or that of each trolley is more than 1998 seconds, this rule will be triggered to notify the engineer to adjust the corresponding control parameters in the process. Through the above discussion, we could find out that some critical factors of glass sputtering process can be effectively discovered by removing irrelevant conditions from the rules. After seven months on-line testing, the results showed that the average lot yield and lot yield standard deviation were improved from 70% and 11.1 to 95.8% and 2.5, respectively. In fact, if we discard the defeat glasses which are caused by human errors, we can get better results than 95.8% and 2.5, respectively.

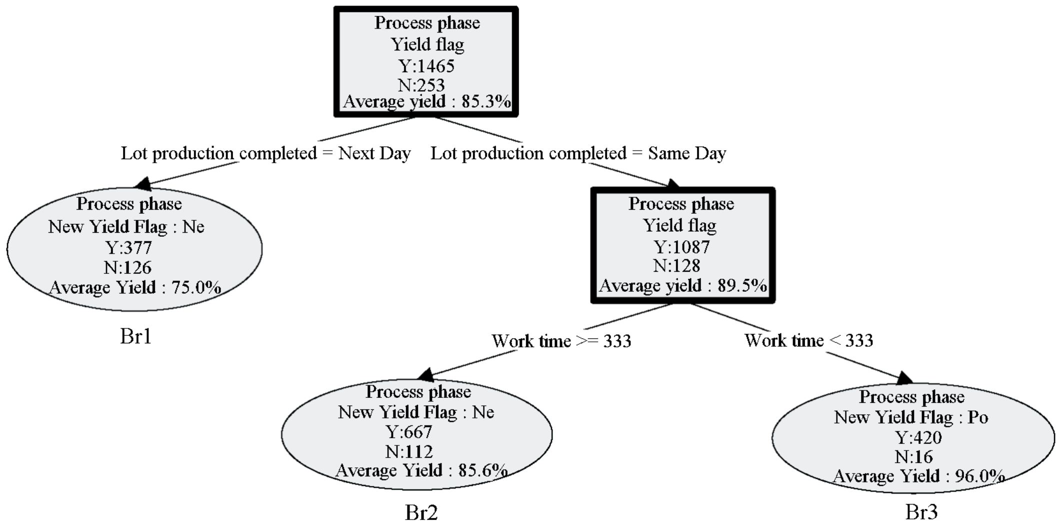

Figure 4. Converted to the new tree.

5. Conclusion

When a decision tree is used to build a yield improvement model, it is important that irrelevant conditions have to be removed; otherwise, critical factors to the yield of glass sputtering process may not be effectively discovered. In this paper, we use a very simple example to demonstrate this point of view. In the current study, we use only one threshold value to distinguish the positive rules and negative rules. The disadvantage of using only one threshold value is that too many negative rules are discovered and some of these rules may not cause abnormal phenomena. Therefore, we plan to use another threshold value to identify the negative rules in the near future to avoid generating too many negative rules.

As indicated by [12], one of methods to simplify decision tree structure is to translate the tree structure into a set of rules. In this paper, we eliminate irrelevant values in the process of converting the decision tree to a set of rules by the information on the tree. In the near future, we plan to use quantitative association rule mining approach to solve this irrelevant problem, that is, we use quantitative association rule mining approach to find out all possible association rules with respect to the tree. Moreover, as pointed out by [16], the use of sharp boundary intervals it is not intuitive with respect to human perception. Therefore, we also plan to introduce fuzziness into the model to remedy the sharp boundary problem.

References

[1] Besse, P. and Legall, C., "Application and Reliability of Change-Point Analyses for Detecting a Defective Stage in Integrated Circuit Manufacturing," *Communications in Statistics: Simulation and Computation*, Vol. 35, pp. 479–496 (2006).

[2] Chien, C. F., Wang, W. C. and Cheng, J. C., "Data Mining for Yield Enhancement in Semiconductor Manufacturing and an Empirical Study," *Expert Systems with Applications*, Vol. 33, pp. 192–198 (2007).

[3] Gardner, M. and Bieker, J., "Data Mining Solves Tough Semiconductor Manufacturing Problems," In *Proceedings*, KDD-2000, ACM, New York, NY, pp. 376–383, USA (2000).

[4] Kundu, B., White Jr, K. P. and Mastrangelo, C., "Defect Clustering and Classification for Semiconductor Devices," *MWSCAS-2002*, Vol. 2, pp. 561–564 (2002).

[5] Last, M., Danon, G., Biderman, S. and Miron, E., "Optimizing a Batch Manufacturing Process through Interpretable Data Mining Models," *Journal of Intelligent Manufacturing*, (2008).

[6] Peng, C. and Chien, C., "Data Value Development to Enhance Yield and Maintain Competitive Advantage for Semiconductor Manufacturing," *International Journal of Service Technology and Management*, Vol. 4, pp. 365–383 (2003).

[7] Braha, D. and Shmilovici, A., "Data Mining for Improving a Cleaning Process in the Semiconductor Industry," *IEEE Transaction Semiconductor Manufacturing*, Vol. 15 (2002).

[8] Cheng, J., Fayyad, U. M., Irani, K. B. and Qian, Z., "Improved Decision Trees: A Generalized Version of ID3," *Proc. of the Fifth Int. Conf. on Machine Learning*, pp. 100–108 (1988).

[9] Fayyad, U. M. and Irani, K. B., "A Machine Learning Algorithm (GID3*) for Automated Mated Knowledge Acquisition Improvements and Extensions," *General Motors Reseat Report CS-634*, Warren, MI: GM Research Labs (1991).

[10] Fayyad, U. M., "Branching on Attribute Values in Decision Tree Generalization," *Proc. Twelfth National Conference on Artificial Intelligence AAAI-94*, Seattle, Washington, pp. 104–110 (1994).

[11] Quinlan, J. R, "C4.5: Programs for Machine Learning," *Morgan Kaufmann Publishers Inc*., San Francisco, CA, USA (1993).

[12] Breslow, L. and Aha, D. W., "Simplifying Decision Trees: A Survey," *Navy Center for Applied Research in Knowledge Engineering Review Technique Report* (1998).

[13] Chiang, D. A., Chen, W., Wang, Y. F. and Hsu, C. F., "The Irrelevant Values Problem in the ID3 Tree," *Computers and Artificial Intelligence*, Vol. 10, pp. 169–182 (2000).

[14] Breiman, L., Friedman, J. H., Olshen, R. A. and Stone, C. J., "Classification and Regression Trees," *International Thomson Publishing* (1984).

[15] Quinlan, J. R., "Introduction to Decision Tree," *Machine Learning*, Vol. 1, pp. 81–106 (1986).

[16] Kaya, M. and Alhajj, R., "Genetic Algorithm Based Framework for Mining Fuzzy Association Rules," *Fuzzy set and system*, Vol. 152, pp. 587–601 (2005).

Manuscript Received: Oct. 31, 2008
Accepted: Apr. 30, 2009