Available online at www.sciencedirect.com
SCIENCE DIRECT
Computers in Industry 57 (2006) 412–423
COMPUTERS IN INDUSTRY
www.elsevier.com/locate/compind

A neural network based information granulation approach to shorten the cellular phone test process

Chao-Ton Su [a,*], Long-Sheng Chen [b], Tai-Lin Chiang [c]

[a] Department of Industrial Engineering and Engineering Management, National Tsing Hua University, Hsinchu 300, Taiwan
[b] Department of Industrial Engineering and Management, National Chiao Tung University, Hsinchu, Taiwan
[c] Department of Business Administration, Ming Hsin University of Science and Technology, Hsinchu, Taiwan

Received 10 March 2005; accepted 18 January 2006
Available online 29 March 2006

Abstract

In the cellular phone OEM/ODM industry, reducing test time and cost are crucial due to fierce competition, short product life cycle, and a low margin environment. Among the inspection processes, the radio frequency (RF) function test process requires more operation time than any other. Hence, manufacturers need an effective method to reduce the RF test items so that the inspection time can be reduced while maintaining the quality of the RF function test. However, traditional feature selection methods such as neural networks and genetic algorithm lead to a high level of Type II error in the situation of imbalanced data where the amount of good products is far greater than the defective products. In this study, we propose a neural network based information granulation approach to reduce the RF test items for the finished goods inspection process of a cellular phone. Implementation results show that the RF test items were significantly reduced, and that the inspection accuracy remains very close to that of the original testing process. In addition, the Type II errors decreased as well.

Keywords: Cellular phone inspection process; Feature selection; Information granulation; Fuzzy ART neural network; Imbalanced data

1. Introduction

Personal wireless communication services have been available to the general public for only about 10 years, since the breakthrough of cellular phones [12]. At the same time the technology employed by mobile telecommunications is evolving rapidly. New designs in cellular phones and novel functions are being introduced at an ever-increasing pace. This is leading to fierce competition and short product life cycles. Consequently, one of the major concerns of original equipment manufacture (OEM) and electronic manufacturing service (EMS) phone manufacturers is how to decrease testing costs [1], especially in the low margin environment in which they operate. This is because testing equipment for mobile phones is expensive, and the testing times are long. In one estimate, it costs around US$ 1 and 1–3 min per phone [28]. However, these testing costs and time will increase dramatically because more and more newly developed modules like digital camera, mp3 player, personal digital assistant (PDA), and blue-tooth transmitter are added to cellular phones. We have to spend extra time and money to inspect these new functions. These factors often hinder the enhancement of the overall output of cellular phones [28].

In the manufacturing process of cellular phones shown as Fig. 1, the radio frequency (RF) function is a crucial test and needs more operation time than any of the other inspection processes. In order to save inspection costs and shorten production time, manufacturers need an effective method to reduce the RF function test items. A number of soft computing approaches, such as neural networks [27], genetic algorithms (GA) [32], decision tree and rough sets [22,23] have been widely used to remove irrelevant, unnecessary, and redundant attributes (test items). However, when these methods are applied to real world problems, there are many issues that need to be addressed. One of them is the ''imbalanced data'' problem which almost all the instances are labeled as one class while far few instances are labeled as the other class [5,10]. When learning from such imbalanced data, traditional classifiers often produce high accuracy over the majority class, but poor predictive accuracy over the minority class (usually the important class).

* Corresponding author. Tel.: +886 35742936; fax: +886 35722204.
E-mail address: ctsu@mx.nthu.edu.tw (C.-T. Su).

0166-3615/$ – see front matter
doi:10.1016/j.compind.2006.01.001

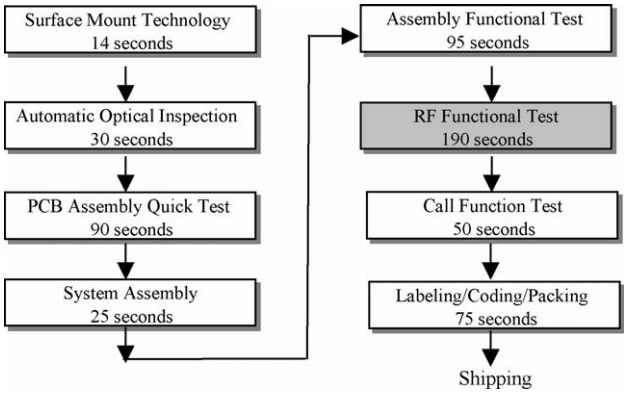

Fig. 1. A manufacturing process of a cellular phone.

In modern production systems, the defective rate of products is becoming quite low. In the six-sigma quality management system for example, we should use parts per million (''ppm'') instead of ''%'' to calculate the defective rate. In a mature manufacturing industry, the amount of good products far exceeds the defective products. This type of data is so-called ''imbalanced data.'' When feature selection approaches encounter imbalanced data such as this, it becomes difficult to acquire knowledge from the few negative examples (defective products). Fewer abnormal products will be viewed as outliers or bias by feature selection methods [17]. This leads to a high level of Type II errors (customer risks, the probability that customers accept defective products) which are critical to OEM/EMS companies. A high level of Type II errors will cause great losses, requires compensation and may result in the loss of orders from important customers.

In this study, we propose a neural network based information granulation approach which can effectively reduce RF function test items. A real case with imbalanced data is studied, and the implementation results show that our proposed method can find relevant test items without losing classification accuracy and increasing the Type II errors.

2. Feature selection from imbalanced data

Reduction of pattern dimensionality via feature selection belongs to the most fundamental steps in data processing [23]. A large feature set often contains redundant and irrelevant information, and can actually degrade the performance of the classifier [14]. The main purpose of feature selection is to remove irrelevant or redundant attributes and improve the performance of data mining.

Feature selection is often applied in pattern classification, data mining, as well as machine learning. Among many feature selection methods, GA, rough sets and neural networks have attracted much attention, and have become popular techniques for feature selection. However, when these methods are applied to imbalanced data, they usually suffer from some drawbacks, such as ignoring the minority examples and viewing them as outliers. It was reported [5,10] that use of these methods in seeking an accurate performance over a full range of instances is not suitable to deal with imbalanced learning tasks since they tend to classify all data into the majority class, which is usually the less important class. This is because typical classifiers are designed to optimize overall accuracy without taking into account the relative distribution of each class.

Rough sets emerged as a major mathematical tool for discovering knowledge and feature selection [29]. One of the fundamental principles of a rough set-based learning system is discovering redundancies and dependencies between the given features of a problem to be classified. A reduct generated by the rough sets approach is defined as the minimal subset of attributes that enables the same classification of objects with full attributes. When applying rough sets in practice, its computational complexity increases dramatically with the growth of the data. In addition, the deterministic mechanism for the description of error is very simple in rough sets. Therefore, the rules generated by rough sets are often unstable and have a low classification accuracy [13].

Feature selection with neural networks can be thought of as a special case of architectural pruning [21], where the input features are pruned rather than the hidden neurons. Su et al. [24] attempted to determine the important input nodes of a neural network based on the sum of absolute multiplication values of the weights between the layers. Unfortunately, the training of neural networks when using imbalanced data is very slow [6].

Another common understanding is that some learning algorithms have built-in feature selection, for example, ID3 [19], FRINGE and C4.5 [20]. Almuallim and Dietterich [3] suggested that one should not rely on ID3 or FRINGE to filter out irrelevant features. There are some cases in which ID3 and FRINGE miss extremely simple hypotheses. In addition, the negative examples of imbalanced data might be removed in the pruning phase of the tree construction.

In other words, when faced with imbalanced data, the performance of feature selection tools drops significantly [2]. Pendharkar et al. [17] mentioned that the ratio of the number of objects belonging to positive and negative examples impacts upon effective learning. If the data set contains many positive examples and very few negative examples, there is a bias in the discriminant function that the technique will identify, and it therefore follows that this bias results in a lower reliability of the technique. Application areas such as gene profiling, medical diagnosis and credit card fraud detection, oil spill detection, risk management, and medical diagnosis/monitoring [2,5,10,18] have highly skewed datasets with very small number of negative instances which are hard to classify correctly, but nevertheless are very important that they be detected.

An and Wang [4] suggested to balance the data by sampling. However, this is sometimes not feasible due to there being so few negative examples. The concept of information granulation may be the way to tackle problems caused by imbalanced data.

3. Information granulation

Information granulation, first pointed out by Zadeh [31], is turning out to be a very important issue for computer science, logic, philosophy, and others [30]. Information granulation is the process of forming meaningful pieces of information, called

information granules (IGs), that are regarded as entities embracing collections of individual elements (e.g. numerical data) that exhibit some functional or descriptive commonalities [9]. Information granulation emphasizes the fact that a plethora of details does not necessarily amount to knowledge. Granular computing, which is oriented towards representing and processing information granules, is a computing paradigm that embraces a number of modeling frameworks.

In many situations, when describing a problem we tend to shy away from numbers, and instead use aggregates to ponder the question. This is especially true when a problem involves incomplete, uncertain, or vague information. It may be difficult sometimes to differentiate distinct elements, and so one is forced to consider granules.

Most positive examples (good products) of production data are similar, duplicated, or redundant [26]. If we gather similar objects into information granules, then a large amount of data will transform into fewer granules. This way, we can reduce the ratio of positive to negative examples, and so possibly reduce the level of Type II errors.

4. Proposed methodology

In this section, a neural network based information granulation approach is proposed to construct information granules, and acquire knowledge from these granules.

4.1. Neural network based information granulation approach

Fig. 2 shows the basic idea of the proposed methodology. A large amount of similar objects are gathered together to form fewer granules. Information granulation can remove some unnecessarily detailed information, avoid an enormous quantity of knowledge rules being generated, and provides a better insight into the data. Moreover, when the information granulation approach is employed, numeric data will transfer to information granules and the number of positive and negative granules will be decreased compared with numeric data. The ratio of negative to positive examples will be increased. It may improve imbalanced data situation. Next, these granules are described with appropriate form and then we can use feature selection method to extract knowledge rules or key attributes from these granules. The detailed procedure of the neural network based information granulation approach is described as follows:

Step 1: Identify condition attributes and class attributes
Step 2: Data preprocessing
  Step 2.1: Data cleaning (fill in missing data and remove noisy or inconsistent data)
  Step 2.2: Data transformation (normalize or discretize the data)
Step 3: Measure the information granules
  Step 3.1: Select the degree of similarity
  Step 3.2: Check the suitability
  Step 3.3: Determine the suitable similarity

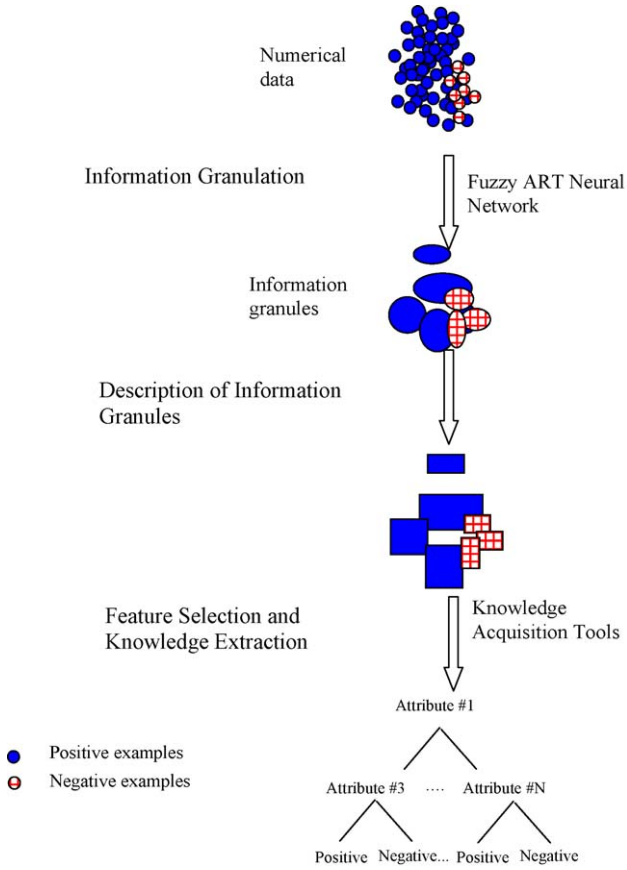

Fig. 2. Basic idea of the proposed methodology.

Step 4: Construct the information granules
Step 5: Define the information granules
  Step 5.1: Describe the information granules
  Step 5.2: Tackle the overlaps among the information granules
Step 6: Acquire key attributes and extract knowledge rules

Steps 1 and 2 are data preparing phases. In these phases, we should identify the condition attributes (inputs) and the decision attribute (output) first. Then, data should be prepared for the process, like removing noisy data, filling missing data, and discretizing data. In Step 3, the users need to determine suitable level of granularity. After that, the Fuzzy adaptive resonance theory (Fuzzy ART) [8] neural network can be utilized to construct the IG, depending on the selected similarity (granularity). Next, we describe these IGs using the appropriate form. Finally, the relevant attributes can be found by feature selection methods. A more detailed discussion of our proposed approach is given in the following subsections.

4.2. Data preprocessing

After identifying input and output variables (Step 1), data need to be preprocessed. Step 2 is to clean data and transform data. Real-world data tend to be incomplete, noisy, and inconsistent. Data cleaning attempt to fill in missed values, smooth out noise while identifying outliers, and correct

inconsistencies in the data. Discretization techniques of Step 2.2 can be used to reduce the number of values for a given continuous attribute, by dividing the range of the attribute into intervals. In this study, "equal frequency bining" approach is utilized to discretize data. This unsupervised method is to divide the range into b bins of equal frequency. This method is less susceptible to outliers, and the intervals would be closer to each other in regions where there are more elements and farther apart in sparsely populated regions, which represents the distribution of each variable better than the equal-width method. In summary, data preprocessing techniques can improve the quality of the data, thereby helping to improve the accuracy and efficiency of data mining process.

4.3. Measurement of information granules

If we want to extract knowledge from granules, the first question that needs to be answered is: how should similar objects be gathered to form granules? In other words, we must determine what kind of similarity the objects must have to form a granule.

In this section, we introduce two indexes, purity and centrality, to measure information granules. The purity expresses the uniqueness of the IG. There are two IGs, A and B, shown in Fig. 3. The purity of A is defined by Eq. (1):

$$\text{Purity}(A) = \frac{\sum_{X=1}^{P} N(A_X \cap \overline{B}_X)/N(A_X)}{P} \quad (1)$$

where $N(A \cap \overline{B})$ is the amount of objects in the intersection between A and $\overline{B}$; $N(A)$ the amount of objects in A; $\overline{B}$ the complementary set of B; X denotes the attributes, and P is the number of attributes.

In Fig. 3, we can clearly see that the higher the purity, the smaller the overlap between A and B is. If A and B are totally separated, $N(A \cap \overline{B})$ will be equal to $N(A)$. In this situation, purity is equal to 1.

Centrality is used to measure the 'within variation' in the IG. This index is defined by Eq. (2):

$$\text{Centrality} = \sum_{i=1}^{N}\sum_{j=1}^{M} \frac{(\max X_{ij} - \min X_{ij}/R_j)}{N} \quad (2)$$

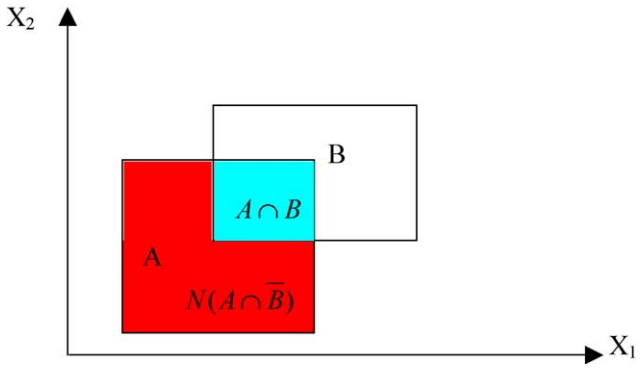

Fig. 3. The overlap between IG A and IG B.

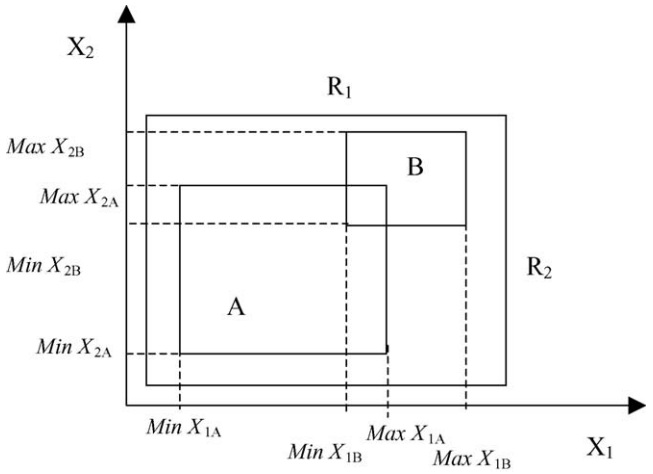

Fig. 4. The centrality of IG A and IG B.

where N is the number of information granules; M is the number of attributes; X denotes the attribute value; R_j is the range of the jth attribute value; max X_{ij} represents the upper limit of the jth attribute in the ith information granule, and min X_{ij} is the lower limit of the jth attribute in the ith information granule.

The more similar to each other the objects are in an IG the smaller the centrality of that IG will be. As Fig. 4 shows, if the upper limit (max X_{ij}) and the lower limit (min X_{ij}) are close, than this situation represents that the 'within variation' is small. Therefore, the centrality will be small.

4.4. Construction of information granules

This study suggests using Fuzzy ART to construct IGs. Fuzzy ART is not only a well-established neural network theory, but also a well known clustering method. Instead of clustering by a given number of clusters, it assigns patterns onto the same cluster by comparing their similarity. The major difference between Fuzzy ART and other unsupervised neural networks is the so-called vigilance parameter (ρ). The Fuzzy ART network allows the user to control the degree of similarity of patterns placed on the same cluster.

Two other similar types of architectures exist as well, namely ART 1 and ART 2. ART 1 is designed for binary-valued input patterns, and ART 2 is designed for continuous-valued patterns. Fuzzy ART provides a unified architecture for both binary and continuous valued inputs. In addition, Fuzzy ART possesses the same desirable stability properties as ART1 and a simpler architecture than that of ART2. With ART1, there is a serious dependency of the classification results on the sequence of input presentation and ART2 experiences difficulty in achieving good categorizations, if the input patterns are not all normalized to a constant length [7]. As a result, Fuzzy ART was utilized to construct information granules in this study.

Fuzzy ART has three parameters: (1) the choice parameter, $\alpha > 0$, which is suggested to be close to zero; (2) the learning parameter, β, which defines the degree to which the weight vector is updated with respect to an input vector, and (3) the vigilance parameter, ρ, which defines the required level of

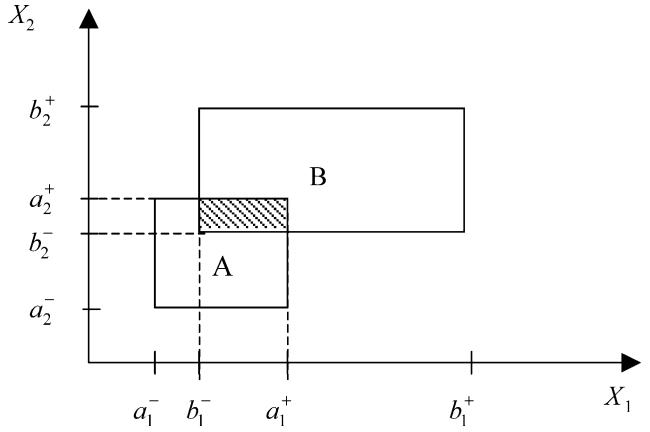

Fig. 5. The overlapping situation between IGs *A* and *B*.

similarity of patterns within clusters. The vigilance parameter is usually defined by the user.

4.5. Description of the information granules

Another issue when extracting knowledge from granules is how to describe IGs. In this study, we utilize hyperboxes to represent IGs [16]. A granule usually contains more than one object. We use the upper boundary and the low limit of the value of the attributes to represent whole objects within a granule.

The overlaps described in Fig. 5 always occur among IGs, and they are difficult to deal with by data mining algorithms which are not designed to deal with IGs, especially when an overlapping situation occurs. In this study, the concept of ''sub-attributes'' is utilized to tackle this problem, where we divided the original condition attributes into sub-attributes. By introducing the sub-attributes, we can easily extract key attributes or knowledge rules from these overlapping IGs.

Consider two IGs *A* and *B* which contain two original condition attributes, X_1 and X_2. In Table 1, IG $A(B)$ are fully described by its lower $a^-(b^-)$ and upper boundary $a^+(b^+)$, where $a^-(b^-)$ and $a^+(b^+)$ are vectors. More specific, we follow a full notation $[\mathrm{IG}A] = [a_i^-, a_i^+]$ and $[\mathrm{IG}B] = [b_i^-, b_i^+]$ to represent those two IGs, where i is the attribute index. We separate the overlapping and non-overlapping parts into independent intervals $[a_1^-, b_1^-]$, $[b_1^-, a_1^+]$ and $[a_1^+, b_1^+]$; $[a_2^-, b_2^-]$, $[b_2^-, a_2^+]$ and $[a_2^+, b_2^+]$ which are the so-called sub-attributes (labeled $X_{11}, X_{12}, X_{13}; X_{21}, X_{22}, X_{23}$). Then we utilize the Boolean variable, 0 or 1, to be the values of sub-attributes. If the value of a sub-attribute is ''0'', that means this sub-attribute does not contain an independent intervals such as $[a_1^-, b_1^-]$, etc. Table 2 lists the results of adding sub-attributes. By using the concept of sub-attributes, we can acquire knowledge from the IGs.

Table 1
IGs *A* and *B* described as a hyperbox form

	Attributes	
	X_1	X_2
IGs		
A	$[a_1^-, a_1^+]$	$[a_2^-, a_2^+]$
B	$[b_1^-, b_1^+]$	$[b_2^-, b_2^+]$

Table 2
Information granules with the addition of sub-attributes

	Original attributes					
	X_1			X_2		
	X_{11}[a] $[a_1^-, b_1^-]$	X_{12}[a] $[b_1^-, a_1^+]$	X_{13}[a] $[a_1^+, b_1^+]$	X_{21}[a] $[a_2^-, b_2^-]$	X_{22}[a] $[b_2^-, a_2^+]$	X_{23}[a] $[a_2^+, b_2^+]$
IGs						
A	1	1	0	1	1	0
B	0	1	1	0	1	1

[a] Sub-attributes.

4.6. Feature selection and knowledge extraction

In Step 6 of the proposed procedure, we employ decision tree, rough sets, and neural network based methods to acquire attributes and to extract knowledge rules. These three methods are briefly described in the following.

4.6.1. Decision tree

The decision tree method is one of the most popular knowledge acquisition algorithms, and has been successfully applied in many areas. Decision tree algorithms, such as ID3 and C4.5, were originally intended for classification purposes. The core of C4.5 contains recursive partitioning of the training examples. Whenever a node is added to a tree, some subsets of the input features are used to pick the logical test at that node. The feature that results in the maximum information gain is selected for testing at that node. In other words, the algorithm chooses the ''best'' attribute to partition the data into individual classes at each node. After the test has been determined, it is used to partition the examples, and the process is continued recursively until each subset contains examples of one class or satisfies some statistical criteria [25].

When decision tree induction is used for feature selection, a tree is constructed from the given data. All attributes that do not appear in the tree are assumed to be irrelevant. The set of attributes appearing in the tree form the reduced subset of attributes [11].

4.6.2. Rough sets

The rough sets theory was introduced by Pawlak [15] to deal with imprecise or vague concepts [23,29]. Rough sets deal with information represented by a table called the *information system* which contains objects and attributes. An information system is composed of a 4-tuple as follows:

$$S = \langle U, Q, V, f \rangle,$$

where U is the universe, a finite set of N objects $\{x_1, x_2, \ldots x_N\}$, Q is the finite set of attributes, $V = \cup_{q \in Q} V_q$, where V_q is the value of attribute q, and f: $U \times Q \rightarrow V$ is the total decision function called the *information function* such that $f(x, q) \in V_q$

for every $q \in Q$, $x \in U$. For a given subset of attributes $A \subseteq Q$ the IND(A)

$$\mathrm{IND}(A) = \{(x, y) \in U : \text{for all } a \in A,\ f(x, a) = f(y, a)\}$$

is an equivalence relation on universe U (called an indiscernibility relation).

Some of the information systems can be designed as a *decision table*

$$\text{Decision table} = \langle U, C \cup D, V, f \rangle$$

where C is the set of condition attributes, D is the set of decision attributes, $V = U_{q \in C \cup D} V_q$, where V_q is the set of values of attribute $q \in Q$, and f: $U \times (C \cup D) \rightarrow V$ is the total *decision function* (decision rule in a decision table) such that $f(x, q) \in V_q$ for every $q \in Q$ and $x \in V$.

For a given information system S, a given subset of attributes $A \subseteq Q$ determines the approximation space $AS = (U, \mathrm{IND}(A))$ in S. For a given $A \subseteq Q$ and $X \subseteq U$ (a concept of X), the *A-lower approximation* $\underline{A}X$ of set X in AS and *A-upper approximation* $\overline{A}X$ of set X in AS are defined as follows:

$$\underline{A}X = \{x \in U : [x]_A \subset X\} = \cup \{Y \in A^* : Y \subseteq X\},$$

$$\overline{A}X = \{x \in U : [x]_A \cap X \neq \varnothing\} = \cup \{Y \in A^* : Y \cap X \neq \varnothing\}$$

where A^* denotes the set of all equivalence classes of IND(A). The process of finding a set of attributes smaller than the original one with the same classificatory power as the original set is called *attribute reduction*. A *reduct* is the essential part of an information system (subset of attributes) which can discern all objects discernible by the original information system. By means of the dependent properties of the attributes we can find a reduced set of attributes, providing that by removing the superfluous attributes there is no loss in classification accuracy.

4.6.3. Feature selection from a trained neural network

Su et al. [24] proposed an algorithm to remove unimportant input nodes from a trained back-propagation neural network (BPNN). The essence of this method is to compare the multiplication values of the weights between the input-hidden layer and the hidden-output layer. Only the multiplication weights with large absolute values are kept and the rests are removed. The equation for calculating the sum of absolute multiplication values is defined as follows.

$$\mathrm{Node}_i = \sum_j \left| W_{ij} \times V_{jk} \right| \tag{3}$$

where W_{ij} is the weight between the ith input node and the jth hidden node, and V_{jk} is the weight between the jth hidden node and the kth output node. Then, we must set a threshold to remove the irrelevant input nodes. The threshold should be determined by the user to obtain a suitable number of input nodes.

5. Case study

The actual case comes from a cellular phone OEM/ODM company which was established in 1984. It is located in Taiwan and the company owns several factories in mainland China. In 2003, its total annual revenue reached US$ 4.713 billion, and it has a worldwide workforce of over 10,000. The production volume of cellular phones in 2004 was about 7.5 million units.

5.1. The problem

In this case, the objectives of the cellular phone manufacturer are to reduce test time and consequently cost. Fig. 1 provides the manufacturing process of the cellular phone including the operation time of each process. We find that the RF functional test is the bottleneck of entire process. The RF test is aimed at inspecting whether or not the mobile phone receive/transmit signal satisfies the enabled transmission interval (ETI) protocol on different channels and different power levels. In order to ensure the quality of communication of mobile phones, the manufacturers usually add extra inspection items, such as several different frequency channels and power levels, resulting in the inspection time being increased and as a result the test procedure becomes a bottleneck.

If we can reduce the numbers of items tested in the RF function test, without losing inspection accuracy, then the inspection time will be shortened. At the same time, this reduction of test items will help lower the cost of testing and the manufacturing time.

5.2. Data collection

The 1006 RF function test data containing 62 test items (27 are continuous attributes and 35 are discrete attributes) as described in Table 3 are collected. There are eight major RF functional tests: the power versus time (PVT; symbol: A), the power level (TXP; symbol: B), the phase error and the frequency error (PEFR; symbol: C), the bit error rate (BER −20; symbol: D and BER −102; symbol: E), the ORFS-spectrum due to the switching transient (ORFS_SW; symbol: F), the ORFS-spectrum due to modulation (ORFS_MO; symbol: G), the Rx level report accuracy (RXP_Lev_Err; symbol: H), and the Rx level report quality (RXP_QUALITY; symbol: I). According to different channels and power levels, each test item has several separate test attributes. Each form of the test attributes is to be represented as: test item-channel-power level. In the 1006 collected objects, there are only 44 negative examples (defective products) and the rests are positive examples (normal products). The defective rate is about 4%. We separate the 1006 examples into a training set which includes 756 objects (722 objects are normal and 34 objects are defective) and a test set that includes 250 objects (240 objects are normal and 10 objects are defective).

5.3. Data preparation

In this case, the inspection data are collected automatically by computers, and there are no missing values. In the data preparation phase we remove 11 attributes (D105, I10–102, D725, I72–102, D1145, I114–102, D9655, I965–102, D6880, I688–102, D8750) that have the same value. These 11 attributes have no classification ability. Consequently, only 51 attributes

Table 3
Test items of the RF function

No.	Test items	Code
1	TXP	B105
2	PEFR	C105
3	BER(−20)	D105
4	BER(−102)	E105
5	ORFS_SW	F105
6	ORFS_MO	G105
7	RXP_Lev_Err	H10–102
8	RXP_QUALITY	I10–102
9	TXP	B725
10	PFER	C725
11	BER(−20)	D725
12	BER(−120)	E725
13	ORFS_SW	F725
14	ORFS_MO	G725
15	TXP	B727
16	TXP	B7211
17	TXP	B7219
18	RXP_Lev_Err	H72–102
19	RXP_QUALITY	I72–102
20	TXP	B1145
21	PFER	C1145
22	BER(−20)	D1145
23	BER(−102)	E1145
24	ORFS_SW	F1145
25	ORFS_MO	G1145
26	RXP_Lev_Err	H114–102
27	RXP_QUALITY	I114–102
28	TXP	B9655
29	PFER	C9655
30	BER(−20)	D9655
31	BER(−102)	E9655
32	ORFS_SW	F9655
33	ORFS_MO	G9655
34	RXP_Lev_Err	H965–102
35	RXP_QUALITY	I965–102
36	TXP	B5220
37	PEFR	C5220
38	BER(−20)	D5220
39	BER(−102)	E5220
40	ORFS_SW	F5220
41	ORFS_MO	G5220
42	RXP_Lev_Err	H522–102
43	RXP_QUALITY	I522–102
44	TXP	B6880
45	PFER	C6880
46	BER(−20)	D6880
47	BER(102)	E6880
48	ORFS_SW	F6880
49	ORFS_MO	G6880
50	TXP	B6883
51	TXP	B6887
52	TXP	B68815
53	RXP_Lev_Err	H688–102
54	RXP_QUALITY	I688–102
55	TXP	B8750
56	PEFR	C8750
57	BER(−20)	D8750
58	BER(−102)	E8750
59	ORFS_SW	F8750
60	ORFS_MO	G8750
61	RXP_Lev_Err	H875–102
62	RXP_QUALITY	I875–102

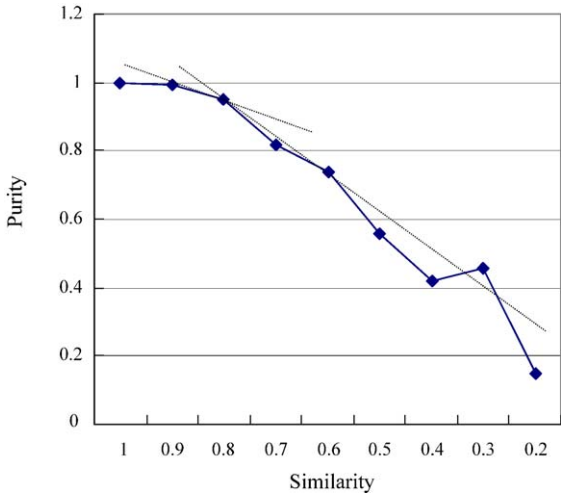

Fig. 6. The purities of IGs in different similarity.

labeled $X1$–$X51$ are left to be analyzed further. Before implementation, these collected data need to be normalized due to different scale of attributes' value, which may affect the performance of Fuzzy ART. All values of attributes were normalized to the interval [0,1] by employing a min–max normalization equation, shown as Eq. (4). In this equation, $\max_i$ is the maximum and $\min_i$ is the minimum of the ith attribute values, and v_{ij} is the value of ith attribute of jth objects and v'_{ij} is the normalized value.

$$v'_{ij} = \frac{v_{ij} - \min_i}{\max_i - \min_i} \tag{4}$$

5.4. Information granulation

Next, we utilize the Fuzzy ART to construct IGs. The proposed procedure is programmed with the use of the software of Matlab 6.1. The purities of different similarities are shown in Fig. 6. The purities of similarities 0.8 and 0.9 are very close to each other. The centralities of the two similarities described in Fig. 7 are in a similar situation. However, the similarity 0.8 owns fewer data size than that of similarity 0.9. This means that similarity 0.8 can reduce more detailed information than similarity 0.9. It is also evident that a turning point exists at the similarity 0.8 in Fig. 6.

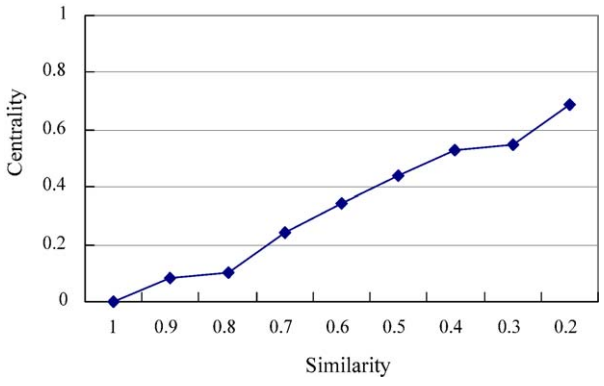

Fig. 7. The centralities of IGs in different similarity.

Table 4
The information granules described as hyperbox form

	X																																																			*Y*
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	
L1	4	3	1	1	1	1	1	1	1	1	1	1	1	2	1	2	2	1	1	1	1	5	2	1	1	1	1	3	3	1	1	1	1	1	1	3	3	1	1	1	2	3	2	1	5	2	1	1	1	1	1	1
U1	4	4	1	1	1	1	2	1	1	1	1	2	2	4	1	3	2	1	1	1	1	5	2	1	1	1	1	3	5	2	1	2	1	2	1	3	7	1	1	1	2	3	3	1	5	4	1	2	2	1	1	1
L2	3	1	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1	1	1	1	1	4	1	1	1	1	1	3	2	1	1	2	1	1	1	3	1	1	1	1	2	3	3	1	4	1	1	1	1	1	1	1
U2	4	3	1	1	1	1	2	1	1	1	1	3	3	4	1	3	2	1	2	1	1	5	1	1	2	1	1	3	4	1	3	2	1	2	1	3	5	1	1	1	2	3	3	1	5	4	2	2	2	1	1	1
L3	2	1	1	1	1	1	1	1	1	1	1	1	1	2	1	3	2	1	1	1	1	2	1	1	1	1	1	3	1	1	1	2	1	1	1	2	1	1	1	1	2	3	3	1	3	1	1	1	1	1	1	1
U3	4	3	1	1	1	1	2	1	1	1	1	2	4	4	1	4	2	1	1	1	1	5	2	1	2	1	1	3	4	3	2	2	1	2	1	3	5	1	1	1	2	3	3	1	5	4	2	2	2	1	1	1
L4	1	1	1	1	1	1	1	1	1	1	1	1	2	2	1	1	1	1	1	1	1	2	1	1	1	1	1	2	1	1	1	2	1	1	1	3	1	1	1	1	2	3	3	1	5	2	1	1	1	1	1	1
U4	4	3	1	1	1	1	2	2	1	1	1	2	3	3	1	4	2	1	1	1	1	5	2	1	2	1	1	3	4	1	2	2	1	2	1	3	7	1	1	1	2	3	3	1	5	4	2	2	2	1	1	1
L5	4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	1	1	5	1	1	1	1	1	3	1	1	1	2	1	1	1	3	1	1	1	1	2	3	2	1	5	1	1	1	1	1	1	1
U5	4	4	1	1	1	1	2	1	1	1	1	2	3	4	1	3	2	1	1	1	1	5	2	1	2	1	1	3	5	1	2	2	1	2	1	3	6	1	1	1	2	3	3	1	5	4	3	2	2	1	1	1
–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
L31	3	1	1	1	1	1	1	1	1	1	1	1	1	3	1	3	1	1	1	1	1	3	2	1	1	1	1	3	1	1	1	2	1	3	1	3	1	1	1	1	2	3	3	1	5	2	1	1	1	2	1	2
U31	4	3	1	1	1	2	1	1	1	1	1	2	2	4	2	3	2	1	1	1	2	5	2	1	1	1	2	3	4	1	1	2	1	3	1	3	1	1	1	1	2	3	3	2	5	3	2	2	1	2	1	2
L32	1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	1	1	1	1	3	1	1	1	1	1	1	4	1	1	1	1	1	2	2	1	1	1	1	1	1	1	2	4	1	2	1	2	2
U32	3	3	2	1	1	1	1	1	5	1	1	2	3	2	1	3	2	2	1	1	1	1	1	5	1	1	1	1	2	1	4	1	1	2	2	1	4	2	1	1	1	1	1	1	1	4	4	1	2	1	2	2
L33	3	1	1	1	1	1	1	1	1	1	1	1	2	3	1	1	1	1	1	1	1	4	1	1	1	1	1	3	1	1	1	2	1	1	1	3	1	1	1	1	2	3	3	1	1	1	1	1	1	1	1	2
U33	4	2	1	1	1	2	2	1	1	1	1	2	2	4	1	3	2	1	1	1	2	5	2	1	2	1	2	3	4	1	1	2	1	2	1	3	1	1	1	1	3	3	3	2	5	2	2	2	1	1	1	2

Notes: (1) L1 and U1 represent the lower limit and upper limit of the 1st IG. (2) *X* represents the condition attributes, and *Y* is the decision attribute. (3) The data shown in the table are discretized.

Table 5
The IGs with the addition of sub-attributes

	Original attributes																					
	*X*1					*X*2					*X*3		*X*4		*X*5		*X*6		··· ··· ···	*X*51		*Y*
	*X*11[a] (*X*1 = 1)	*X*12[a] (*X*1 = 2)	*X*13[a] (*X*1 = 3)	*X*14[a] (*X*1 = 4)	*X*15[a] (*X*1 = 5)	*X*21[a] (*X*2 = 1)	*X*22[a] (*X*2 = 2)	*X*23[a] (*X*2 = 3)	*X*24[a] (*X*2 = 4)	*X*25[a] (*X*2 = 5)	*X*31[a] (*X*3 = 1)	*X*32[a] (*X*3 = 2)	*X*41[a] (*X*4 = 1)	*X*42[a] (*X*4 = 2)	*X*51[a] (*X*5 = 1)	*X*52[a] (*X*5 = 2)	*X*61[a] (*X*6 = 1)	*X*62[a] (*X*6 = 2)	··· ··· ···	*X*175[a] (*X*51 = 1)	*X*176[a] (*X*51 = 2)	
IG #1	0	0	0	1	0	0	0	1	1	0	1	0	1	0	1	0	1	0	··· ··· ···	1	0	1
IG #2	0	0	1	1	0	1	1	1	0	0	1	0	1	0	1	0	1	0	··· ··· ···	1	0	1
IG #3	0	1	1	1	0	1	1	1	0	0	1	0	1	0	1	0	1	0	··· ··· ···	1	0	1
IG #4	1	1	1	1	0	1	1	1	0	0	1	0	1	0	1	0	1	0	··· ··· ···	1	0	1
IG #5	0	0	0	1	0	1	1	1	1	0	1	0	1	0	1	0	1	0	··· ··· ···	1	0	1
IG #6	0	1	1	1	0	1	1	1	1	0	1	0	1	0	1	0	1	0	··· ··· ···	1	0	1
IG #7	0	0	1	1	0	1	1	1	0	0	1	0	1	0	1	0	1	0	··· ··· ···	1	0	1
⋮	···	···	···	···	···	···	···	···	···	···	···	···	···	···	···	···	···	···	··· ··· ···			
⋮	···	···	···	···	···	···	···	···	···	···	···	···	···	···	···	···	···	···	··· ··· ···			
IG #27	1	0	0	0	0	0	1	1	1	1	0	1	1	0	1	0	1	0	··· ··· ···	1	1	2
IG #28	0	0	0	1	0	1	1	1	0	0	1	0	1	0	1	0	1	1	··· ··· ···	1	0	2
IG #29	0	1	1	1	0	0	0	0	0	1	1	0	1	1	1	0	1	0	··· ··· ···	1	0	2
IG #30	0	0	1	1	0	1	1	0	0	0	0	1	1	0	1	0	1	0	··· ··· ···	0	1	2
IG #31	0	0	1	1	0	1	1	1	0	0	1	0	1	0	1	0	1	1	··· ··· ···	1	0	2
IG #32	1	1	1	0	0	1	1	1	0	0	0	1	1	0	1	0	1	0	··· ··· ···	0	1	2
IG #33	0	0	1	1	0	1	1	0	0	0	1	0	1	0	1	0	1	1	··· ··· ···	1	0	2

[a] Sub-attributes.

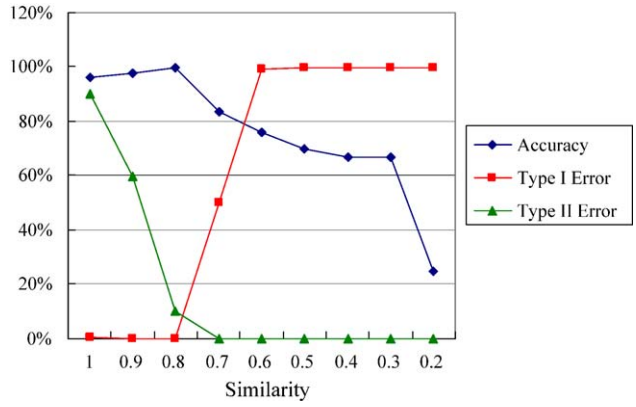

Fig. 8. Sensitivity analysis in different similarities.

In order to choose a better solution, we carry out the sensitivity analysis in different similarities. The result is shown in Fig. 8. The classification accuracy at similarity 0.8 (99.6%) is slightly higher than that at similarity 0.9 (97.6%). Also, at similarity 0.8, we have a lower level of Type II errors with the same level of Type I errors (0%). Hence, it seems that the similarity of 0.8 is a better choice compared to similarity 0.9 in this case. It is a difficult to determine a suitable similarity. In this case, we not only consider the classification accuracy, but also the level of Type II error. Consequently the results should be better than the original one.

Once the similarity is determined, Fuzzy ART is again utilized to construct IGs. We set the Fuzzy ART parameters α, β, ρ to be 0.01, 1, 0.8, respectively. Thirty-three IGs are constructed. Twenty-four of them are IGs of good products and the rest belong to the defective products. Each IG is described by using the lower limit and upper boundary (hyperbox form) as shown in Table 4. In addition, the overlapping parts among granules are separated from the original attribute by designating them as new attributes or so-called ''sub-attributes.'' We divide the original attribute X_1 into sub-attributes X_{11}, X_{12}, X_{13}, X_{14}, X_{15} and the same happens for the other attributes. These 33 granules are rewritten as Table 5.

5.5. Feature selection and knowledge acquisition

Now three feature selection algorithms, rough sets method, decision tree (C4.5 algorithm) and neural network, are implemented. The computation of rough sets is executed using the ROSETTA software (http://www.idi.ntnu.no/~aleks/rosetta/). See5 (C4.5 commercial version) software was utilized to construct a decision tree. In See5 there are two parameters that can be tuned during the pruning phase: the minimal number of examples represented at any branch of any feature-value test and the confidence level of pruning. To avoid the occurrence of over-fitting and generating a simple tree, 2 was set as the minimum number of instances at each leaf, and the confidence level for pruning was set at 25%. The inputs and outputs of the decision tree and rough sets are 176 sub-attributes and defined classes respectively. In the neural network based method, the back-propagation neural network with one hidden layer is adopted and implemented using Professional II PLUS software. All parameters of the BPNN are obtained by trial and error, including the number of training iterations and the structure of the network.

Table 6
The implementation results by rough sets

Method	After granulation		Before granulation	
	Training phase	Test phase	Training phase	Test phase
Data size (good:bad)	33 (24:9)	14 (9:5)	756 (722:34)	250 (240:10)
Type I error (%)	0	0	0.7	0.4
Type II error (%)	0	10	0	90
Accuracy (%)	100	99.6	99.34	96
No. of rules	4		433	
Extracted features	B725, H114-102		C105, B727, B1145, H114-102, C9655, H522-102, B8750, E8750	

Note: (24:9) is the proportion of good products to bad products.

Table 7
The implementation results by decision tree (C4.5)

Method	After granulation		Before granulation	
	Training phase	Test phase	Training phase	Test phase
Data size (good:bad)	33 (24:9)	14 (9:5)	756 (722:34)	250 (240:10)
Type I error (%)	0	0	0	0
Type II error (%)	0	10	23.53	40
Accuracy (%)	100	99.6	98.9	98.4
No. of rules	3		7	
Extracted features	B725, H114-102		E105, C725, G725, H72-102, H965-102, H688-102	

Table 8
The implementation results by BPNN (full attributes)

Method	After granulation		Before granulation	
	Training phase	Test phase	Training phase	Test phase
Data size (good:bad)	33 (24:9)	14 (9:5)	756 (720:34)	250 (240:10)
Type I error (%)	0.5	0	0.14	0
Type II error (%)	11.76	0	29.41	50
Accuracy (%)	98.9	100	98.54	98
Structure	16-15-1		17-4-1	
Parameters	Learning rate: 0.2 Momentum: 0.9 50000 iterations		Learning rate: 0.2 Momentum: 0.8 2000 iterations	
Extracted features	B7211, H114-102, E8750, B8750, C8750, B725, H965-102, H688-102, H10-102, B727, E5220, B7219, C105, C6880, C9655, B68815		C9655, B725, C725, B8750, B105, B727, C8750, F1145, B5220, B7211, H114-102, B6880, B68815, F725, B6887, E1145, I522-102	

Implementation results are shown in Tables 6–8. In Tables 6 and 7, our proposed approach obviously outperforms the traditional approach without granulation, in both classification accuracy and Type II error. In addition, fewer knowledge rules and attributes are obtained. In Table 8, the classification accuracy and Type II error of our approach are still better than those by the original BPNN. All the attributes, kept and ranked by priority, are listed in Table 8. By comparing the implementation results of these three methods, six attributes {B7211, H114–102, B725, B8750, C8750 and E8750} are reserved as final test items for the RF functional test. The knowledge rules listed in Fig. 9(a and b) are generated by using rough sets and decision tree methods. These rules may not only help engineers to predict the yield rate of products, but may also enhance the performance of knowledge management.

5.6. The benefits

By implementing the proposed method, test items are reduced from 62 to 6 items. The test time is reduced from 190 to 95 s. The amount of employed test equipment is reduced from eight machines to four machines. As a result the company will save about US$ 200,000 per year. In addition we should not forget the resulting rise in customer satisfaction and the reduction in risk for the customers. The potential benefits of implementation are substantial.

6. Discussions

In most cases of inspection data, the amount of good products is far greater than the amount of defective products. The few defective products are usually viewed as outliers and

(a) Rule 1:
$B725 \in [31.7440, 31.7446)$ AND $H114-102 \notin (1,*]$ TEHN Class = Good Product
[Accuracy: 1.0; Supports: 24]
Rule 2:
$B725 \in [31.7440, 31.7446)$ AND $H114-102 \in (1,*]$ TEHN Class = Bad Product
[Accuracy: 1.0; Supports: 2]
Rule 3:
$B725 \notin [31.7440, 31.7446)$ AND $H114-102 \notin (1,*]$ TEHN Class = Bad Product
[Accuracy: 1.0; Supports: 4]
Rule 4:
$B725 \notin [31.7440, 31.7446)$ AND $H114-102 \in (1,*]$ TEHN Class = Bad Product
[Accuracy: 1.0; Supports: 3]

(b) Rule 1:
$B725 \in [31.7440, 31.7446)$ AND $H114-102 \notin (1,*]$ TEHN Class = Good Product
[Accuracy: 0.962; Supports: 24]
Rule 2:
$B725 \notin [31.7440, 31.7446)$ TEHN Class = Bad Product
[Accuracy: 0.889; Supports: 7]
Rule 3:
$H114-102 \in (1,*]$ TEHN Class = Bad Product
[Accuracy: 0.857; Supports: 5]

Fig. 9. (a) Knowledge rules extracted by rough sets. (b) Knowledge rules extracted by decision tree (C4.5).

are removed in the generalization phase of the classification tools. Actually, all normal products look alike, and the abnormal products have individual styles. That phenomenon is also noted by Taguchi and Jugulum [26]. We should pay more attention to this, and consider the categories of instances instead of the data size when developing feature selection algorithms.

Classification accuracy is widely utilized to evaluate the performance of classification tools. But, in modern manufacturing systems, this becomes meaningless due to the fact that the defective rate of products is so extremely low. Therefore, it becomes necessary to consider Type II error together.

7. Conclusions

Traditional data mining tools tend to generate a huge amount of knowledge rules and lead to a high level of Type II errors when dealing with imbalanced data. This study proposed a neural network based information granulation approach which removes unnecessary details and provides a better insight into the essence of the data. The proposed approach not only extracts fewer knowledge rules, but also outperforms the traditional methods regarding the amount of Type II errors and classification accuracy.

A real case study of a cellular phone test process was employed to demonstrate the effectiveness of our proposed approach. When encountering imbalanced data, our proposed method is effective in removing unnecessary RF function test items, saving testing costs and shortening the inspection time. It is suitable for reducing the inspection process in the high technology industry, especially now that we are facing the six-sigma age, i.e. the defective rate of products is becoming extremely low.

The experimental results also show that there is a trade-off relationship between the Type I and Type II errors. The proposed method can reduce the level of Type II errors without increasing the level of Type I errors. This is very important to OEM/ODM manufacturers because a high level of Type II errors will inevitably lead to orders being lost.

The inconsistency of the extracted attributes when using different feature selection methods is an important issue for future research, because it might confuse users (engineers) when applying these feature selection techniques in practice. To solve the inconsistence, a robust approach is needed to be developed in the future.

Acknowledgements

This work was supported in part by National Science Council of Taiwan (Grant No. NSC-93-2213-E-007-111).

References

[1] Agilent Technologies, Agilent's TS-5550 cellular phone functional test platform. Available at: http://www.home.agilent.com.

[2] R. Akbani, S. Kwek, N. Japkowicz, Applying support vector machines to imbalanced datasets, in: J.-F. Boulicaut, et al. (Eds.), ECML 2004: 15th European Conference on Machine Learning, *LNAI* 2004; 3201: 39–50.

[3] H. Almuallim, T. Dietterich, Learning boolean concepts in the presence of many irrelevant features, Artificial Intelligence 69 (1994) 279–305.

[4] A. An, Y. Wang, Comparisons of classification methods for screening potential compounds, in: Proceedings of the IEEE International Conference on Data Mining (ICDM.01), San Jose, CA, USA, (2001), pp. 11–18.

[5] G. Batista, R.C. Prati, M.C. Monard, A study of the behavior of several methods for balancing machine learning training data, SIGKDD Explorations 6 (1) (2004) 20–29.

[6] L. Bruzzone, S.B. Serpico, Classification of imbalanced remote-sensing data by neural networks, Pattern Recognition Letters 18 (1997) 1323–1328.

[7] L. Burke, S. Kamal, Neural networks and the part family/machine group formation problem in cellular manufacturing: a framework using Fuzzy ART, Journal of Manufacturing Systems 14 (1995) 148–159.

[8] G.A. Carpenter, S. Grossberg, D.B. Rosen, Fuzzy ART: fast stable learning and categorization of analog patterns by an adaptive resonance system, Neural Networks 4 (1991) 759–771.

[9] G. Castellano, A.M. Fanelli, Information granulation via neural network-based learning, in: IFSA World Congress and 20th NAFIPS International Conference, 2001, 3059–3064.

[10] N.V. Chawla, N. Japkowicz, A. Kolcz, Editorial: special issue on learning from imbalanced data sets, SIGKDD Explorations 6 (1) (2004) 1–6.

[11] J. Han, M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers, San Francisco, 2001.

[12] M. Hannikainen, T.D. Hamalainen, M. Niemi, J. Sarinen, Trends in personal wireless data communications, Computer Communications 25 (2002) 84–99.

[13] R. Li, Z. Wang, Mining classification rules using rough sets and neural networks, European Journal of Operational Research 157 (2004) 439–448.

[14] O. Oyeleye, E.A. Lehtihet, A classification algorithm and optimal feature selection methodology for automated solder joint defect inspection, Journal of Manufacturing Systems 17 (1998) 251–262.

[15] Z. Pawlak, Rough sets and fuzzy sets, Fuzzy Sets and Systems 17 (1985) 99–102.

[16] W. Pedrycz, G. Vukovich, Feature analysis through information granulation and fuzzy sets, Pattern Recognition 35 (2002) 825–834.

[17] P.C. Pendharkar, J.A. Rodger, G.J. Yaverbaum, N. Herman, M. Benner, Association, statistical, mathematical and neural approaches for mining breast cancer patterns, Expert Systems with Applications 17 (1999) 223–232.

[18] F. Provost, T. Fawcett, Robust classification for imprecise environments, Machine Learning 42 (2001) 203–231.

[19] J. Quinlan, Induction of decision trees, Machine Learning 1 (1986) 81–106.

[20] J. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, Bari, 1993.

[21] R. Reed, Pruning algorithms- a survey, IEEE Transactions on Neural Networks 5 (1993) 740–747.

[22] R.W. Swiniarski, L. Hargis, Rough sets as a front end of neural-networks texture classifiers, Neurocomputing 36 (2001) 85–102.

[23] R.W. Swiniarski, A. Skowron, Rough set methods in feature selection and recognition, Pattern Recognition Letters 24 (2003) 833–849.

[24] C.-T. Su, J.-H. Hsu, C.-H. Tsai, Knowledge mining from trained neural network, Journal of Computer Information Systems (2002) 61–70.

[25] C.-T. Su, Y.-R. Shiue, Intelligent scheduling controller for shop floor control systems: a hybrid genetic algorithm/decision tree learning approach, International Journal of Production Research 12 (2003) 2619–2641.

[26] G. Taguchi, R. Jugulum, The Mahalanobis-Taguchi Strategy: A Pattern Technology System, John Wiley & Sons, New York, 2002.

[27] A. Verikas, M. Bacauskiene, Feature selection with neural networks, Pattern Recognition Letters 23 (2002) 1323–1335.

[28] VI Services Network, Mobile phone online test application. Available at: http://www.vi-china.com.cn/.

[29] B. Walczak, D.L. Massart, Tutorial: rough sets theory, Chemometrics and Intelligent Laboratory Systems 47 (1999) 1–16.

[30] B.-X. Xiu, W.-M. Zhang, S. Wang, J.-Y. Tang, Generalized multilayer granulation and approximations, in: Proceedings of the Second International Conference on Machine Learning and Cybernetics, 2003, pp. 1419–1423.
[31] L.A. Zadeh, Fuzzy graphs, rough sets and information granularity, in: Proceedings of the Third International Workshop On Rough Sets and Soft Computing, San Jose, USA, 1994.
[32] F. Zhu, S. Guan, Feature selection for modular GA-based classification, Applied Soft Computing 4 (2004) 381–393.

Chao-Ton Su is professor of Department of Industrial Engineering and Engineering Management at National Tsing Hua University, Taiwan. He received his PhD in industrial engineering from the University of Missouri, Columbia, USA. Dr Su has received two-time Outstanding Research Awards from the National Science Council, Taiwan. He also obtained the Individual Award of the National Quality Awards of the Republic of China (Taiwan).

Long-Sheng Chen received his BS and MS degrees in industrial management from National Cheng Kung University, Tainan, Taiwan in 1998 and 2000, respectively. He is currently a PhD candidate in the Department of Industrial Engineering and Management at National Chiao Tung University, Hsinchu, Taiwan. His research interests include granular computing, machine learning, data mining, class imbalance problems and neural networks applications.

Tai-Lin Chiang is associate professor of Department of Business Administration at Ming Hsin University of Science and Technology, Taiwan. He received his PhD in industrial engineering and management from National Chiao Tung University, Hsinchu, Taiwan. He is a certified Master Black Belt of the American Society for Quality.