Paper

Enhancement of Prediction for Manufacturing System Using Bayesian Decision Recognition

Yang Jianhua* Student Member
Yasutaka Fujimoto* Member

A decision model stemmed from Bayesian theorem is proposed to describe the process of decision making for job sequence in manufacturing system. The construction of feature vector is firstly discussed with respect to the manufacturing system's characteristic. Then a non-parametric model is employed to deal with general distribution for decision acquisition, where a binary division methodology is developed to limit the size of non-parametric model, including elimination of irrelevant features. At last, a PCB manufacturing system is given to demonstrate the efficiency of the model.

Keywords: simulation, prediction, Bayesian theorem, decision recognition

1. Introduction

Precisely predicting the completion time of jobs in manufacturing systems can meet the needs from clients, improving the competitive ability of a company. For some manufacturing systems it is easy to predict because all jobs are strictly arranged and processed according to well-defined routings. But for some manufacturing systems, it is often hard to obtain a satisfying answer due to complex constraints and uncertain factors. For the latter we consider the case that the deviation of completion time of jobs is primarily influenced by the job sequence determined by operators.

Given a manufacturing system composed of a set of machines and a set of buffers, operators select jobs from buffers and mount them on corresponding machines. Suppose that each operator makes his decision based on instructions given by upper level, his experience, and even his mood of that day. Prediction errors might be decreased if we can effectively recognize the operator's decision mechanism and determine the job sequence. In this paper, we employ Bayesian decision theory [1] to describe the operator's decision mechanism. That is, we will build a Bayesian decision model based on the past behavior of the operator, and then predict how the operator determine the sequence of jobs to be mounted on corresponding machine in the future.

Bayesian decision theory is widely applied in filed of pattern recognition [2]. While it is introduced to human decision recognition, some characteristics appear. A remarkable one is that we can only get results based on existed data because random sampling is impossible. Thus sufficient data are always necessary in order to avoid defects of sampling data themselves. Another notable one is that features are often of implication. Some transformations are needed to obtain distributions over corresponding features. Meanwhile, we attempt to apply a non-parametric distribution to describe human decision recognition for the sake of generality because we know nothing about distribution type of the human decision recognition when Bayesian method is employed. The question is that it may result in the explosion of model scale [3] [4]. In this paper we propose a binary division methodology to overcome the obstacle. Furthermore, regarding practical factories, we usually do not know which factors influence the human's decision before the recognition model is constructed. Obviously factors irrelevant to decision mechanism are redundant and should be deleted [5]. Proposed binary division methodology shows that it is also possible to eliminate factors irrelevant to decision mechanism.

The remainder of the paper is organized as follows. In Section 2 the Bayesian equation is introduced to describe the decision recognition and three kinds of transformations from job attributes to features are also given. In Section 3 the reasons for using non-parametric model are illustrated and detailed binary division methodology is given. In Section 4 the numeric computation for actual manufacturing system is described and a PCB (printed-circuit-board) manufacturing system is given to demonstrate our proposed method. Finally concluding remarks appear in Section 5.

2. Bayesian Decision Recognition

2.1 Bayesian Thinking and Prediction The basic concept of Bayesian thinking is very simple [6]. Let Ω be sampling space, which is composed of n independent hypotheses, noted by $\{B_1, B_2, \cdots, B_n\}$. Given result x the probability of occurrence of hypothesis $B_k (k = 1, 2, \cdots, n)$ can be computed by following equation:

$$p(B_k|x) = p(x|B_k)p(B_k)/p(x) \quad \cdots\cdots\cdots\cdots\cdots (1)$$

* Dept. of Electrical & Computer Engineering, Yokohama National University
79-5, Tokiwadai, Hodogayaku, Yokohama 240-8501

Equation (1) is referred to as Bayesian equation or Bayesian theorem, which tells us that a posterior probability $p(B_k|x)$, the updated prediction, is the product of the conditional probability of the hypothesis, given the influence of the result being investigated, multiplying the prior probability $p(B_k)$ of those results, divided by the total probability of x which assures the resulting quotient fall on the $[0, 1]$ interval, as all probabilities should be. Bayesian theorem can be also expanded to probability distribution using the same form.

For recognizing operator's behavior, we let hypotheses be operator's decision D, let results be the decision data denoted by a feature vector X. According to equation (1), we get

$$p(D|X) = p(X|D)p(D)/p(X). \quad (2)$$

The feature vector X is usually of implication and some transformations should be done, which will be discussed in detail in the Section 2.2. Furthermore the prior distribution $p(D)$ can be assumed as a uniform distribution implying that we know nothing about decision mechanism beforehand. As a result, the posterior distribution $p(D|X)$ is fundamentally determined by $p(D|X)/p(X)$, denoted by

$$\eta(X|D) = p(X|D)/p(X). \quad (3)$$

The $\eta(X|D)$ can be regarded as a force, the operator's decision, driving the prior probability distribution $p(X)$ to the posterior probability distribution $p(X|D)$. Thus $\eta(X|D)$ is referred to as decision acquisition distribution.

Finally let J be the current jobs waiting in a buffer at current time F. For each job $j \in J$ its feature vector is represented by X_j^F. Then we can predict that operator would select a job $j^* \in J$ such that

$$\eta(X_{j^*}^F|D) = \max_j \eta(X_j^F|D). \quad (4)$$

2.2 Feature Vector Transformation The feature vector is the basic element for Bayesian application. The description of feature vector should be done before decision recognition. We discuss in this section how to get a recognition distribution over one feature where one machine with a buffer is considered.

Let's begin with a continuous variable, h, representing an attribute of jobs, for example the processing time. Let J_i stand for a set of identifiers of jobs disordered, waiting in a buffer at time i where the total number of jobs is changeable at different time point. The attribute value of each job $j \in J_i$ is denoted by $h(j)$. Let θ_i be a sequenced set over J_i, specifying the jobs' sequence to be processed. For an example, given two job sets $J_1 = \{1, 2, 3, 4, 5\}$ and $J_2 = \{6, 7, 8, 9\}$ at time 1 and 2, where $h(1) = 13$, $h(2) = 38$, $h(3) = 30$, $h(4) = 10$, $h(5) = 25$, $h(6) = 20$, $h(7) = 18$, $h(8) = 50$, and $h(9) = 49$, suppose that we have known their processing sequenced sets, $\theta_1 = \{4, 1, 5, 3, 2\}$ and $\theta_2 = \{7, 6, 8, 9\}$, implying that jobs have been mounted according to 4→1→5→3→2 and 7→6→8→9. Now given the current jobs in buffer by $J_F = \{10, 11, 12, 13\}$ at current time F, where $h(10) = 12$, $h(11) = 5$, $h(12) = 23$, and $h(13) = 6$, the question is what the θ_F will be.

From θ_1, θ_2, we can intuitively infer that the job with smaller attribute value might be earlier mounted except for two special cases. Thus we can get a heuristic rule that the operator will most probably select the job with minimum attribute value for next mount. If we predict the future sequence of jobs J_F as $\theta_F = \{2, 4, 1, 3\}$, we believe that errors might be minimized. For mathematical description of above process, a feature variable x_1 is introduced as

$$x_1 = h(j) - \min_k h(k), \quad (5)$$

where $h(j)$ stands for the attribute value of the selected job and $h(k)$ for the attribute value of each job including the selected one. For θ_1, it is always such that $x_1 \equiv 0$. Also it can be represented by a probability variable $p(x_1 = 0|D) = 1$. If the second job sequence θ_2 is taken into account, a probability distribution on x_1 will appear.

However, we don't know whether an operator has an idea that he will select a job with minimum attribute value. Maybe he selects a job with maximum attribute value. So we get another formula

$$x_2 = \max_k h(k) - h(j). \quad (6)$$

As a result, we can get two feature variables given in formula (5) and (6) for the attribute h. We define the attribute h be *magnitude sensitive* to the operator because the operator will determine job sequence according to the magnitude of attribute value. In some case, we can conclude that one of formulas works. In some case we might find that both of them function.

Next, we introduce another type of attribute, which is referred to as *similarity sensitive* to operator. Let a discrete variable,s, be an attribute of jobs. The feature variable x_3 is calculated by

$$x_3 = \begin{cases} 0 & s(j) = s(j_0) \\ 1 & s(j) \neq s(j_0) \end{cases}, \quad (7)$$

where $s(j)$ stands for the attribute value of the selected job and $s(j_0)$ for the attribute value of the previously mounted job. The formula (7) implies that the operator possibly select the job with the same attribute value for next mount. E.g., selecting the same size of parts may decrease the preparation time.

Based on the investigation to real manufacturing system, above two types of transformed features are usually involved. Probably more complicated transformation exists in real world, but with respect to the manufacturing system instance in Section 4.2, we attempt to end it by just introducing the third well-used attribute, the *preference sensitive* one. The third one reveals its characteristic by that the operator will probably select a job based on the discrete attribute value. For example, let the discrete attribute value $d \in \{280, 300, 340\}$ and the operator might prefer 300 to 280, and prefer 280 to 340. For this case, we almost need not transform it but let

$$x_4 = d. \quad \cdots\cdots (8)$$

In the end, we summarize forms of transformation from attributes to feature vector as: (1) Magnitude sensitive; (2) Similarity sensitive; (3) Preference sensitive; and (4) The others.

3. Description of Decision Distribution

3.1 Non-parametric Distribution To identify the operator's decision process, the distribution of $\eta(X|D)$ should be known. Generally, three kinds of models are employed to describe multi-dimensional distribution. They are parametric distribution model, non-parametric distribution model, and semi-parametric distribution model. The last one is a combination of previous two models.

Hereby, we attempt to consider using non-parametric distribution model to describe our problem based on following facts.

① Nothing about distribution is known beforehand.
② Discrete and continuous distributions coexist.
③ Irrelevant factors are also involved.

A non-parametric distribution model is generally described by dividing sampling space into many tiny domains, where probability density $p(X)$ is almost constant. Let a domain be S, corresponding volume be V. The probability of feature vector in S can be calculated by

$$p(S) = \int_S p(X)dX \cong p(X)V. \quad \cdots\cdots (9)$$

According to Monte Carlo simulation [7], given m sampling data, if among them k data enter the domain S, the probability of feature vector in S can be obtained by

$$p(S) = k/m. \quad \cdots\cdots (10)$$

Thus the probability density in domain S can be determined by

$$p(X) = \frac{k}{mV}. \quad \cdots\cdots (11)$$

Two basic methods for modeling non-parametric distribution are kernel density method and k-nearest neighbors method. For kernel density method [8][9], the probability density of a domain can be calculated by fixing the volume of the domain, counting the data that fall in it. For k-nearest neighbor method [10], it can be calculated by fixing the number of data that fall in the domain, changing the volume of the domain.

A main drawback of kernel density method is that a large domain division might result in low smooth while a small domain division, which is so-called hyper cube, might result in low reliability because of limited history data. Moreover, sometimes its implementation is almost infeasible. Following is an example, which illustrates the size obstacle of model.

Given a non-parametric model composed of 6 features, four features are magnitude sensitive and continuous, one is similarity sensitive, and one is 4-value discrete. Let each of continuous features be divided into 10 intervals with the same width. Let the total number of the domains be Z. Then we have $Z = 10^4 \times 2 \times 4 = 8 \times 10^4$. If we found 10 intervals are not enough to describe it, for instance, 100 intervals needed, it will become $Z = 8 \times 10^8$. Obviously it is almost unimaginable to build such a model.

K-nearest neighbors method emphasizes that the volume of domain is changeable, fixing the counts of data that fall in the domain. Here, the question is that such a domain is usually hard to be obtained.

In fact, no matter what kind of method, the fundamental problem is the division of sampling space. In next section, a binary division method is proposed to provide such a solution, where both the volume and counts are changeable.

3.2 Binary Division Methodology Noticed that an effective decision means that decision distribution $p(D|X)$ is not a uniform distribution. The larger difference among domains generally implies the more effective decision. So we should emphasize the feature with less variance and consider how to divide it firstly. Here a binary division method is one of possible choices.

Let Ω be the sampling space, $X = [x_1 \quad x_2 \quad \cdots \quad x_k]$ be a feature vector. At first, a binary division is done along each feature x_i $(i = 1, 2, \cdots, k)$, so we get a group of bi-subspaces, i.e., domains, denoted by $S(x_i, L, \Omega)$ and $S(x_i, R, \Omega)$, where L stands for the left domain, R for right domain, respectively. As described previously, instead of computing probability density $p(D|X)$, $\eta(X|D)$ is applied to describe recognition distribution therefore we define $\eta(x_i, L, \Omega|D)$ standing for density distribution of $S(x_i, L, \Omega)$, $\eta(x_i, R, \Omega|D)$ for density distribution of $S(x_i, R, \Omega)$. Among $k$divisions only one along the feature x_{i*} $(i* \in \{1, 2, \cdots, k\})$ is really selected to be executed, which is such that

$$\Delta\eta(x_{i*}, \Omega) = \max_i \Delta\eta(x_i, \Omega), \quad \cdots\cdots (12)$$

where

$$\Delta\eta(x_i, \Omega) = |\eta(x_i, L, \Omega|D) - \eta(x_i, R, \Omega|D)|. \quad \cdots\cdots (13)$$

Similarly for each subspace $S_u \in \{S(x_{i*}, L, \Omega), S(x_{i*}, R, \Omega)\}$ we can obtain its furthermore divided subspaces $S(x_i, L, S_u)$ and $S(x_i, R, S_u)$ by binary divisions. And the really executed division along the feature x_{i*} $(i* \in \{1, 2, \cdots, k\})$ at this step is also such that

$$\Delta\eta(x_{i*}, S_u) = \max_i \Delta\eta(x_i, S_u), \quad \cdots\cdots (14)$$

where

$$\Delta\eta(x_i, S_u) = |\eta(x_i, L, S_u|D) - \eta(x_i, R, S_u|D)|. \quad \cdots\cdots (15)$$

Apparently such a division might be carried out infinitely, producing countless domains therefore a termination condition should be added. Hereby, we introduce two thresholds: an integer σ (≥ 0) standing for a threshold of sampling points for a subspace S_u and a real number δ (≥ 0) for a threshold of the difference of density distribution between two subspaces of the subspace S_u. The binary division process will be stopped if

$$C(S_u) \leq \sigma || \Delta\eta(x_{i*}, S_u) \leq \delta, \quad (16)$$

where $C(S_u)$ is the sampling points of the subspace S_u and symbol || represents 'OR' Boolean operator.

The domain division for non-parametric distribution is equivalent to sampling problem in signal processing. An effective technique is that the higher density makes more divisions, vice versa. It is the threshold σ that determines how small a domain should be.

Furthermore, as we consider the problem of division of sampling space, distinguishing relevant and irrelevant features should be also taken in account. Clearly the model will become redundant if an irrelevant feature is involved. Therefore is it possible that irrelevant features can be kicked out when domains are divided?

It is clear that the times of binary division along the each feature x_i, denoted by $\kappa(x_i)$, might be different. And it can be applied to deal with the problem of elimination of irrelevant features. Before some conclusions are induced, the definitions of relevant and irrelevant feature are discussed as follows.

In reference (5), a basic definition of relevance is given using its relationship to so-called target.

[Definition 1] Relevant to target

A feature x_i is relevant to a target concept c if there exist a pair of examples A and B in the instance space such that A and B differ only in their assignments to x_i and $c(A) \neq c(B)$

In reference (5), definitions of the relevance in strong sense and in weak sense are also argued due to limitation of determining a relevant relationship using *definition 1*. To the contrary, we primarily define not the relevance but the irrelevance by means of statistics.

[Definition 2] Irrelevant feature in strong sense

A feature x_r is an irrelevant feature if decision distribution $p(D|x_r)$ is a uniform distribution and independent of other features.

Using above definition and the sampling division method, we obtain the following theorem.

[Theorem]

The times of binary division along a feature x_r is denoted by $\kappa(x_r)$. $\kappa(x_r) = 0$ if the feature x_r is irrelevant to operator's decision in strong sense.

[Proof]

Based on equations (2), (3), we get

$$\eta(X|D) = p(X|D)/p(X) = p(D|X)/p(D). \quad (17)$$

For the feature x_r, we have

$$\eta(x_r|D) = p(D|x_r)/p(D). \quad (18)$$

The distribution $\eta(x_r|D)$ should be uniform because $p(D|x_r)$ and $p(D)$ are uniform distributions, according to definition and assumption.

The uniform property is kept for all domains if a feature is independent of others, therefore the binary division on x_r for any subspace S_u is always such that

$$\Delta\eta(x_r, S_u) = 0. \quad (19)$$

But according to binary division method, only the binary division such that $\Delta\eta(x_i, S_u) > 0$ is possibly selected and really executed. Thus the binary division will be never really executed on x_r, i.e.,

$$\kappa(x_r) = 0. \quad (20)$$

[End]

However we cannot induce that a feature is irrelevant one in strong sense even if $\kappa(x_r) = 0$ using proposed binary division. Therefore we introduce the definition of irrelevant feature in weak sense as follows.

[Definition 3] Irrelevant feature in weak sense

A feature x_r is an irrelevant one if $\kappa(x_r) = 0$.

That is, we can eliminate irrelevant features in weak sense using binary division method.

4. Computation of Decision Acquisition

4.1 Acquisition of Decision Information

The decision acquisition distribution $\eta(X|D)$ cannot be computed from equation (3) directly because sampling space might change at any time due to alternation of jobs in the buffer. In this section, we address how to calculate information acquisition for every decision, $\eta_*(X|D)$.

Given a machine with its buffer, there are n jobs waiting in the buffer, whose attribute belongs to m discrete values, denoted by $\{\alpha_1, \alpha_2, \cdots, \alpha_m\}$. Corresponding to each value, the numbers of jobs is denoted by $C(\alpha_j)$, $(j = 1, 2, \cdots, m)$. The probability of selecting a job belonging to α_j can be easily calculated by

$$p(x = \alpha_j) = \frac{C(\alpha_j)}{n}. \quad (21)$$

Let the amount of acquisition be

$$g(x = \alpha_j|D) = \frac{1}{p(x = \alpha_j)} = \frac{n}{C(\alpha_j)}, \quad (22)$$

if the operator selects a job belonging to α_j.

Let the amount of loss be

$$g(x \neq \alpha_j|D) = \frac{1}{p(x \neq \alpha_j)} = \frac{1}{1 - p(x = \alpha_j)}, \quad (23)$$

if the operator selects a job not belonging to α_j.

In order to make the statistics in the same standard for all decisions, the amount of acquisition for the decision should be transformed and we get

$$\eta_*(x = \alpha_j|D) = \frac{g(x = \alpha_j|D)}{g(x = \alpha_j|D) + g(x \neq \alpha|D)} = 1 - \frac{C(\alpha_j)}{n}. \quad (24)$$

An extreme case is that $\eta_*(x = \alpha_j|D) = 0$ if $C(\alpha_j) =$

n. It means that all jobs belong to the same type therefore no acquisition information will be obtained. Conversely $\eta_*(x = \alpha_j|D) \to 1$ if $C(\alpha_j)/n \to 0$. It implies that selecting a job with very small probability will result in very large amount of acquisition of decision information.

Furthermore, we consider that the feature of n jobs is a continuous variable x. The feature value of job j $(j = 1, 2, \cdots, n)$ is represented by x_j. In the same way, we calculate the amount of acquisition and the amount of loss by

$$g(x = x_i|D) = 1/\int_{x_j-\Delta x}^{x_j+\Delta x} p(x)dx, \quad \cdots\cdots\cdots (25)$$

$$g(x \neq x_j|D) = 1/\left(1 - \int_{x_j-\Delta x}^{x_j+\Delta x} p(x)dx\right), \cdots (26)$$

where Δx is introduced to specify a tiny interval near x_j, if the operator selected the job j. Similarly the amount of acquisition for the decision this time can be obtained by

$$\eta_*(x = x_j|D) = \frac{g(x = x_j|D)}{g(x = x_j|D) + g(x \neq x_j|D)} = 1 - \int_{x_j-\Delta x}^{x_j+\Delta x} p(x)dx \cong 1 - 2p(x_j)\Delta x. \quad \cdots\cdots (27)$$

Finally the amount of acquisition for every decision is expressed in more general formula as follows:

$$\eta_*(X = X_j|D) = 1 - \int_{\Delta S} p(X)dX \cong 1 - p(X_j)\Delta V, \quad \cdots\cdots (28)$$

where ΔS stands for a division space around X_j, ΔV for the volume of ΔS in generalized sense because both discrete and continuous variables are included. Moreover, $p(X_j)$ is computed from X_j $(j = 1, 2, \cdots, n)$ because the distribution $p(X)$ should be induced from the current n jobs.

As illustrated in previous section, multi-dimensional distribution is generally hard to be obtained. But for the case that we try to construct the distribution using a small quantity of sampling data, we apply a simple method to approximate to it.

We regard each sampling point X_j as a l-dimensional normal distribution [11], which can be expressed by

$$p_j(X) = \frac{1}{l\sqrt{(2\pi)^l|\Sigma|}} \exp\{-\frac{1}{2}\Delta X_j^T \Sigma^{-1} \Delta X_j\}, \quad \cdots\cdots (29)$$

where Σ is a variance matrix, $\Sigma = diag[1 \ \ 1 \ \ \cdots \ \ 1]$, and $\Delta X_j = K^T(X - X_j)$. Hereby a scaling vector $K^T = [k_1 \ \ k_2 \ \ \cdots \ \ k_l]$ is introduced to avoid calculating the reverse matrix. Especially, we have $k_i = M$, (M is a large real number) if $x_k - x_{kj} \neq 0$, and $k_i = \log_e 2\pi$ if $x_k - x_{kj} = 0$ for a discrete variable x_k and the discrete value x_{kj} of job j. Consequently, the probability density at X_j can be obtained by

$$p(X_j) = \sum_{u=1}^{n} p_u(X_j). \quad \cdots\cdots (30)$$

The next question is calculation of the division space. If all variables are discrete ones, we can simply let $\Delta V = 1$. If both discrete and continuous variables are taken in to account, we have

$$\Delta V = \prod \lambda_s, \quad \cdots\cdots (31)$$

where λ_s is the basic statistical unit for the continuous feature variable x_s. For example, an operator may be sensitive to processing time minutely therefore the basic statistical unit for processing time is one minute.

At last, the decision recognition distribution takes the statistics of $\eta_*(X_j|D)$ for all jobs' decisions.

4.2 Manufacturing System Example A PCB (printed- circuit-board) manufacturing system with 29 manufacturing cells, each of which is composed of several machines is considered in this paper. Detailed description for each manufacturing cell is given in Table 1. All manufacturing cells are grouped into three stages according to processed parts type. Manufacturing cells W01~W07 belong to inner-layer stage, processing single boards. At W7 several single boards are pressed into a multi-layer board therefore cells W08~W25 belong to outer-layer stage. At W25 the multi-layer board is cut into several general PCB boards hence cells W26-W29 belong to PCB stage.

In Table 1, EDD stands for mounting a job according to early due-date, FIFO for first-in-first-out. The symbol "*" represents that the rules are modified because each job to be processed is attached by a priority to distinguish emergency job and ordinary job. It means that

Table 1. Description of a PCB line

Stage	Cell No	Cell Name	Rule
Inner Layer	W01	Lamination	EDD*
	W02	Etching	FIFO*
	W03	Oxide	EDD*
	W04	Verification	EDD*
	W05	Drying	FIFO*
	W06	Organization	FIFO*
	W07	Press	FIFO*
Outer Layer	W08	Face Shave	FIFO*
	W09	NC Drilling	EDD*
	W10	Scrubbing	FIFO*
	W11	No Electrolysis	FIFO*
	W12	Electrolysis	FIFO*
	W13	Belt Sander	FIFO*
	W14	Lamination	FIFO*
	W15	Through Hole Digging	EDD*
	W16	Etching	FIFO*
	W17	Tin Stripping	FIFO*
	W18	Oxide	EDD*
	W19	Verification	EDD*
	W20	Spray	EDD*
	W21	Exposure	FIFO*
	W22	Developing	FIFO*
	W23	Marking	FIFO*
	W24	Soldering	FIFO*
	W25	V Cutting	FIFO*
PCB	W26	Router	FIFO*
	W27	Cleaning	FIFO*
	W28	Test	EDD*
	W29	Inspection	EDD*

a job with higher priority will be processed earlier and jobs with the same priority will obey specified rule. In following section the priority is always involved when we concern with the rule-based results.

4.3 Data Analysis One-year's practical manufacturing data are used to verify our proposed method. To evaluate the effectiveness of decision recognition, an accuracy percentage is defined as follows:

$$a_R = \frac{C_R}{C_J} \times 100\%, \quad \cdots\cdots\cdots (32)$$

where C_J stands for all jobs passed through a manufacturing cell within a year, among them C_R for the jobs obeying recognition pattern such as the dispatching rules given in Table 1. The accuracy percentages based on rules given in Table 1 for all manufacturing cells are shown in Table 2. Unfortunately, most of obtained results using rules given in Table 1 show very poor performance. We are apparently unable to predict the future status of the PCB manufacturing system based on rules given in Table 1. It implies that the operators' decisions in the PCB manufacturing system are not consistent with management command and have their own disciplines. To try to determine whether it is possible to recognize the decision discipline based on other rules for each manufacturing cell, a wider investigation is done using current given parameters, which include time into buffer, due-date, processing time, current buffer length and priority. 6 rules are generated based on these parameters. They are

LIFO: The job that lastly enter into buffer will be mounted at first (Last-in-first-out),

FIFO: The job that firstly enter into buffer will be mounted at first (First-in-first-out),

LDD: The job with last due date will be mounted at first,

EDD: The job with earliest due date will be mounted at first,

LPT: The job with longest processing time will be mounted at first,

SPT: The job with shortest processing time will be mounted at first.

Noted that each rule should be combined with the priority as described previously, denoted by a superscript "*", when they are applied. Then all rule-based results are shown in Table 2.

In Table 2, the minimum and maximum quantities of jobs waiting in corresponding buffer show that the load for each manufacturing cell greatly fluctuates. And the total number of jobs, C_J, passing through every manufacturing cell within a year might be different. The accuracy percentage, a_R, indicates how many jobs obey the corresponding rule, where the datum with underline means that corresponding rule might be the best one for each manufacturing cell. To evaluate the efficiency of the most possibly employed rule in practice for each manufacturing cell, five level standards are proposed according to accuracy rate, i.e., *Good*: 80–100%, *Strong*: 60–80%, *General*: 40–60%, *Weak*: 20–40%, and *Worse*: 0–20%. As a result, we have following results shown in Table 3. Table 3 shows that no rule is *Good* enough to identify the operator's selection. Three manufacturing cells reveal that the corresponding rules are *Strong* and six manufacturing cells make use of rules in *General*

Table 2. Accuracy percentages based on single rule

Cell No	Jobs in Buf.		C_J	a_R					
	Min	Max		LIFO*	FIFO*	LDD*	EDD*	LPT*	SPT*
W01	1	72	11415	21.12	7.52	50.79	56.56	19.49	30.45
W02	2	70	11422	13.7	10.46	35.98	47.5	14.73	24.88
W03	1	171	20577	4.06	8.22	10.2	28.7	7.1	7.95
W04	1	163	20581	1.39	24.65	3.28	32.04	5.9	15.53
W05	1	113	20578	21.5	7.15	13.86	24.11	26.49	28.34
W06	1	102	20578	17.29	13.87	13.64	25.88	11.49	11.23
W07	1	72	20464	55.91	59.6	33.79	43.1	38.63	29.31
W08	1	94	20556	64.96	56.57	32.27	44.66	41.57	34.14
W09	1	300	20563	4.83	5.65	5.9	13.94	4.65	4.68
W10	1	192	25570	5.84	23.73	4.66	25.73	16.15	15.41
W11	1	101	25485	39.4	37.92	18.98	31.78	30.85	22.85
W12	1	136	25511	10.53	23.9	8.13	21.85	15.13	11.66
W13	1	80	25113	40.13	60.34	23.83	36.56	37.04	32.25
W14	2	69	16180	26.76	26.45	14.09	30.62	18.62	26.63
W15	1	79	16164	23.92	28.87	17.5	27.2	14.79	25.25
W16	2	110	16180	10.39	8.92	9.71	20.61	15.14	19.05
W17	1	77	4775	23.02	30.14	24.02	38.01	31.43	19.75
W18	1	118	18714	5.88	16.31	5.92	15.03	13.34	11.6
W19	2	162	25546	1.27	10.24	2.47	13.81	10.02	8.47
W20	2	217	25525	7.15	3.63	4.03	10.93	16.27	12.08
W21	1	122	25475	8.38	40.47	9.04	17.17	14.09	22.47
W22	1	62	25449	35.59	61.11	23.26	30.37	32.83	33.32
W23	1	300	25504	5.02	27.54	6.67	16.13	12.4	21.6
W24	2	112	21413	12.49	14.55	7.62	16.15	29.4	27.95
W25	1	75	18673	28.36	59.84	20.92	28.61	28.63	32.26
W26	1	106	25420	9.3	53.88	9.85	24.1	11.51	19.46
W27	1	61	24529	19.45	75.08	20.67	33.01	29.23	27.37
W28	1	170	25475	2.69	15.37	3.72	10.99	5.95	7.8
W29	4	258	25456	0.61	3.17	0.91	6.04	4.47	2.24

Table 3. The efficiency of rules

Cell No	a_R	Best Rule#	Level	Cell No	a_R	Best Rule#	Level
W01	56.56	EDD*	General	W16	20.61	EDD*	Weak
W02	47.5	EDD*	General	W17	38.01	EDD*	Weak
W03	28.7	EDD*	Weak	W18	16.31	FIFO*	Worse
W04	32.04	EDD*	Weak	W19	13.81	EDD*	Worse
W05	26.49	LPT*	Weak	W20	16.27	LPT*	Worse
W06	25.88	EDD*	Weak	W21	40.47	FIFO*	General
W07	59.6	FIFO*	General	W22	35.59	LIFO*	Weak
W08	64.96	LIFO*	Strong	W23	27.54	FIFO*	Weak
W09	13.94	EDD*	Worse	W24	29.4	LPT*	Weak
W10	25.73	EDD*	Weak	W25	59.84	FIFO*	General
W11	39.4	LIFO*	Weak	W26	53.88	FIFO*	General
W12	23.9	FIFO*	Weak	W27	75.08	FIFO*	Strong
W13	60.34	FIFO*	Strong	W28	15.37	FIFO*	Worse
W14	30.62	EDD*	Weak	W29	6.04	EDD*	Worse
W15	28.87	FIFO*	Weak				

#The best rule selected from Table 2 with underline

Table 4. The division times of features

Cell No	Total Domains	The Division Times $\kappa(x_i)$ x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
W01	385	99	51	32	--	69	75	28	30
W02	305	86	35	34	--	36	36	35	42
W03	348	81	36	40	33	37	33	50	37
W04	409	132	74	75	46	37	--	44	--
W05	397	104	57	61	51	19	17	46	41
W06	379	88	53	90	43	38	--	27	39
W07	302	103	32	44	25	29	20	24	24
W08	352	136	45	64	31	--	16	30	29
W09	309	65	57	61	33	54	--	--	38
W10	314	76	25	45	46	44	--	50	36
W11	374	78	52	69	50	49	30	28	17
W12	370	82	60	59	30	22	25	47	44
W13	318	87	54	43	32	25	12	30	34
W14	340	98	53	54	28	20	13	37	36
W15	368	90	41	52	36	34	30	45	39
W16	322	82	40	61	25	24	21	29	39
W17	360	75	45	54	30	54	37	35	29
W18	376	94	42	49	38	38	17	54	43
W19	317	58	47	47	31	49	38	6	40
W20	313	88	73	59	43	--	--	49	--
W21	365	113	59	94	44	--	--	35	19
W22	346	109	39	56	46	22	6	41	26
W23	317	87	45	51	30	29	12	23	39
W24	321	162	22	37	--	--	--	23	26
W25	356	106	46	65	35	73	--	--	30
W26	342	86	42	45	37	38	33	24	36
W27	321	94	30	31	24	42	36	23	40
W28	339	83	36	57	27	49	37	--	49
W29	387	65	64	56	38	72	61	--	33

sense. For other manufacturing cells, rules are *Weak* or *Worse* ones and can almost reflect nothing.

4.4 Prediction Enhancement In this section, we apply Bayesian decision theory and proposed method previously to recognize operators' decisions. For above PCB manufacturing system, we know following parameters:

h_1: The time when jobs enter the buffer;
h_2: The due date given by corresponding orders;
h_3: The processing time on each manufacturing cell;
s_1: The size of parts;
d_1: The priorities of jobs given by orders;
Therefore 8 features are formed as follows:

$$x_1 = h_1(j) - \min_k h_1(k), \quad \cdots (33)$$

$$x_2 = \max_k h_1(k) - h_1(j), \quad \cdots (34)$$

$$x_3 = h_2(j) - \min_k h_2(k), \quad \cdots (35)$$

$$x_4 = \max_k h_2(k) - h_2(j), \quad \cdots (36)$$

$$x_5 = h_3(j) - \min_k h_3(k), \quad \cdots (37)$$

$$x_6 = \max_k h_3(k) - h_3(j), \quad \cdots (38)$$

$$x_7 = \begin{cases} 0 & s_1(j) = s_1(j_0) \\ 1 & s_1(j) \neq s_1(j_0) \end{cases}, \quad \cdots (39)$$

$$x_8 = d. \quad \cdots (40)$$

Along these 8 features carry out the binary divisions proposed in Section 3.2. Hereby, the procedure always

Table 5. The enhancement results

Cell No	a_R Best Rule	New No.1#	New No.1 Level	Cell No	a_R Best Rule	New No.1#	New No.1 Level
W01	56.56	66.53	Strong	W16	20.61	29.93	Weak
W02	47.5	52.79	General	W17	38.01	59.42	General
W03	28.7	31.26	Weak	W18	16.31	25.52	Weak
W04	32.04	34.27	Weak	W19	13.81	19.82	Weak
W05	26.49	48.95	General	W20	16.27	23.36	Weak
W06	25.88	36	Weak	W21	40.47	39.87	Weak
W07	59.6	92.98	Good	W22	35.59	77.31	Strong
W08	64.96	87.91	Good	W23	27.54	32.71	Weak
W09	13.94	21.5	Weak	W24	29.4	38.9	Weak
W10	25.73	33.4	Weak	W25	59.84	70.94	Strong
W11	39.4	64.93	Strong	W26	53.88	65.75	Strong
W12	23.9	32.91	Weak	W27	75.08	73.15	Strong
W13	60.34	82.2	Good	W28	15.37	18.15	Worse
W14	30.62	51.13	General	W29	6.04	10.79	Worse
W15	28.87	49.11	General				

#Result based on Bayesian decision model, which consists of all relevant features in Table 4 for each manufacturing cell.

continues if the sampling points are more than $\sigma = 10$, although the maximum difference for binary divisions along all features is very small. That is $\delta = 0$. Among above divisions, we take the statistics of ones such that difference of decision acquisition of two divided child-domains is larger than 3% that of parent domains. A feature is regarded as an irrelevant one if the division times equals 0. Those irrelevant features are deleted and binary division process is done once again. Finally, the total domains and the division times for all features are given in Table 4, where symbol "--" represents that corresponding feature is irrelevant and unnecessary.

Next, we test the model using the one-year's jobs given in Section 4.2. Predicted new results are given in Table 5. Three manufacturing cells fall in *Good* level and six appear in *Strong* sense.

Considering the operator's behavior, we think the blocking degree of buffer, the quantity of jobs waiting in buffer, might influence the manner of decision. Therefore a ratio, describing the blocking degree, is introduced by

$$\rho = \log \frac{n_c}{n_{\max}}, \quad \cdots (41)$$

where n_c stands for current quantity of jobs, $n_{\max}$ for "Max" jobs in the buffer given in Table 2. Three models are built for $\rho \in (-\infty, -0.6)$, $\rho \in [-0.6, -0.3)$, and $\rho \in [-0.3, 0]$. The predicted results are given in Table 6, where the *Good* recognitions are increased by 2 the *Strong* ones by 1 and the *Worse* ones disappeared.

The results considering buffer length in Table 6 are furthermore improved and the final results have been greatly enhanced compared with the best-rule-based results. However, it is still difficult to precisely predict the future status for many manufacturing cells. We think there are three factors determining the feasibility of prediction. First, the operator's behavior should be stable during past recognition period and the future prediction. Second, the manufacturing surrounding data should be sufficiently provided. The accuracy will be greatly decreased if some of important parameters

Table 6. The enhancement results using buffer length

Cell No	a_R		New No.2 Level	Cell No	a_R		New No.2 Level
	Best Rule	New No.2#			Best Rule	New No.2#	
W01	56.56	77.03	Strong	W16	20.61	34.09	Weak
W02	47.5	69.7	Strong	W17	38.01	61	Strong
W03	28.7	35.43	Weak	W18	16.31	29.8	Weak
W04	32.04	39.9	Weak	W19	13.81	36.38	Weak
W05	26.49	67.5	Strong	W20	16.27	42.77	General
W06	25.88	41.29	General	W21	40.47	55.26	General
W07	59.6	90.1	Good	W22	35.59	88.75	Good
W08	64.96	92.9	Good	W23	27.54	33.93	Weak
W09	13.94	25.8	Weak	W24	29.4	48.4	General
W10	25.73	43.77	General	W25	59.84	88.85	Good
W11	39.4	79.51	Strong	W26	53.88	61.79	Strong
W12	23.9	47.28	General	W27	75.08	81.2	Strong
W13	60.34	90.7	Good	W28	15.37	25.4	Weak
W14	30.62	54.44	General	W29	6.04	20.76	Weak
W15	28.87	59	General				

#Results based on Bayesian decision model considering buffer length

cannot be gathered. Third, the recognition method should be powerful to identify the factors relevant to human's decision.

5. Conclusion

The decision mechanism for manufacturing system using Bayesian thinking is discussed in this paper. A distribution of decision recognition is firstly given and transformation of the feature vector is also addressed. Then a non-parametric model is employed to describe distribution, where a binary division methodology is developed to limit the size of the model, making it possible to eliminate irrelevant features as well. Finally, a PCB manufacturing system is given to demonstrate the methodology proposed in this paper. Obtained results show that prediction precision can be greatly improved.

(Manuscript received Oct. 28, 2002,
revised June 6, 2003)

References

(1) J.O. Berger: Statistical Decision Theory and Bayesian Analysis, Springer-Verlag (1985)

(2) A.K. Jain, R.P.W. Duin, and J. Mao: "Statistical Pattern Recognition: A Review", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol.22, No.1, pp.4–37 (2000-1)

(3) J.N. Hwang, S.R. Lay, and A. Lippman: "Nonparametric Multivariate Density Estimation: A Comparative Study", *IEEE Trans. SP*, Vol.42, pp.2750–2810 (1994-10)

(4) A.G. Gray and A.W. Moore: "'N-Body' Problem in Statistical Learning", in Advances in Neural Information Processing System 13, pp.521–527 (2001-5)

(5) A.L. Blum and P. Langley: "Select of Relevant Feature and Examples in Machine Learning", Artificial Intelligence, pp.245–271 (1997)

(6) J.M. Bernardo and A.F.M. Smith: Bayesian Theory, John Wiley (1994)

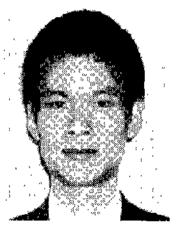
(7) P.E. Lassila and J.T. Virtamo: "Nearly optimal importance sampling for Monte Carlo simulation of loss system", ACM Trans. on Modeling and Computer Simulation, Vol.10, No.4, pp.326–347 (2000-10)

(8) B.W. Silverman: Density Estimation for Statistic and Data Analysis, Chapman and Hall, London (1986)

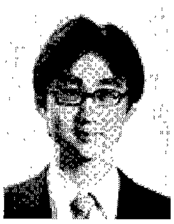
(9) G. Terrell and D. Scott: "Variable Kernel Density Estimation", *Ann. Statistic*, Vol.20, No.3, pp.1236–1265 (1992-3)

(10) T.K. Ho: "Nearest Neighbors in Random Subspaces", Lecture Notes in Computer Science: Advances in Pattern Recognition, pp.640–648 (1998)

(11) R.A. Jonson and D.W. Wichem: "Applied Multivariate Analysis", 4th Ed., Pretice-Hall (1998)



Yang Jianhua (Student Member) was born in Jiangsu province, China, in 1968. He received the B.S. degree from South-east University, Nanjing, China in 1990 and the M.S. degree from Tsinghua University, Beijing, China in 1993. He had worked in Automation Department of Tsinghua University from 1993 to 2000. He has been a doctorial candidate of the Department of Electrical and Computer Engineering, Yokohama National University since 2001. His research interests include flexible manufacturing system, shop-floor control and Petri Nets. Mr. Yang is a student member of the Institute of Electrical Engineers of Japan.



Yasutaka Fujimoto (Member) was born in Kanagawa prefecture, Japan, in 1971. He received B.E., M.E., and Ph.D. degrees in electrical engineering from Yokohama National University, Japan, in 1993, 1995, and 1998, respectively. In 1998, he joined the Department of Electrical Engineering, Keio University. Since 1999, he has been with the Department of Electrical and Computer Engineering, Yokohama National University, where he is currently an Associate Professor. His research interests include manufacturing automation, discrete event systems, motion control, and robotics. He is a member of IEEE and Robotics Society of Japan.