Investigating the Effectiveness of Conditional Classification: An Application to Manufacturing Scheduling

Alok R. Chaturvedi and Derek L. Nazareth

***Abstract*— This paper examines the problem of multidimensional classification, an automated learning process where "rules" are to be inferred on separate but related aspects of a problem, using identical or overlapping data sets. A general framework describing the various types of multidimensional classification is provided. The paper specifically concentrates on conditional classification, wherein the order of classification is based on domain semantics. Drawing from concept learning and information theory, algorithms are presented for acquiring tree-structured knowledge from available data. An application to manufacturing scheduling is presented. Results indicate that conditional classification may provide some ability to better interpret related decisions in automated manufacturing contexts. Further work is necessary to ascertain if the approach is robust, particularly on more complex decisions, larger data sets, and noisy data.**

I. Introduction

KNOWLEDGE-BASED SYSTEMS (KBS) are finding increasing use as viable options for problems involving human judgment. These systems operate using a set of knowledge that captures the logic needed to address the problem at hand. This knowledge may be represented in a variety of formats including production rules, semantic nets, decision trees, frames, and the like. Most KBS's are built around a core set of knowledge that covers the bulk of cases presented to the system. With time, this knowledge is updated and enhanced to cover additional cases, rectify problematic cases, and expand the scope of the system. The sources of this knowledge include recognized human experts, documented procedures, observations, among others. For the most part, human experts are the preferred source as they provide immediate access to knowledge, reasoning concerning its usage, and mechanisms for rectifying noticeable errors and gaps. A variety of mechanisms have been devised to extract this knowledge from human experts [2]. In some domains human experts may be hard to find, or if available, may possess expertise in only a portion of the domain. The scheduling of manufacturing systems is a good example of this. It has been argued that there are few, if any, recognized scheduling experts [12]. Even if experts are available, it is expected that they would be familiar with a limited set of manufacturing configurations and operating conditions, and are likely to perform below expectations when faced with unfamiliar problems. In such situations, the availability of alternative knowledge sources assumes vital importance to the construction and maintenance of KBS's. Given the paucity of human scheduling experts, the availability of universal scheduling procedures that can be codified is an unlikely prospect. Therefore, creation of a scheduling knowledge base will be data driven to some extent, drawing from basic structural knowledge from the experts coupled with information on historical performance of a scheduling system. Integration of this knowledge requires the extraction of meaningful "rules" concerning the functioning of the domain, drawing from a set of robust observations, coupled with some general induction principles. This process of autonomous knowledge extraction has been termed machine learning.

Several different forms of machine learning have been devised. In general, they can be differentiated on the basis of the learning principle involved, and the knowledge structure created. Learning strategies available include learning by being told, learning by example, learning by observation, learning by analogy, learning by explanation, learning by discovery, connectionist learning, genetic learning, and the like [7]. Knowledge representations employed include neural networks, predicate calculus, decision trees and tables, semantic nets, frames, deep domain models, plans, and scripts, among others [6]. A rather popular inductive strategy is concept learning, which is a form of learning by example, wherein learning occurs in a supervised mode, and the acquired knowledge is coded in the form of a decision tree. This method has been implemented in several algorithms, most notably ID3 [19], and C4 [22]. It is a simple yet elegant approach to discovering rule patterns based on generalizable concepts that have little intrinsic semantic meaning. However, its elegance, ease of implementation, and general accuracy make it a popular form of machine learning. There are currently several commercial products that implement this basic algorithm for several computer platforms, including First Class Fusion$^{(TM)}$ from Programs in Motion, Rulemaster$^{(TM)}$ from Radian, and Michie's Expert-Ease$^{(TM)}$.

Concept learning is based on information content (in an information theoretic sense) of subsets formed from a training observation set based upon externally specified concepts, and

Manuscript received October 23, 1992. Review of this manuscript was processed by Editor S. S. Erenguc.

A. R. Chaturvedi is with the Krannert Graduate School of Management, Purdue University, West Lafayette, IN 47907.

D. L. Nazareth is with the University of Wisconsin-Milwaukee, School of Business Administration, Milwaukee, WI 53201 USA.

IEEE Log Number 9400868.

0018–9391/94$04.00 © 1994 IEEE

can be traced back to work by Hunt [10], [11]. It has been applied to several problems including chess end-game strategies [19], assessing credit requests [5], selection of statistical significance tests [15], among others. The overriding assumption is that the recommendation to be made is characterized by a single dimension. In the chess end-game example, the dimension of interest is the number of moves needed to capture the opposing king; in the statistical system, the output dimension is the statistical test that is most appropriate for comparing the means of two populations. While this approach may be useful for problems of involving a single decision, it becomes problematic when recommendations are to be made about several aspects of the system, either jointly or in sequential fashion. Under these circumstances an alternative approach is warranted. Using the manufacturing scheduling example, as part of the scheduling process, it is expected that decisions need to be made regarding the release of new orders into the system as well as the assignment of individual workpieces to available workstations. Though related, these are clearly different decisions. If concept learning principles were to be applied to these decisions separately, the result would represent two independent decision trees to be used at different points by the KBS. However, any attempt to view these two decisions as being strictly independent is problematic, in that an incomplete view of the problem is adopted. Correspondingly, some manner of integrating the two decision outcomes during the induction process is necessary. One approach would be to view these decisions jointly, i.e., for a specific scenario (described by batch characteristics and cell status), the preferred decision strategies for release and assignment need to be identified as a dichotomous pair. This approach involves an assumption of probabilistic independence between the two decisions, which is clearly not the case. A more likely scenario involves making a recommendation on the release decision first, and using that resulting information in conjunction with batch and cell information to make the assignment decision. This situation is characterized as a process of conditional classification, whereby observations are first classified on one dimension, and that information is then employed to classify the observations on subsequent dimensions.

The paper discusses the importance of conditional classification in the inductive process. It outlines the limitations of independent or single dimension classification as a means of deriving knowledge when classification is necessary on multiple dimensions. A set theoretic formulation of the conditional classification problem is presented. Information content measures, *viz.* entropy are introduced. These are subsequently adapted to deal with the problems of joint and conditional classification. Different forms of conditional classification are outlined. An algorithm is developed for the situation where semantic knowledge is available for selecting sequence of the output dimensions for classification. The application of this algorithm to a problem of manufacturing scheduling is described. Implications of concept-based conditional classification round out the paper.

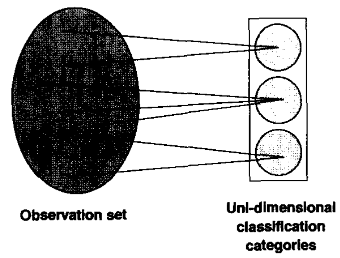


Fig. 1. Simple classification.

II. Multidimensional Classification

Multidimensional classification represents a variant of traditional machine learning using semantic-free concepts. It allows for the creation of multiple sets of training rules for several output dimensions using the same input data. While it may be possible to employ different learning strategies for different dimensions, this paper considers the use of the same inductive mechanism for the multiple dimensions of interest. Several forms of multidimensional classification are possible, some of which are described here. A more complete treatment of multidimensional classification is available in [18].

This paper is concerned with concept-based conditional classification. It should be stressed that conditional classification involves a definite sequence of output dimensions for classification. Thus it is different from independent, joint, and sequential classification methods. These differences are illustrated through a schematic representation of the classification types. *Simple classification* involves the mapping of observations characterized by several dimensions onto a single output dimension, as illustrated in Fig. 1. Examples of simple classification include diagnosis of soybean plant diseases based on observable symptoms [13], credit card assessment [5], and classification of mushrooms as poisonous or not based on physical characteristics [23], [25]. The ID3 algorithm is representative of simple classification.

It is possible to use the same observations to classify data on more than one dimension, e.g., manufacturing release and assignment decisions, based on batch and cell status information. If the classification exercise is performed using the same algorithm multiple times on the identical observations, but employing different output dimensions for classification, the learning strategy is termed *independent classification*, and is illustrated in Fig. 2. While independent classification is possible in principle, it is unlikely that the two different classification dimensions will involve the same input dimensions for characterization, and yet be totally independent of each other as far as the resulting decision tree is concerned. Nonetheless, this situation is presented for completeness of the analysis. An example of independent classification would be a situation where an applicant's demographic, scholastic, and financial information was used to independently classify applicant eligibility for admission and financial aid.

Given the possible relationships that can exist between two or more output dimensions for classification, it can be

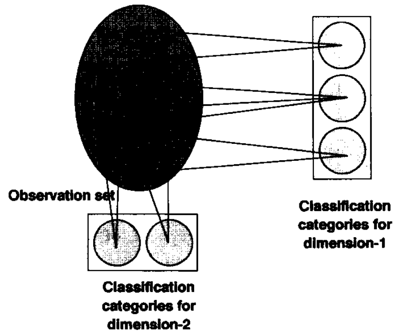


Fig. 2. Independent classification.

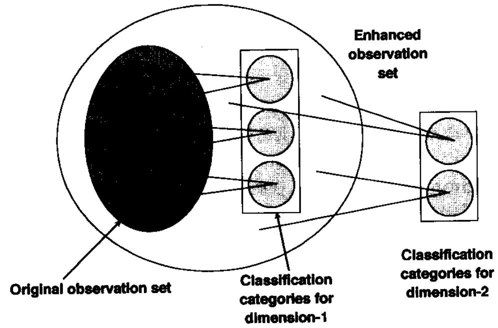


Fig. 4. Sequential classification.

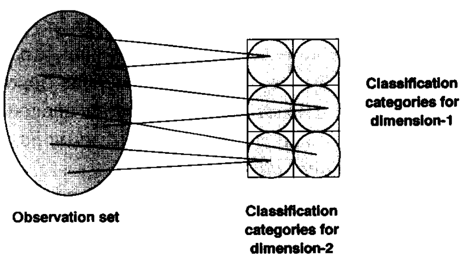


Fig. 3. Joint classification.

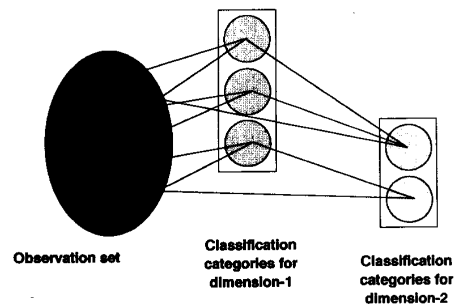


Fig. 5. Conditonal classification.

speculated that the relationship may be viewed as probabilistically independent, but correlated. In this case, the classification is to be performed for both dimensions jointly, as in classification of restaurants into joint categories of quality (good, fair, poor) and cuisine (French, American, Oriental, etc.) simultaneously. This essentially involves treating the number of possible outcome categories as a multiple of the outcome categories for each outcome dimension, and then employing simple classification strategies. This strategy is termed *joint classification*, and is illustrated in Fig. 3. Note that joint classification will produce a single decision tree, unlike independent classification, which produces multiple decision trees.

Though joint classification would appear to be a reasonable strategy from a probabilistic point of view, it remains problematic for several reasons. First, it is dependent on the number of output dimensions and their individual categories. This is subject to combinatorial explosion problems. Second, a number of the resulting (cross) categories may not have observations that can be mapped to them. Worse still, the cross categories may be meaningless, particularly if the decisions are marginally related. Lastly, the lack of semantic information concerning the cross categories makes for a less meaningful learning exercise. A more likely scenario involves the use of the results from one classification to guide the classification for subsequent dimensions. This approach is termed *sequential classification*, and is illustrated in Fig. 4. In this case the actual output categories are aggregated with the existing input classification dimensions and a subsequent simple classification exercise is performed. If the second decision can be shown to be dependent on the first, then this approach may be selected. An example of sequential classification in a medical context involves the diagnosis of possible infection and the selection of subsequent confirmatory testing strategy based on patient symptoms.

Though sequential classification has the attraction of being able to use the output of a prior decision for subsequent decision making, it does require the indiscriminate use of earlier input classification dimensions. This may be problematic, particularly if it can be argued that the input dimensions for the subsequent decisions need to be different from that of the prior decisions. Likewise, the importance of the prior decision is now downplayed considerably as it forms just one of the inputs to the next stage. It is suggested that a different approach be adopted for such situations, wherein the importance of the overall classification from the first stage play a significant role in subsequent classifications. This involves the use of conditional probabilities for the second classification given the prior probabilities for the first classification. This approach is termed *conditional classification*, and is illustrated in Fig. 5. Diagnosis of possible infections and prescription of appropriate medication, based on patient symptoms and characteristics, is representative of a conditional classification situation.

Conditional classification requires a different approach to computing the information content than does simple, independent, joint, or sequential classification. Also, unlike the prior approaches, where the same process can be used for different classification stages, conditional classification requires different process to be adopted during different stages.

III. Conditional Classification of Discrete Data

This section covers the problem of conditional classification using observations that are specified using discrete scales. Different forms of conditional classification that are expected to be encountered are discussed. The concept of joint classification is also introduced, though not employed here. The notation employed for computing information content of the respective classification stages is then outlined. In this paper, information content is specified in terms of entropy, drawing from work by Quinlan [19]. Other measures have been employed for classification including chi-square statistics [9], [14], G-statistics [16], and impurity measures in the GINI function [4], among others. While some studies have shown that these measures perform differently in terms of the size of the resultant tree, with information theoretic measures producing more parsimonious trees [3], [17], the overall accuracy of these measures appears to be comparable [17]. Given the wide use and relative accuracy of the entropy-based measures, they formed the basis for the creation of conditional classification algorithms in this paper. Formulae for computing entropy and marginal entropy in the single dimensional classification are provided. The concept of joint entropy for simultaneous multiple classification is also introduced. Mechanisms for computing conditional entropy, where classification is to be performed sequentially on several dimensions are also included. Finally, an algorithm is presented for conditional classification when conceptual (or semantic) information is available.

A. Forms of Conditional Classification

Conditional classification represents a situation where induction is performed on the output classification dimension in a well defined sequence. The selection of the first output dimension to classify on could be based on semantic information relevant to the classification task. This form of classification is termed *concept-based conditional classification*, as it is driven by externally provided semantic concepts. In some situations, the sequencing of classification dimensions may be specified by structural or temporal constraints. An example is a flexible manufacturing case, where the release classification necessarily precedes the assignment classification task. Not only is there an explicit temporal relationship, but in this case the assignment decision is affected by the release decision. This assumes that scheduling is being performed with minimal lookahead, i.e., decisions are made based on the prevailing state of the system. Alternative scheduling practices may choose to set aside such assumptions, as in cases where an entire machine sequence is identified prior to the release of the workpiece into the system. This study concentrates on scheduling practices that are based on the primacy of decisions, wherein every decision is evaluated based upon the current state of the system. This approach provides greater flexibility in the scheduling process, at the potential expense of greater scheduling overhead.

In case no semantic knowledge is available, the selection of the first dimension to classify upon can be driven by information theoretic methods. These strategies generally classify on the basis of intrinsic information content (usually expressed as entropy). Several strategies are available, including optimal selection methods (branch and bound algorithms), heuristic selection methods (best-first, hill-climbing), and exhaustive selection methods (breadth-first, depth-first, random selection). Candidate selection criteria include parsimony of the resulting decision tree, and the greatest marginal improvements in classification at each stage. Stopping rules are guided by complete classification in the case of optimal methods, and prespecified maximal classification levels for other methods. This form of classification is termed *content-based conditional classification.*

It is also possible that the expert or decision maker may elect to specify a sequence for selection of output dimensions for classification. The sequence may be guided by prior experience, personal knowledge, or individual preference. While this may not necessarily be extremely scientific, it serves as a mechanism for direct interaction on the part of the decision maker, thereby providing him or her the opportunity to explore the problem space and gain a better understanding of the decision task. This form of classification is termed *user-specified conditional classification.*

This paper restricts itself to concept-based conditional classification, as it best fits the problem at hand. Algorithms for concept-based conditional classification are developed. Information measures for independent, joint, and sequential classification are also presented.

B. Notation

The notation employed in this paper follows a set theoretic notation.

Let

X_i be an input dimension for classification, $i = 1, 2, \cdots, n$

C_j be an output dimension for classification, $j = 1, 2, \cdots, m$

x_{ip} be the possible states (values) for X_i, $p = 1, 2, \cdots, a_i$

c_{jq} be the possible states (values) for C_j, $q = 1, 2, \cdots, b_j$

T be the set of training observations

S be a subset of T, $S \subseteq T$

$f(S, c_{jq})$ be the fraction of S that belongs to c_{jq}

Likewise, let

$f(S, c_{j_1q_1}c_{j_2q_2})$ be the fraction of S that belongs to $c_{j_1q_1}$ and $c_{j_2q_2}$ jointly,

or more generally,

$f(S, c_{j_rq_r r=1,\cdots,m})$ be the fraction of S that belongs to $c_{j_1q_1}, c_{j_2q_2}, \cdots, c_{j_mq_m}$ jointly.

The information content of any subset S of T can be expressed as the entropy (H) of that set. In general, the entropy of a set is given by

$$H_y(S) = -\sum_y L(y) \tag{1}$$

where

$$L(y) = \begin{cases} -p(y) \cdot \log_2(p(y)) & \text{if } 0 < p(y) \leq 1 \\ 0 & \text{if } p(y) = 0 \end{cases} \tag{2}$$

where $p(y)$ represents the proportion of observations that are classified into distinct subsets on dimension y. In the interests of consistency with traditional information theory, the formula used for entropy will be represented as

$$H_y(S) = -\sum_y p(y) \cdot \log_2(p(y)) \tag{3}$$

with the understanding that in case $p(y) = 0$, then a value of 0 will be used in the entropy computation. The use of 2 as the base for the logarithm stems from early work by Shannon [24] and is retained for consistency. The use of any other base would yield similar results.

In general, the desirable properties of an measure of entropy include symmetry, expandability, decisivity, additivity, recursivity, among others. It has been shown that entropy measures based on Shanon's original formulation possess these properties [1]. Additionally, entropy measures should also be measurable, monotonic, and stable [8], which also holds for the basic measure adopted here. Using the notation developed here, entropy is defined for a subset S of T, based upon a classification dimension C_j, and is given by

$$H_j(S) = -\sum_{q=1}^{b_j} f(S, c_{jq}) \cdot \log_2(f(S, c_{jq}))\ j = 1, 2, \cdots, m \tag{4}$$

When a specific dimension X_i is used for classification, then the resulting entropy is given by

$$H_j(S, X_i) = -\sum_{p=1}^{a_i} f(S \mid X_i = x_{ip}) \cdot H_j(S \mid X_i = x_{ip}) \tag{5}$$

where

$$H_j(S \mid X_i = x_{ip}) = -\sum_{q=1}^{b_j} f(S \mid X_i = x_{ip}, c_{jq}) \times \log_2(f(S \mid X_i = x_{ip}, c_{jq}))i = 1, 2, \cdots, n \tag{6}$$

Selection of a dimension X_i is based on the minimum overall entropy, i.e.

$$X_i^* = \left\{X_i \middle| \min_i [H_j(S, X_i)]; i = 1, 2, \cdots, n\right\} \tag{7}$$

At subsequent stages, dimensions already used for classification are excluded from the set of candidate dimensions available. The process continues recursively until no further classification is possible, i.e.,

$$H_j(S, X_i) = 0 \tag{8}$$

Drawing from information theory, joint entropy can be defined assuming classification on output dimensions $c_1, c_2, \cdots, c_m$ simultaneously. This is defined for the dual case first, and then extended to the multidimensional case.

$$H_{j_1, j_2}(S) = -\sum_{r=1}^{2} \sum_{q_r=1}^{b_{j_r}} f(S, c_{j_r q_r r=1,2}) \cdot \log_2(f(S, c_{j_r q_r r=1,2})) \tag{9}$$

More generally, this can be expressed as

$$H_{j_1 j_2 \cdots j_m}(S) = -\sum_{r=1}^{m} \sum_{q_r=1}^{b_{j_r}} f(S, c_{j_r q_r r=1,\cdots,m}) \times \log_2(f(S, c_{j_r q_r r=1,\cdots,m})) \tag{10}$$

The procedures for computing entropies for selection of a dimension X_i and the stopping rule remains the same as in the independent case, except that the joint entropy formula is now employed.

In cases when classification has already been performed on one output dimension, e.g., z, it is possible to determine conditional entropy, for classifying other on another output dimension, e.g. y. Information theoretic formulations of conditional entropy generally take the form

$$H_{y|z}(S) = -\sum_y \sum_z p(y, z) \cdot \log_2(p(y \mid z)) \tag{11}$$

This can be specialized to the problem at hand, in which case the entropy is given by

$$H_{j_2|j_1}(S) = -\sum_{r=1}^{2} \sum_{q_r=1}^{b_{j_r}} f(S, c_{j_r q_r r=1,2}) \times \log_2\left(\frac{f(S, c_{j_r q_r r=1,2})}{f(S, c_{j_1 q_1})}\right) \tag{12}$$

This formula can be extended to m dimensions quite easily to compute $H_{j_m|j_{m-1},\cdots,j_1}(S)$.

As with the case of single output dimension classification, the procedures for computing entropies for selection of a dimension X_i and the stopping rule remains the same, except that the formula for conditional entropy is now used.

C. Concept-Based Conditional Classification(C^3) Algorithm

The algorithm for concept-based conditional classification (C^3) parallels that developed for simple classification as embodied in ID3, C4, and the like [19], [20], [22]. Given the expanded notation employed in this paper, algorithms will be presented for simple, joint, and conditional classification.

Simple Classification Algorithm (SC):

0. $S \leftarrow T$
1. Compute $H_{j_1}(S)$
2. If $H_{j_1}(S) = 0$
    stop
   Else
    $(\forall X_i, i = 1, 2, \cdots, n)$, X_i not selected before
     Compute $H_{j_1}(S, X_i)$
     Select $X_i^* = \{X_i | \min_i [H_{j_1}(S, X_i)];\ i = 1, 2, \cdots, n\}$
     $\forall p$ in $x_{ip}, p = 1, 2, \cdots, a_i$
      $S \leftarrow (S \mid X_i = x_{ip})$
      Go to 1

The algorithm is simple, elegant, and is generally amenable to complete automation. However, it may encounter some difficulty under certain circumstances for the selection of X_i^*, if the entropy values are tied. In this situation, ties need to be resolved appropriately. Several strategies can be employed for resolving ties, including a first-come-first-select strategy, a random strategy, or a strategy based on semantic input from the decision maker. Quinlan has advocated a strategy based on probabilities of further classification [21]. It should be pointed out that the strategy used for the resolution of ties has a definite impact on the structure of the decision tree, and hence affects the efficiency and perhaps effectiveness of classification. For the moment, it is assumed that a reasonable strategy is available for breaking ties. Section IVA. lists available strategies for the application, and discusses implications of selecting a specific strategy.

The algorithm for joint classification is identical to that for simple classification except that joint entropy is used instead of the standard entropy measure. The algorithm is presented for two output dimensions but can be easily extended to m output dimensions.

Joint Classification Algorithm (JC):

0. $S \leftarrow T$
1. Compute $H_{j_1,j_2}(S)$
2. If $H_{j_1,j_2}(S) = 0$
   stop
   Else
   $\forall X_i, i = 1, 2, \cdots, n,\ X_i$ not selected before
   Compute $H_{j_1,j_2}(S, X_i)$
   Select $X_i^* = \{X_i | \min_i [H_{j_1,j_2}(S, X_i)]; i = 1, 2, \cdots, n\}$
   $\forall p$ in $x_{ip}, p = 1, 2, \cdots, a_i$
   $S \leftarrow (S | X_i = x_{ip})$
   Go to 1

The algorithm for concept-based conditional classification is presented for two output dimensions only, though it is easily extensible to multiple dimensions. The algorithm is based on that for simple classification and is relatively simple. Classification on the first output dimension is performed using the simple classification algorithm, SC. Subsequent classification employs a measure of conditional entropy, and is presented as algorithm CC.

Concept-Based Conditional Classification Algorithm(C^3):

0. Let C_a, C_b represent the dimensions to be classified in the order specified.
1. Apply SC using C_a, i.e. compute $H_a(S), H_a(S, X_i)$, etc.
2. Apply CC using C_b, i.e. compute $H_{b|a}(S), H_{b|a}(S, X_i)$, etc.

Conditional Classification Algorithm (CC):

0. $S \leftarrow T$
1. Compute $H_{j_2|j_1}(S)$
2. If $H_{j_1|j_2}(S) = 0$
   stop
   Else
   $\forall X_i, i = 1, 2, \cdots, n,\ X_i$ not selected before
   Compute $H_{j_2,j_1}(S, X_i)$
   Select $X_i^* = \{X_i | \min_i [H_{j_2,j_1}(S, X_i)]; i = 1, 2, \cdots, n\}$
   $\forall p$ in $x_{ip}, p = 1, 2, \cdots, a_i$
   $S \leftarrow (S \mid X_i = x_{ip})$
   Go to 1

It should be noted that the observations at the leaf level for conditional classification will be different than those for simple and joint classification. In the case of simple or joint classification, the stopping rule ensures that the observations at the leaf level share the same value of the output dimension. In the case of conditional classification though, the observations at the level may possess different values for the first output dimension, but these will map to consistent values for the second output dimension.

IV. Conditional Classification for Manufacturing Scheduling

The utility of concept-based conditional classification is investigated within the context of manufacturing scheduling. The problem involved the establishment of twin goals of production rate and workstation utilization, based on the manipulation of several controllable variables. Given that there is a tangible relationship between these goals, it follows that concept-based classification is more appropriate than context-based or user-defined conditional classification.

Observations for the classification algorithm C^3 were generated by a simulation model. A fraction of this data was employed to generate the decision trees, and the balance holdout set was used to test the effectiveness of the classification algorithm. The next few segments describe the problem setting, the simulation model, the decision tasks, and the experimental conditions in some detail.

A. The Problem Setting

The simulation models a European factory that is engaged in the production of industrial equipment. It produces a wide variety of workpieces from eight major part families. The production quantities and probability of occurrence determine the manufacturing mission for the system. The physical layout of the plant is shown in Fig. 6. The material flow in the system can be traced from the delivery of castings from vendors. These casting are delivered via the external material handling system (EMHS) to a palletizing work area, where they will be transferred to the stacker crane interface. Castings remain in the storage rack until required for production. At this stage, they are fetched by the stacker crane, taken to the material handling system (MHS) interface, and transported to the shuttle of the appropriate workstation. After the operations are completed, material is routed back via the MHS to the stacker crane, which places it back on the rack.

1. The Simulation Model: The simulation model of the system is implemented in MAST in a PS/270 environment. An overview of the model is presented here. For further details, the reader is referred to [7]. The model comprises two stacker cranes, four automatic guidance vehicles (AGV), four workstations, and three storage racks. One stacker crane serves the EMHS through transfer of raw castings to the system and finished products to storage. The other handles the movement of fixtures, castings, and workpieces from storage racks to the MHS interface. The MHS component of the system also encompasses turnout facilities so at to prevent blocking during loading and unloading at a workstation. Individual AGV's can carry a single workpiece at a time. The AGV's interface with workstations via a rotary shuttle that accommodates up to two palletized workpieces. Workstations in the system comprise a vertical machine center, two horizontal machine centers, and a measuring system. Individual storage racks are used

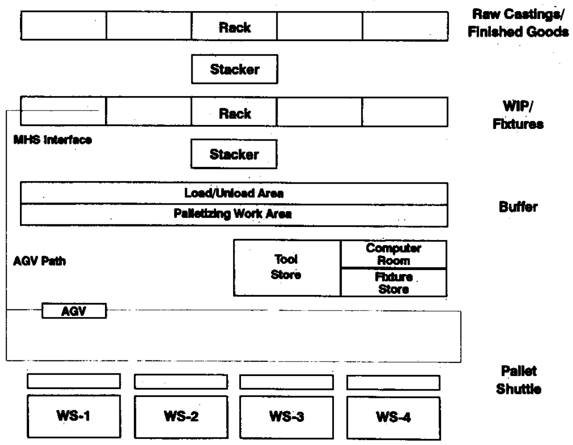


Fig. 6. Physical layout of the FMS.

for castings, finished goods, fixtures, WIP, and also serve as a buffer for the palletizing work area. All stackers are pass-through, enabling loading and unloading from either side.

For simplicity, only the major entities and their attributes are described in this section. The driving function for the model is the arrival of orders. The corresponding entity is *BATCHES*, which consists of a number of *WORKPIECES*, and their technological processing requirements, given by a *ROUTING*. The routing is an ordered linear list of *OPERATIONS* needed to transform a raw casting into a finished product. The operation specifies the type of *WORKSTATION* on which it is to be performed and the duration of the operation. The MHS comprises several entities— *STACKER*, *RACK*, and AGV. The stacker crane serves the racks, while the AGV's move material between the MHS interface and the respective workstations. In addition, each workstation interfaces with the AGV through a *SHUTTLE*.

All stacker operations occur between the MHS interface and the rack, with fetch and store times specified from computed averages. The prioritization of stacker resources for fetch and store tasks presents a problem though. If the stacker is idle, then any request for service, either load or unload, would be responded to immediately. If another request arrives during the execution of the activity, then it would be queued up, and processed upon completion of the first task. However, when multiple tasks are queued, a first-come-first-serve approach would be clearly inappropriate. For example, if two AGV's arrive to unload workpieces, and while the first is being unloaded, there is a request for a workpiece from a workstation. Since the MHS interface can accommodate only one AGV at its bay, it would appear prudent that the request for a workpiece be serviced by the first AGV, thereby displacing the unload request from the second AGV. The problem is further complicated when batch priorities are to be included in the analysis. In the simulation model, all requests for stacker functions were coordinated through a set of outstanding notices, which logged the workpiece involved, its source and destination, request timestamp, and the status of originating or target resources. This enabled some lookahead and planning of efficient stacker utilization.

Since the amount of buffer space could not be accurately estimated ahead of time, it was decided to model the racks with unbounded capacity. This allowed the tracking of maximum storage requirements more accurately, while simultaneously simplifying the location and retrieval problems. Information stored about the racks included current utilization, and location of specific workpieces in the rack.

It was determined that, in principle, two AGV's would suffice for the movement of material throughout the system. However, given the possibility of breakdown, and the need to provide reserve capacity, it was decided that up to four AGV's would be employed. As with the stacker, a set of outstanding notices was also maintained for AGV's. However, while each stacker had its own list, a common list was maintained for all AGV's. The information tracked about AGV's included their current status, location, destination (if applicable), and a code to indicate if its progress was impeded for any reason.

Four workstations were employed in this system—two horizontal spindle machines, one vertical spindle machine, and a measuring machine. Each workstation used a rotary shuttle to transfer workpieces from the AGV to the machine. The shuttle was capable of handling two workpieces, thereby enabling the next workpiece to be available when an operation was completed. This strategy was employed in an effort to maximize workstation utilization. One shuttle position was always kept open to prevent station blockage. Decisions as to which workpiece would be loaded onto a workstation form part of the control problem that is being investigated, and are discussed in the next segment.

The parameters used during the simulation are based on the actual data from the factory setting. All units are in min (or min^2). The arrival of batches was modeled as expo(700). The type of workpiece was specified based on a histogram specifying the relative probabilities of different workpieces. The number of workpieces was drawn from $U(5, 15)$. The lead time for a batch was computed from an expected time per workpiece multiplied by the batch size. Workpiece operation times were drawn from $N(750, 40\,000)$. This lead time, coupled with the arrival time, was used to specify the due date. A workpiece could follow any of three possible routings, based on prior probabilities of occurrence. Each routing specified the operations to be performed for producing the final product. Within a routing, several variations were possible, reflecting a spectrum of operation times. As with the case of routings, the variations were selected based on an input histogram reflecting prior probabilities. The number of active batches in the system was restricted to 2. As soon as a batch was completed, another was selected from the backlog of orders for completion. The selection of a specific order from this set formed the other aspect of the control problem that was investigated as part of this exercise.

2. The Scheduling Task: The model affords significant variability in terms of physical design characteristics, operations managed, and decision making strategies adopted. Physical design considerations include size and scope of the FMS, as in working at the level of a facility, shop, or cell, with a limited number of workstations, robots, material handling system, and automated guided vehicles. They also cover the

FMS organization including hierarchical or cellular design. Operations of interest in FMS scheduling and control include the selection of parts from an order stream for inclusion in the job stream, assignment of parts to cells for production, routing within the cell, loading and tooling decisions, and maintenance of resources in the system. Decision strategies will vary according to the operation at hand—e.g., part loading decisions could follow heuristics as First-Come-First-Serve, Shortest Processing Time, Earliest Due-Date, or other appropriate strategy. Clearly, the number of alternative scenarios for complete coverage of the domain represents an exponentially complex term.

The task involved the selection and assignment of parts to workstations in a cellular structure, with a view to achieving specific management goals. The first decision addressed in the scheduling process was the selection of a batch from the order stream for processing. The decision was influenced by such factors as the need to maximize workstation utilization, minimize work in process, and adhere to the due dates. Decision strategies employed included the *First-Come-First-Serve*, *Shortest Processing Time*, *Earliest Due Date* and *Minimum number of Jobs Remaining* heuristic scheduling rules. The second decision of interest addressed the assignment of workpieces to workstations. Factors of interest here included adherence to due dates, and utilization of cell resources, including workstation, AGV, and stacker cranes. The same four heuristic decision rules were employed at this stage too. The choice of heuristic scheduling rules was to some extent influenced by its relevance to the selected objectives, and extent of usage in practical settings. Consequently, scheduling rules that introduced significant complexity were omitted from the analysis. Additionally, this kept the problem tractable.

B. The Classification Task

The application of the C^3 algorithm differs from the traditional in some respects. Normal classification applications involve the use of explicit specification of all input and output dimension values for each observation. There is no room for inconsistent observations. Moreover, output dimension values are treated as given. These may be based on incontrovertible facts as in whether a mushroom is poisonous or not, or may be specified through expert opinion, as in when to accept an application for credit. In the case of FMS scheduling, the lack of provably superior decision rules and the paucity of human experts render this strategy for determining output dimension values inadmissible.

Yet another strategy calls for the application of all decision rules in several simulation runs, and then selecting the "best" for that scenario. The "best" decision rule can then be employed for classification tasks. This approach, though appealing in some respects, suffers from some obvious flaws. First, the notion of "best" is not easily defined. Several criteria may be employed to evaluate FMS scheduling performance, and the decision rule that is selected will vary with the criteria. This entails the creation of separate classification tasks for each criterion, which aside from being potentially expensive, provides little guidance to the decision maker when multiple scheduling criteria are to be used. An alternative approach is to employ some combining function, as in a utility or other multi-criteria function, to simultaneously consider all desired scheduling criteria. The multiple criteria decision making literature provides several approaches to accomplish this effectively, though it imposes a caveat that any such combining function is likely to be specific to the individual decision maker, and hence of limited generalizability. Second, and perhaps more important, the knowledge gleaned from such an approach is likely to be applicable to the chosen FMS model only, thereby limiting its usefulness in other scheduling applications.

Given these restrictions, it was decided that classification would be performed using the raw values of the output dimensions. This does pose the risk that there may be individual observations where the input dimensions are identical, but the output dimensions differ. This situation may be considered to be an anomaly. Anomalies of this sort can be resolved in a variety of ways. The anomaly can be treated as a conflict, in which case the conflict is to be resolved prior to classification. This can be achieved by discarding either or both observations. In situations where no information is available, both need to be discarded. If expert assessment is possible, then only one needs to be discarded. Since scheduling experts were not easily available, this approach was ruled out.

The alternative strategy is to treat an anomaly as given, recognizing that both observations are genuinely possible. In such cases, the classification algorithm will terminate at a point with non-zero entropy, indicating the inability to further classify. A simple probabilistic strategy may be applied at this stage, based on the relative size of the subsets involved. Since data was generated through the simulation model, it is possible to ensure that no anomalies are present.

C. Implementation of C^3

The C^3 algorithm used in this research was implemented in C++ and runs on an IBM PS2/70. Given the overall simplicity of the algorithm, the choice of language was dictated more by convenience rather than other factors as efficiency, data structures supported, recursive ability, and the like.

One problem faced during the execution of the program was the propensity to encounter situations where the conditional entropies for several candidate dimensions were identical at a classification stage. This situation was termed a tie. In general, there are several mechanisms for breaking ties. A simple strategy is to select the first dimension as the preferred dimension at all stages. The initial ordering of dimensions will dictate the resolution of ties under this strategy. Another strategy that can be employed involves the random selection from the set of tied dimensions. While this is relatively easy to implement, it may lead to an unbalanced tree. Yet another strategy involves lookahead to determine which is the preferred dimension for selection at this stage, based on the information content of the next level of the tree. However, this approach violates the "greedy" structure of the classification algorithms, and its logical extension would comprise a branch-and-bound solution.

The above procedures are devoid of any semantic content, using some information theoretic measures at best. It may sometimes be desirable to employ domain knowledge to break ties. This may be done at a global level, assuming transitivity of importance of individual dimensions, and the overall importance can be then specified by an ordered list of input dimensions. If transitivity of preferences cannot be guaranteed, then a matrix depicting pairwise preferences can be used. However, it may be the case that the preference for utilizing a dimension at any stage over other tied dimensions may indeed be context sensitive. In this case, for every branch of the decision tree, a list or matrix indicating relative preference among dimensions is needed.

The overall objectives of the classification exercise will dictate the strategy for resolving ties. If the objective is merely to create a decision tree that can be later employed at will for classification purposes, then the random or first-come-first-select approach may be useful. If efficiency of classification is a criterion of significant importance, the lookahead strategy is preferred, in that it engenders parsimony of the resulting tree. If, on the other hand, the classification exercise is intended to shed light on the domain, such that this knowledge can be employed at a later point, the semantic-based approach is preferred. This will facilitate the creation of systems that employ alternative knowledge structures, for example, production rules, frames, and the like.

In this particular application of conditional classification, the overall objective was to generate an effective decision tree, i.e., one which performed well on subsequent classification tasks. Parsimony of the tree, though desirable, was not the overriding objective. Since the knowledge encapsulated in the tree will not be converted to other forms, semantic information for breaking ties is not warranted. Consequently, the random strategy was employed to break ties.

D. Classification Exercises

The essential aim of the research was to assess the viability of concept-based conditional classification for manufacturing scheduling decisions. Several mechanisms are available for evaluating the outcome of the classification process. One strategy is to examine the outcome of the classification exercise itself. This can be done from an information theoretic perspective (e.g., parsimony of the decision tree), or from a domain semantic perspective (where the tree is examined for soundness of the classification rules). A more effective strategy involves the predictive validity of the resulting decision tree by applying it to a set of new observations. This study elected to use a set of training observations to construct the decision trees, and then evaluate its ability to effectively classify a set of holdout data.

A set of 64 observations was employed for classification and evaluation. The input dimensions for classification included batch data and cell status information, and are summarized in Table I. Though there were several variables or dimensions that could be manipulated in the simulation runs, it was decided to restrict the number to a value that could be handled by a human decision maker. Further, only decision variables that of greatest concern to the decision maker were employed. The 64 observations represented all possible combinations of the input dimensions.

TABLE I
INPUT DIMENSION FOR CLASSIFICATION

X_i	x_{ip}
Batch characteristics	
Number_of_part_types	2, 3
Cell Status characteristics	
Palette_capacity	5, 6
Number_of_AGVs	2, 4
AGV_speed	Low, High (100, 125)
WS_availability	3, 4
Buffer_capacity	2, 3

TABLE II
OUTPUT DIMENSION FOR CLASSIFICATION

C_j	c_{jq}
Production rate	Met, Not_met
Utilization	Met, Not_met

The output classification dimensions were workstation utilization and production rates with possible states (values) depicted in Table II. The criteria for determining whether the goals were met or not was based on a boolean expression involving the production rates for individual product lines, and the utilization of individual workstations.

The observations were generated through a set of simulation runs. The input parameters were systematically manipulated so as to produce all possible combinations of the data values. The set of 64 observations was randomly split into equal sets of 32 reflecting training and holdout sets. A quick inspection of the data indicated that all possible states for each dimension were covered in both sets.

V. Results and Discussion

Classification was performed on the training set, yielding the two decision trees in Fig. 7. Note that while the simple classification yields disjoint sets at the leaf level (entropy = 0), the conditional classification tree will generally produce a fully decomposable matrix relating the values of the variables in question at the leaf level.

The holdout set was then verified against the decision trees generated. At the first stage, i.e., using production rate as the output dimension, the accuracy of classification was 100%. At the second stage, classification accuracy reached 90.63% for predicting achievement of utilization goals given the attainment of production rate goals.

The accuracy of classification can be misleading. Firstly, the sample sizes were relatively small, and so there was less room for errors during classification. Additionally, the data set employed produced some ties during the classification process. A random strategy for breaking ties fortuitously resolved them in a manner that permitted greater accuracy. For

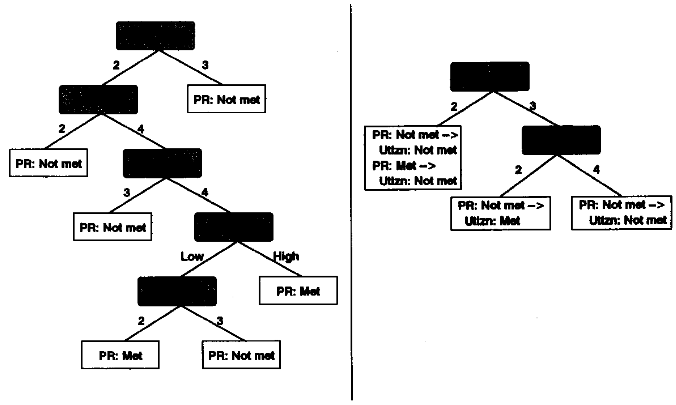


Fig. 7. Decision trees (simple & conditional).

example, in the first stage, after the first split, the dimension *Palette_capacity* could have been chosen instead of *Number_of_AGV's*. Had this been the case, the classification accuracy would have deteriorated to 90.63%. Likewise, during the conditional classification stage, other dimensions than those selected were also candidates for inclusion.

The results are very much data dependent. An examination of the decision trees indicated that some of the rules were counter-intuitive. This prompted an examination of the accuracy of the classification process, which proved to have worked correctly. This in turn led to an investigation of the data sets themselves and the simulation runs used to generate the data. This yielded some interesting insights as to the nature of the data itself. It was found that in some cases the system attained near complete utilization, and thus the outcomes were really not affected by the decisions taken. However, this does not detract from the efficacy of the conditional classification process.

Further, the categorizing functions employed also play an important role in the effectiveness of this approach. Thus for example, if the criteria for determining if the production rate and utilization goals had been met were altered, the results most likely would have been different. Attempting to generalize from these results is clearly foolhardy.

Two specific issues direct the need for caution in interpreting these results. First, the relationships between input and output dimensions are tenuous at best. As with any data driven strategy, a relationship can always be determined, though the semantic or causative nature of the relationship is difficult to establish. The inherent limitations of any classification process are clearly applicable in this case. The knowledge gained is relatively superficial, and may not apply in other cases. Any form of deep understanding of the domain is lacking. The second concern is that the use of repetitive and dynamic decisions tends to make any interpretation extremely context sensitive.

However, the results are encouraging, and a case can clearly be made for further examining the effectiveness of conditional classification. Additional investigation is necessary, first to establish the viability of conditional classification as a meaningful learning strategy, and second to assess its utility vis-a-vis other concept-based learning strategies. To achieve the former, the algorithm must be applied to other decisions, possibly more complex ones, using larger data sets, both in terms of input dimension cardinality and number of observations. The use of noisy data sets, coupled with anomalies is also suggested to investigate the robustness of the algorithm. To this end, the system is now being ported to ProKappa on an HP 9000 series 700 platform, thereby providing greater flexibility for a suite of experiments. On the second front, a series of experiments comparing the effectiveness of conditional classification versus simple and other variants of entropy-based classification needs to be performed. This paper represents a starting point for research in multidimensional classification.

VI. Conclusion

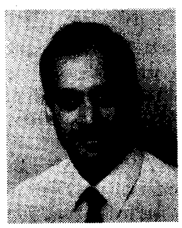
This paper examines the problem of learning from observations when "rules" are to be inferred on separate but related aspects of a problem, using identical or overlapping data sets. This form of learning is termed multidimensional classification. A general framework describing the various types of multidimensional classification is provided. The paper specifically concentrates on conditional classification, wherein the classification is initially performed on a dimension specified by the user, and the resulting knowledge then used to classify on other dimensions of interest. Drawing from concept learning and information theory, algorithms are presented for acquiring tree-structured knowledge from available data. An application to FMS scheduling is presented. Results indicate that conditional classification may provide some ability to better interpret related decisions in automated manufacturing contexts. Further work is necessary to ascertain if the approach is robust, particularly on more complex decisions, larger data sets, and noisy data. The paper lays the groundwork for sustained research in multidimensional classification, with some pointers for further research outlined.

ACKNOWLEDGMENT

The authors are grateful to the anonymous referees for their comments and suggestions on an earlier version of the paper.

REFERENCES

[1] J. Aczél and Z. Daróczy, *On Measures of Information and Their Characterizations*. New York: Academic Press, 1975.
[2] J. H. Boose, "A survey of knowledge acquisition techniques and tools," *Knowledge Acquisition*, vol. 1, no. 1, pp. 3–37, Mar. 1989.
[3] W. Buntine and T. Niblett, "A further comparison of splitting rules for decision-tree induction," *Mach. Learning*, vol. 8, no. 2, pp 75–85, 1992.
[4] L. Brieman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees*. Belmont, CA: Wadsworth, 1984.
[5] C. Carter and J. Catlett, "Assessing credit card applications using machine learning," *IEEE Expert*, vol. 2, pp. 71–79, Fall 1987.
[6] A. R. Chaturvedi, G. K. Hutchinson, and D. L. Nazareth, "A synergistic approach to manufacturing systems control using machine learning and simulation," *J. Intell. Manufact.*, vol. 3, pp. 43–57, 1992.
[7] A. R. Chaturvedi, G. K. Hutchinson, and D. L. Nazareth, "Supporting complex real-time decision making through machine learning," to appear in *Decision Support Syst.*, 1993.
[8] R. G. Gallager, *Information Theory and Reliable Communication*. New York: Wiley, 1968.
[9] A. Hart, "Experience in the use of an inductive system in knowledge engineering," in *Research and Developments in Expert Systems*, M. Bramer, ed. Cambridge, MA: Cambridge Univ. Press, 1984.
[10] E. B. Hunt, *Concept Learning: An Information Processing Problem*. New York: Wiley, 1962.
[11] E. B. Hunt, J. Marin, and P. T. Stone, *Experiments in Induction*. New York: Academic Press, 1966.
[12] R. H. Jackson and A. W. Jones, "An architecture for decision making in the factory of the future," *Interfaces*, vol. 17, no. 6, pp. 15–28, Nov.–Dec. 1987.
[13] R. S. Michalski and R. Chilauski, "Knowledge acquisition by encoding expert rules versus computer induction from examples: A case study using soybean pathology," *Int. J. Man-Mach. Stud.*, vol. 12, no. 1, pp. 63–87, 1980.
[14] J. Mingers, "Inducing rules for expert systems—Statistical aspects," *The Professional Statist.*, vol. 5, pp. 19–24, 1986.
[15] J. Mingers, "Expert systems—Experiments with rule induction," *J. Oper. Res. Soc.*, vol. 37, no. 11, pp. 1031–1037, 1986.
[16] J. Mingers, "Expert systems—Rule induction with statistical data," *J. Oper. Res. Soc.*, vol. 38, no. 1, pp. 39–47, 1987.
[17] J. Mingers, "An empirical comparison of selection measures for decision-tree induction," *Mach. Learning*, vol. 3, no. 4, pp. 319–342, Mar. 1989.
[18] D. L. Nazareth and A. R. Chaturvedi, "Multidimensional classification of discrete data," working paper, School of Business Admin., Univ. of Wisconsin-Milwaukee, Mar. 1992.
[19] J. R. Quinlan, "Semi-autonomous acquisition of pattern-based knowledge," in *Introductory Readings in Expert Systems*, D. Michie, ed. New York: Gordon and Breach Science, 1982, pp. 192–207,.
[20] J. R. Quinlan, "Induction of decision trees," *Mach. Learning*, vol. 1, no. 1, pp. 81–106, 1986.
[21] J. R. Quinlan, "The effect of noise on concept learning," in *Machine Learning II: An Artificial Intelligence Approach*, R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, eds. Los Altos, CA: Morgan Kaufmann, 1986, pp. 149–166.
[22] J. R. Quinlan, "Probabilistic decision trees," in *Machine Learning III: An Artificial Intelligence Approach*, Y. Kodratoff and R. S. Michalski, eds. Los Altos, CA: Morgan Kaufmann, 1990, pp. 140–152.
[23] J. C. Schlimmer, "Concept acquisition through representational adjustment," Ph.D. dissertation, Dept. of Info. and Comp. Sci., Univ. of California, Irvine, 1987.
[24] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423, 623–656, 1948.
[25] J. Wirth and J. Catlett, "Experiments on the costs and benefits of windowing in ID3," in *Proc. Fifth Int. Conf. Mach. Learning*, 1988, pp. 87–99.



Alok R. Chaturvedi received the Ph. D. in management information systems from the University of Wisconsin-Milwaukee, in 1989.

He is an Assistant Professor of Management at the Krannert Graduate School of Management, Purdue University. His main research interests include machine learning, distributed systems design, and intelligent manufacturing.

Dr. Chaturvedi is a member of the ACM, the American Association for Artificial Intelligence, TIMS, and ORSA.

Derek L. Nazareth received the Ph. D. degree in management from Case Western Reserve University.

He is an Associate Professor of Management Information Systems at the University of Wisconsin-Milwaukee. His research interests include knowledge-based system verification, machine learning, and distributed object-oriented systems.

Dr. Nazareth is a member of the American Association for Artificial Intelligence, the ACM, IEEE Computer Society, and TIMS.