How Can AI Procedures Become More Effective for Manufacturing?

Boris Kovalerchuk[1], Evgenii Vityaev[2] and Evangelos Triantaphyllou[1]

[1]Department of Industrial and Manufacturing Systems Engineering
Louisiana State University, Baton Rouge, LA 70803, USA
borisk@unix1.sncc.lsu.edu

[2]Institute of Mathematics, Russian Academy of Science,
Novosibirsk, 630090, Russia
vityaev@math.nsk.su

Abstract

The problems of AI effectiveness in manufacturing for Design, Scheduling, Control and Process Diagnosis are considered. We have developed an effective dialog procedure for a designer. This procedure helps him to identify the needed parameters of a designed product, i.e., to distinct acceptable parameters, non acceptable parameters and parameters that require additional design study. In scheduling we developed a new intelligent procedure to formulate and find an effective schedule. Often such approach can avoid complicated time-consuming computations. In Control we developed simple and robust control procedures, which join the advantages of pure/conventional interpolation and fuzzy control methods for design of quick and cost-effective controllers. In Process Diagnosis we overcome some difficulties of such known methods as neural networks, linear discriminant analysis and the method of nearest neighbors. The main difficulties which we overcome are related to the speed of a dynamic learning process and reliability of diagnosis. We also make the extracted diagnostic regularities easily understandable by a manufacturing expert. This approach was successfully used for several tasks related to engineering and medical problems.

Keywords: design, diagnosis, scheduling, fuzzy control, knowledge discovery.

1. Introduction

We study a way how AI procedures can become more effective in manufacturing. A developed interactive procedure allows designers to identify the needed parameters of a designed product. We are discovering properties of job sequences which allow us to formulate effective criteria of optimal scheduling. In control tasks the main AI tools are fuzzy control methods. These methods have shown the effectiveness, but still have lots of problems. A design of quick and cost-effective controllers especially for mobile real-time systems is one of them. In process diagnosis there are many difficulties of such known methods as Neural Networks, linear discriminant analysis and the method of nearest neighbors. The speed of a dynamic learning process, reliability of diagnosis and understanding of procedures by the user--manufacturing expert are among main difficulties of these methods. We present an overview of our developments.

2. Design Problems

In Design our interest isfocused on procedures, to assist formulation of design criteria as formal requirements for a product. Usually it is a very complicated multilevel and multi-attribite task. This is a new important problem. The solution of this problem can significantly speed up design.

We have developed an original effective dialog procedure (Kovalerchuk, Triantaphyllou, Despande & Vityaev 1996a), which allows designers to restore lower and upper borders in a hierarchical multi-attribute space. These borders help a designer to identify the needed parameters of a designed product. They help to distinct parameters which must be reached from parameters which must not be reached, and parameters which require additional design study.

An illustrative example below shows the main steps of the approach. Suppose that one wishes to design a mid-sized car under $25,000 better than the average level on the market. Table 1 presents features of the cars on the market, taken from Consumer Reports, 1994 (p.160). We use the following notation in this table: x_1 for the overall score, x_2 for predicted reliability, x_3 for overall mpg, x_4 for driver's air bag, x_5 for passenger's air bag, x_6 for antilock brakes, x_7 for auto-transmission and x_8 for air-conditioning. Also "P" denotes "optional", "1"and "0" denote available and not available, respectively.

Table 1. Rating 1993 mid-sized cars under $25.000

car	x_2	x_2	x_3	x_4	x_5	x_6	x_7	x_8
Camry 6	4	5	21	1	0	P	1	P
Camry 4	4	5	24	1	0	P	P	P
Ford Taurus	4	3	20	1	P	P	1	P
Mercury	4	3	20	1	1	P	1	1
Maxima	4	5	21	1	0	P	P	1
Chrysler NY	3	3	21	1	0	P	1	1
Buick Regal	2	3	20	0	0	P	1	1
Chevrolet Lumina	1	2	22	0	0	P	1	P

Next. the designer needs to identify **a combination of car features**, which should be **design requirements** (see table 1). It requires to analyze a huge amount of possible feature combinations with different overall scores (x_1 in table1). There are 5 values for predicted reliability, 5 values for mpg (20,21,22,23,24), 2 for driver's air bag, 2 for passenger's air bag, 3 for each of the next three features (brakes, transmission, air-conditioning). The values for these three features are "yes", "no" and "optional". There are 2x5x5x2x2x3x3x3=5,400 of the combinations. A designer should choose some of them. This number demonstrates the size of problem. Table 1 brings information about only 5 cases from these 5,400. The problem is how to **assist a designer** to analyze this 5,400 cases and discriminate acceptable cases from not acceptable.

Some cases can be excluded from further considerations relatively easy. For example, if all existing cars have antilock brakes, auto-transmission and air-conditioning options, probably it will not be wise to design a car without these options. The same is true for the driver's air bag. If all cars with 4 overall scores have air bags then a new car should also have it. We can also restrict an acceptable range of reliability with values 3, 4 and 5. All cars with overall scores 4 have this reliability. By combining all these restrictions we obtain a new number of design cases 3x5x2x2x2x2x2=480. We need to classify them into two classes: (1) acceptable for design and (2) not acceptable for design. This design problem has the **monotonicity** property. If some car (a combination of values of features) is acceptable for design then a better car (a combination with stronger feature values) should also be acceptable. This property together with the hierarchy of design features allows to optimize the sequence of the analysis of cases, using the powerful mathematical theory of monotone Boolean functions and software.

An interactive system presents to the designer a case (values of feature combinations), obtains his answer (acceptable/not acceptable) and presents the next case depending on all his previous answers. This optimal procedure allows to reduce the number of analyzed cases up to 50-100 times, as we have shown for another applications (Kovalerchuk et, Triantaphyllou, Despande & Vityaev 1996a). In our experiment we decreased the number of requests for analysis from 4,000 cases to 40 cases.

3. Scheduling

In Scheduling we are interested in criteria formalization for an optimal schedule and intelligent procedures to find such optimal schedule. Often the right formulation of criteria allows to avoid complicated time-consuming computations.

This is also a new important problem, where AI research methods are very promising. Currently, this problem is out of formal analysis. Non formalized expert knowledge is a base of mostly intuitive solutions. There are some interesting attempts based on fuzzy logic made in Boeing Co. (Kipersztok & Patterson, 1995) for monitoring network load of jobs submitted to clusters of workstations.

To clarify the situation, let us also consider a task from (Kipersztok & Patterson 1995) in more detail. For these tasks most existing systems employ the first-in, first-out rule (FIFO) to assign programs to a cluster of machines independently of the network load and communication requirements. Advanced scheduling should address which of the incoming jobs should be sent to the network cluster first, and which of the currently running jobs should be suspended. It requires a process of **prioritization** of the jobs and **rules** that match the resource specification requirements of the jobs to the current state of the available resources in the cluster. Next the authors introduce a priority rank for each job. Importance of cluster features defines priority ranks. Priority ranks show how sensitive an incoming job is to each parameter, which characterize the cluster. These parameters are introduced using fuzzy logic. Then they are combined with a weighted linear function in the **priority rank**. This is the weakest point of this kind of approaches. The priority rank is a heuristic construction. It can fit the task or not. The designer's experience is of critical importance for the construction of a priority rank. We develop an alternative approach related to direct extraction of **rules that match the resource of the jobs to the current state of the available resources** using data of the previous performance of the system. The main difficulties, which restrict existing methods of machine learning to extract rules from data are that for scheduling the data are not usual points in a feature space, but they are much more complicated. A technique based on first-order logic (Vityaev & Moskvitin 1993) can overcome these difficulties. Let us illustrate this idea in the previous example. For each time we may have records how effectively the system uses resources. We compute percentage of the used processors' time with the length of this interval for all jobs and for each job separately for some time interval. Next we can collect these data for sequential time intervals. Then we use methods of data mining and knowledge discovery to discover relations between sequential data. We present data for two pairs of time intervals (t,t+1) and (k,k+1) in figures 1 and 2. Jobs are presented beginning with those that required more time. We also measure the total effectiveness of a resource usage for all jobs. Let it be 90% for figure 1(b) and 40% for figure 2(b). It allows us to extract the following rule. If jobs require resources as in figure 1(a) or 2(a) at the moment *i* THEN for the next moment *i*+1 the set of jobs should be as in Figure 1(b). It means that these jobs should require the same or close share of resources as in figure 1(b). It will allow to have total effectiveness up to 90%. This job sequence is much more complicated than FIFO and can be significantly different from that used in heuristic priority formulas. To be reliable this rule should be discovered on a bigger data set. This method allows us to discover many rules without a human expert. Then we develop a procedure to match rule premises with current system states. Next we describe how we discover rules using figures 1 and 2 and the first order logic.

Let a1, a2, b1 and b2 be job sets from figure 1(a),(b) and figure 2(a), (b) respectively. Next, the predicate V(a1,a2) is true if and only if the plot in figure 1(a) is above the plot in figure 2 (a). Similarly predicate P(a1,a2) is true if and only if E(b1)>E(b2), where E(b1), E(b2) are effectiveness' of resource's usage by all jobs in figure 1 (b) and figure 2 (b), respectively. Then we can describe the rule:

IF V(a1,a2) THEN P(a1,a2),

which should be tested against a data set of such figures. If this rule is confirmed on the data set, we can use it for scheduling as we described above. In the same manner we automatically extract similar rules using larger sets of predicates not only V and P.

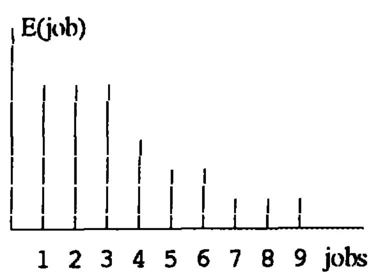

Fig. 1 (a) Job effectiveness E at time t

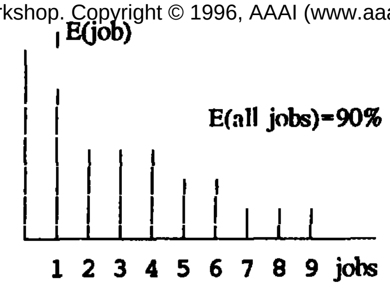

Fig. 1 (b) Job effectiveness at time t+1

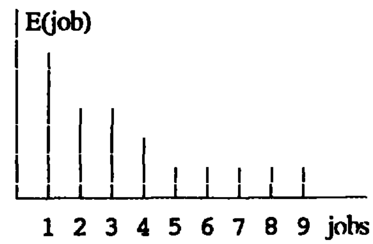

Fig.2 (a) Job effectiveness at time k

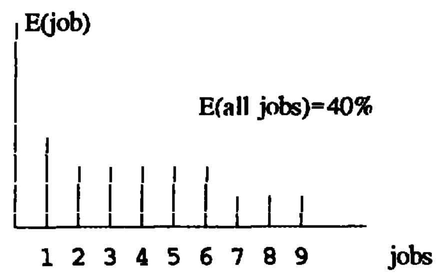

Fig. 2 (b) Job effectiveness at time k+1

4. Control Problems

In **Control** we are interested in the development of simple and robust control procedures. The main AI tools here are fuzzy control methods, which have shown the effectiveness, but still have lots of problems. One of these problems is a **design of quick and cost-effective**

controllers especially for mobile real-time systems.

There are three areas of control methods:

(1) the area of conventional control methods;
(2) the area of pure/conventional interpolation methods;
(3) the area of fuzzy control methods.

Conventional control methods cover control tasks with a known model of the controlled process. This model is usually presented with differential equations.

Pure/conventional interpolation methods cover control tasks without a model but with sufficient training data to interpolate a control function. Many interpolation methods are used here: polynomial interpolation, spline functions, piecewise linear interpolation etc.

Fuzzy control methods cover control tasks without a model and without sufficient training data, but with expert linguistic control rules.

Furthermore, there are many intermediate cases between these three cases. Intermediate cases require mixed approaches, which are now under development. The problem is that there is no way to know in advance **if the training data are sufficient or not** to choose an appropriate method. Usually, it becomes clear only after testing with large independent test data or several trials of use the system. In (Mouzouris & Mendel 1996) it was shown that with extension of a training set linguistic information becomes less important.

We offer an approach which helps to choose an appropriate method when we do not have a model of the process. This approach is related to the perspective direction -- the development of AI tools for an **adequate fuzzy control task formulation.** Adequacy means that we need to avoid oversimplifications as well as too complicated methods, balancing between these two extremes.

At first we have found conditions where the fuzzy control is practically equivalent to the conventional interpolation (Kovalerchuk 1994a, 1996a). If these conditions for input and output membership functions are fulfilled we can use conventional interpolation methods and obtain a very simple piecewise linear interpolation of the control function. As a result we may have a simple and cost-effective controller. The output of this interpolation differs from Mamdani controller output no more than 5.05% of the length of the support of the used fuzzy sets (Kovalerchuk, 1994a, 1996a). The length of the support of a fuzzy set is the length of the interval where its membership function $\mu(x)$ is above zero. For triangular membership functions it is the length of the interval between slopes, where $\mu(x)>0$.

If our conditions for membership function shapes and overlapping are not fulfilled, we organize a **guided extension of training data interactively with a designer** to fulfill these conditions. The guided extension of training data requires less data to tune control function than a usual random extension. The relations between input and output fuzzy sets of control function give us guidance how to extend the training data. The main idea is to adjust the fuzzy sets to form the so called exact complete context spaces (Kovalerchuk, 1995, 1996b). Next we select peaks of membership functions (MFs) of these adjusted fuzzy sets. These peaks form a new training data set. This procedure allows us to immediately find a piecewise linear interpolation of the control function. This interpolation has the same deviation from the Mamdani controller as above mentioned 5.05% of the fuzzy set support.

We join advantages of fuzzy control and pure interpolation in this method, realizing a mixed approach. We call our version of interpolation the second interpolation. This interpolation was inferred from Mamdani fuzzy controller. Mamdani controller itself also represents an interpolation of a real control function, but it is not a pure interpolation based on training data only

Figures 3, 4 and 5 illustrate conditions when our interpolation can simplify a fuzzy controller, thus substituting the fuzzy controller. Our conditions are represented in figures 4 and 5. These conditions are very common in fuzzy control applications. For example, a classical control problem of balancing an inverted pendulum was successfully solved using membership functions presented in figure 5 (see, for example, (Beale & Demuth 1994)). These triangular membership functions were used for position of the cart, velocity in meters/second and radians/second, angles and force.

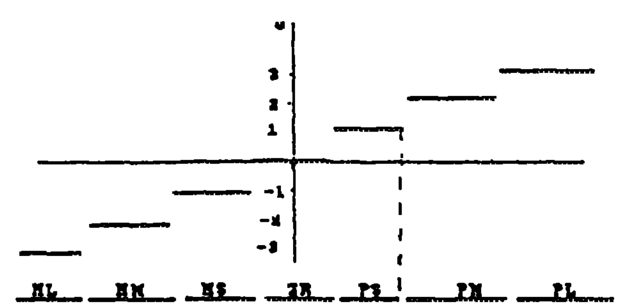

Fig. 3. Interpolation for non overlapping intervals

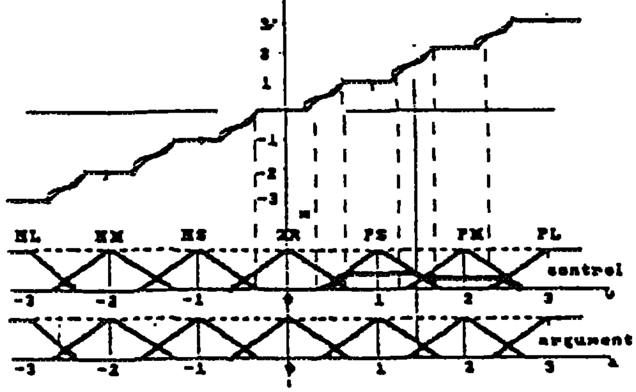

Fig. 4. Interpolation for partly-overlapping fuzzy sets

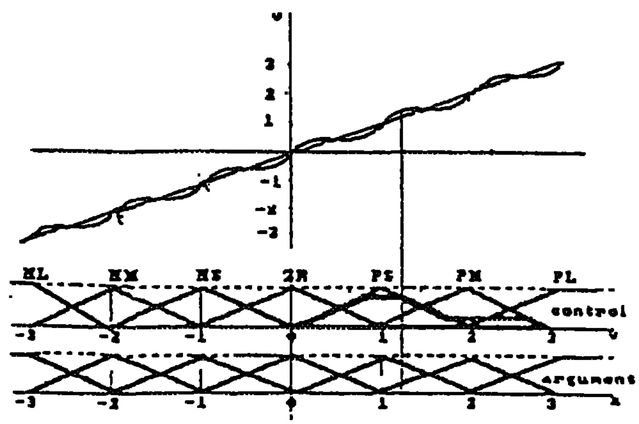

Fig 5. Interpolation for fuzzy sets with an "overlap of one"

Figure 3 shows a Mamdani fuzzy control function for non-overlapping intervals. It is a simple step function. In this example each input is presented as an interval and each output as an exact number. Also, control rules are: "If argument a is about zero (i.e., from the interval [-0.5,+0.5)), then control u should be 0"; "If argument a is positive small (i.e., from the interval [+0.5,+1.0)), then control u should be 1". The other rules are formulated similarly. In this simple case the fuzzy control method and the pure interpolation method give the same control function. Then we do not have a problem choosing one of them.

A fuzzy control function is presented for partly-overlapping fuzzy sets in figure 4. Here control rules are slightly different. Let us describe one of them: "If argument a is positive small then control u should be also positive small". The term "positive small" (PS) for argument a is formalized with a fuzzy set in the bottom of figure 4. The term "positive small" for control u is also formalized with a fuzzy set. This fuzzy set is presented above PS fuzzy set for argument a in figure 4. The next control rule is "If argument a is positive medium then control u should also be positive medium". The term "positive medium" (PM) for argument a is formalized with a fuzzy set in the bottom of figure 4 next to the fuzzy set for PS. The term "positive medium" (PM) for control u is also formalized with a fuzzy set. This fuzzy set is presented above the PM fuzzy set for argument a in figure 4. The other control rules are defined similarly. Figure 4 shows that the fuzzy sets "positive small" and "positive medium" are overlapping on the part of their supports. Control functions for the fuzzy control method and the pure interpolation are shown above the fuzzy sets. The control function for the fuzzy control method has a small wave in the area where PS and PM fuzzy sets are overlapped. The pure interpolation method gives a straight line in this area. Out of the overlapping areas the fuzzy control method and the pure interpolation method give the same linear pieces. The difference between these two interpolations in the overlapping area is described with the formula (Kovalerchuk 1996b, theorem 1):

$$u=[(1-(p+e)^2+p(1+e)[2+(1+e-p)]]/[(1-(p+e)^2+p(1+e)],$$

where e is the distance between the peak point of PS fuzzy set and the beginning of the PM fuzzy set and p is the distance from the beginning of the PM fuzzy set to the input point a, for which we compute a value u of the control function (see also figure 6). Computer simulation using the last formula allowed us to show that for this single input single output (SISO) case the difference between the fuzzy controller and the pure interpolation is no more than 2.07% of the length of the support of the used fuzzy sets. In figure 6 we show results of this simulation, when $w=1+e$.

A fuzzy control function is presented for optimally overlapped fuzzy sets in figure 5. Linguistic control rules are the same as for partly-overlapping fuzzy sets. The term "positive small" (PS) for argument a is formalized with a fuzzy set in the bottom of figure 5. The term "positive small" (PS) for control u is formalized similarly as a fuzzy set too. This fuzzy set is presented above PS fuzzy set for argument a in figure 5. The term "positive medium" (PM) for argument a is formalized with

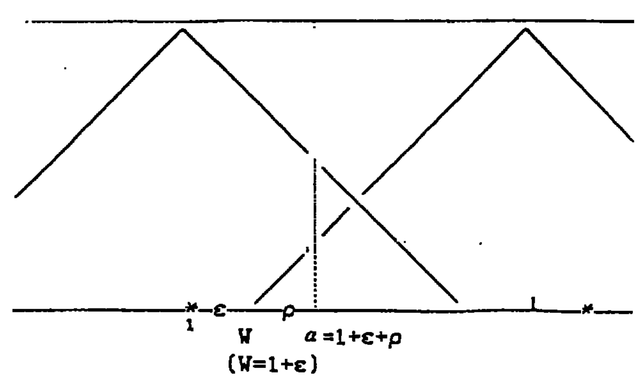

Fig. 6. Overlapping of triangular fuzzy sets

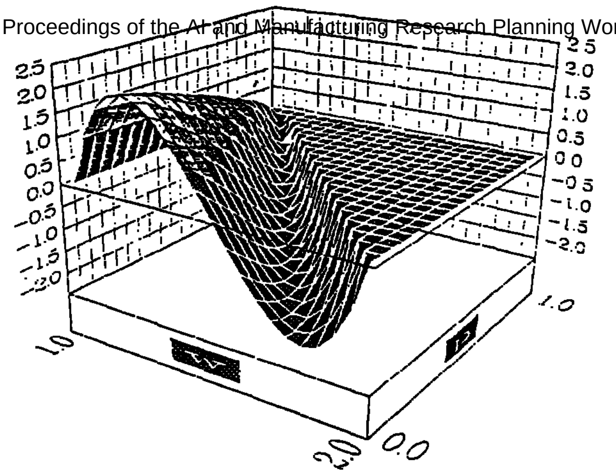

Fig. 7. Difference between fuzzy control function and the second interpolation

a fuzzy set in the bottom of figure 5 next to the fuzzy set for PS. The term "positive medium" for control u is formalized similarly as a fuzzy set. This fuzzy set is presented above PM fuzzy set for argument a in figure 5. Figure 5 shows that fuzzy sets "positive small" and "positive medium" are overlapping on the half of their supports. Control functions for fuzzy control method and pure interpolation are shown above fuzzy sets. The control function for the fuzzy control method has a slight wave shape for all domain. The pure interpolation method gives a single straight line. The difference between these two interpolations in the overlapping area is described with the formula (Kovalerchuk, 1996b; Corollary 1):

$$u=(2-3p^2+5p)/2(1-p^2+p),$$

where p is the distance from the beginning of the fuzzy set to the input point a, for which we compute a value u of the control function. Computer simulation using the last formula allowed us to show that for this single input single output (SISO) case the difference between fuzzy controller and pure interpolation reaches only 2.07% of the length of the support of used fuzzy sets. This maximum deviation 2.07% is reached for e=0 and two values of p: p_1=0.197200388 and p_2=0.8022799611. For two inputs and single outputs (TISO) case the difference between a fuzzy control function and a pure interpolation is no more than 5.05% of the support (Kovalerchuk, 1994, 1996a). Recall that the optimal fuzzy set overlapping shown in figure 5 is most common in fuzzy control. This study shows how to combine fuzzy control and pure interpolation to construct simple control functions. If fuzzy sets have a large support then difference between these two interpolations can be significant and the "wave" on figures 4 and 5 can be relatively large. In this case we need to decide which of the two control functions should be used. We argue that a piecewise linear interpolation between peak points of fuzzy sets has an important advantage. We consider a control task where all fuzzy sets are equal and symmetrical. How can they generate a wave? How to explain the wave in Mamdani control function in terms of a particular applied control task? There is no such explanation. The only explanation is out of context of the particular task. The source of the wave is Mamdani defuzzification procedure, i.e.. Center of Gravity (COG). Piecewise linear interpolation between peak points does not have this weakness.

Let us summarize our offer for control tasks without model and very restricted possibility to have a large training data set. We combine fuzzy control and pure interpolation methods. Fuzzy control methods are used to choose interpolation points and pure interpolation methods are used to interpolate between these points. With fuzzy control methods we extract linguistic rules, construct respective fuzzy sets preferably as in figure 5, identify their peaks. Then interpolation methods are used to identify a control function interpolating between these peak points. If the constructed fuzzy sets meet the above mentioned requirements, we do not need any more fuzzy control procedures to have practically the same control function as Mamdani control function (see also mentioned the above theorem 1 (Kovalerchuk, Triantaphyllou, Despande & Vityaev 1996a).

5. Process Diagnosis

In **Process Diagnosis** we are interested in using our original sophisticated diagnostic methods, which overcome some difficulties of known methods such as Neural Networks, linear discriminant analysis and the method of nearest neighbors. The main difficulties which we overcome are related to the speed of a dynamic learning process and reliability of diagnosis (Kovalerchuk, Triantaphyllou, Despande & Vityaev 1996a, Vityaev & Moskvitin, 1993). We also make the extracted diagnostic regularities easily understandable by a manufacturing expert.

Expert systems, linear discriminant analysis, neural networks, decision trees, and similar classification methods are the most known and effective tools for computer-aided diagnosis. They are used in many areas of diagnosis of machinery failures in the process of manufacturing and exploitation, military target recognition, detection of contamination of radioactive materials, non-destructive detection of damages in composite materials, drug design, medical diagnosis etc. Usually the accuracy of 90%, 95% or 99% is considered as an evidence of effective diagnosis. We study the following practical questions:

Do these numbers really evaluate the performance of a diagnostic system?

Should we buy and use a system with this accuracy for practical tasks?

We show that there is many misconceptions/ misunderstanding/illusions related to these critical for practitioners questions. Next we show how to distinct illusions of reliable diagnosis from a really reliable diagnosis and demonstrate how a reliable diagnosis can be accomplished.

Let us consider the **first illusion**, which we call **an illusion of a single index of accuracy**. Consider following real data: about 0.2% of 15,000 screened women have breast cancer (data from Woman's Hospital of Baton Rouge, 1995). If a system diagnoses all these women as not having cancer, then accuracy is 99.8% and only 0.2% of women have a wrong diagnosis according to this *"all right"* optimistic strategy. The general accuracy of this *"all right"* strategy looks like a dream 99.8%. Is it good? Of cause, the answer is negative. We must not only use accuracy itself, but examine *false-positive* and *false-negative mistakes* separately as is in real evaluation of cancer diagnostic methods. In these terms the system with *"all right"* strategy **makes no mistakes** in diagnosis of non cancer cases (100% of them are diagnosed accurately). But **all cancer** cases are diagnosed as non cancer cases (0% of accuracy). These two indexes allow us immediately to reject the use of such system. There are many other diagnostic fields where illusion of a single index of accuracy is not so obvious. They require a special analysis.

Next let us study how to evaluate the performance of a diagnostic system **in reality** having two indexes. If both of them are 99% for the entire population the total accuracy is also 99%. In this case usually the system performance can be evaluated positively. But how to evaluate the performance of a diagnostic system with 30% of false-positive and 80% of false-negative rates? In this case the total accuracy can be any number between 30% and 80%, including 50%. It means that a special study for a particular task will be needed to evaluate whether we should use or not this system for practical diagnosis.

We have shown above how critical is to find false-positive and false-negative error rates. It requires to solve a much more fundamental problem: identify a **real border** between diagnostic classes and compare it with a **formal border,** which have found by a diagnostic method, for example, by *the "all right"* strategy . Note, that the real border can be as narrow as a very wide area.

Let the border area consist of only 10% of all possible cases and a system improperly diagnoses all these border cases, i.e., 100% of mistakes on border cases from both diagnostic classes. Also let all other non border cases, i.e., 90% of cases be diagnosed accurately. This gives 90% of the total accuracy. Should such diagnostic system be used? It can be used only for relatively simple for diagnosis cases out of the border area. For all truly complicated cases the system gives 100% of mistakes. Therefore the presented 10% of mistakes is **the second illusion** of the diagnosis problem solution--**accurate diagnosis without complicated cases.** In reality this 10% means 100% mistakes for complicated cases, which really require sophisticated methods of diagnosis. Note, that often majority of simple cases can be diagnosed without any computer-aided diagnostic system. Therefore such a system with 90% accuracy can be useless.

The third illusion is related to a random choice of the test data. The standard approach to test diagnostic systems performance includes a test of a system on the randomly chosen set of cases. In the example above (with 10% of border cases and 90% out of this area) the random choice of cases will repeat this ratio. As a result, we will have the same 100% accuracy out of border area and 0% accuracy in the border area and the total accuracy of 90%. This means that we have the same second illusion of **accurate diagnosis without complicated cases,** but caused also by random chose of test cases. We call this illusion--the **random choice illusion.**

Next we discuss a critical for practical diagnosis question: *Why procedures of accurate evaluation of diagnostic methods have not been developed before?*

The reason is that diagnostic methods were developed and used in the frames of the **paradigm: the real border principally cannot be known.** We may know something about the real border only after **approximating it using** some method. Different methods give different approximations of the border using different **a priory assumptions** such as metrics of feature space, a type of distribution, a class of discriminant functions, a type of rules and etc. This **a priory assumptions approach** often leads to confusing diagnostic solutions and/or **illusions of accurate diagnosis**. Note, that often these assumptions are not formulated and should be discovered.

Next, what leads to **reality**, i.e., real solution of a diagnostic problem? It is **a method of finding real borders** between positive and negative examples and their diagnostic classes. **This problem has not been studied before,** because there was not known another way to find a real border except the construct of an approximation.

We develop another approach to restore the real border. This approach is based on the **Empirical Paradigm** and respective methods, which were developed in the former Soviet Union during the last 25 years (Zagoruiko 1981, Zagoruiko, Sviridenko & Samokhvalov 1978, Vityaev et al., 1976,1992,1993, Kovalerchuk et. al., 1975, 1996a). The core idea of this approach is in replacing a metric feature space by an **Empirical System** as this concept is defined in Measurement Theory (Krantz at al. 1971). Discriminant functions are replaced by hypotheses in the first-order logic, which express **empirically testable** properties of diagnostic classes of functions instead of choosing a particular function from this class (usually with interpolation technique).

We demonstrate our approach on the **empirically**

testable property of **monotonicity** and the class of all monotone functions. This is a rather general class of functions. But it is not necessarily for us to identify a particular monotone function to discriminate two patterns as it is, for example, in regression and discriminant analysis. We can directly use the empirically confirmed property of monotonicity for diagnosis. In this case all other non empirical a priory assumptions, needed for approximation methods, are eliminated. Also we show that the property of monotonicity can be **empirically discovered** with a high accuracy using a data set or interactively with an experienced expert. In the last interactive case we can discover empirically an exact monotone discriminant function too.

Let us illustrate the effectiveness of the property of monotonicity for diagnosis. We consider the task to discriminate two classes of design requirements: acceptable and non acceptable for a new car design using the example presented in table 1. For illustrative purposes we consider only two features from table 1: predicted reliability (x_2) and miles per gallon(x_3). Figure 8 shows these data. Numbers in the grid show the overall scores for cars from table 1. For example, $(x_2,x_3)=(2,22)$ corresponds to Chevrolet Lumina, which has grade 2 for forecasted reliability and 22 mpg; $(x_2,x_3)=(3,20)$ corresponds to three cars from table 1: Ford Taurus, Mercury Sable and Buick Regal. We use feature x_1 from table 1 as a criterion for discrimination of two classes: if $x_1 = 4$ design requirements are acceptable and if $x_1 < 4$ then they are not acceptable. If we consider 5 values for x_2 and 8 values for x_3 we will have 40 possible design cases. Recall that 8 cars from table 1 cover only 5 possible cases. We do not have sufficient statistics to construct a border between these two classes. Moreover, the classes are overlapping: two cars from the positive class and one car from the negative class correspond to the same point (3,20). There are dozens of possible discrimination lines. For example one of the simplest one is $x_2 \geq 4$ for the positive class and $x_2 < 4$ for the negative class. The other one if $x_2 \geq 3$ for the positive class and $x_2 < 3$ for the negative class. Both of these discrimination rules generalize these 5 points for all other points without argumentation. For example, both of them classify (5,19) as acceptable, but where are the arguments? Neural nets, linear discriminant functions and other methods usually do not control this situation for large data sets. For small data sets, as the one is in table 1, these methods cannot give any answer.

We resolve this problem exploring the semantics of the features. Semantics of features x_2 and x_3 give us the following monotonicity property. Let us consider two cars a and b with their properties:

(P1) forecasted reliability x_2 of car b is more than forecasted reliability x_2 of car a, i.e., $x_2(a)<x_2(b)$;

(P2) car a makes less miles per gallon than car b, i.e., $x_3(a)<x_3(b)$.

(P3) we are ready to design a car as car a, i.e., with

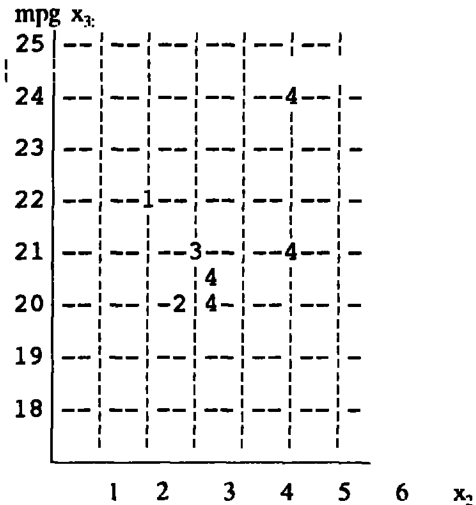

Fig. 8. Points from table 1.

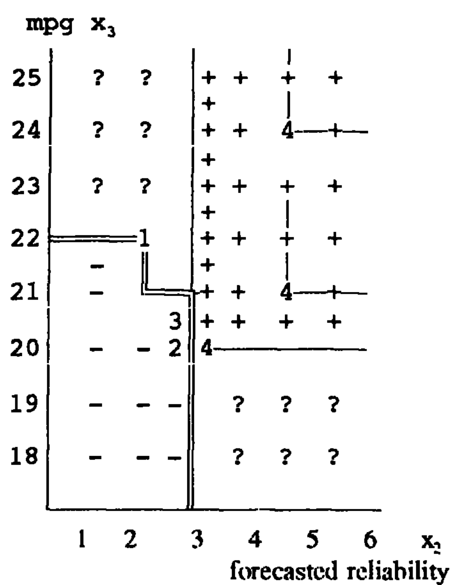

Fig 9. Monotone discrimination of positive and negative design classes

properties x_2 and x_3 of car a. We denote this as a property $D(a)$.

We see from (P1) and (P2) that car b is better than car a. To be logically consistent we should consider car b also as acceptable for design, i.e., to agree that $D(b)$ is true as well. Now we can write this property formally:

IF $x_2(a)<x_2(b)$ and $x_3(a)<x_3(b)$ and D(*a*) THEN D(*b*)
This is the **property of monotonicity**. For our features x_2 and x_3 and *D* the property of monotonicity is a result of their semantics. For other features monotonicity may be not so clear. In this case we need to discover this property empirically using data set and a method for discovery of regularities in the first order logic {Vityaev and Moskvitin, 1993].

The main advantage of the **semantical or empirical discovering monotonicity** that by this way we obtain **interpretable properties** of diagnosed classes, instead of interpolation of border using a priory assumptions. More often these assumptions are not known to the author of a method explicitly.

We exploited the discovered monotonicity for our illustrative task. Figure 9 shows the border between the positive and negative classes discovered using the semantical monotonicity. The points marked with "+" and "-" are from positive and negative classes, respectively. The points marked as "?" represent cases, for which our 8 training cases (cars from table 1) are not sufficient. A designer should analyze these cases with additional information. The other features from table 1 can be one of the sources of this information. This approach was successfully used for several tasks related to engineering design problems, signal recognition, breast cancer diagnosis, forecast of the surgery after effects, forming the secondary structure of proteins and so on.

6. Concluding Remarks

AI methods require deep understanding by each user of applicability of a method for his/her task. Ignoring analysis of applicability often leads to illusions of solutions, spending resources **without obtaining a reliable solution.** Currently many excellent user-friendly software systems allow to try any method easily without any analysis of applicability and reliability of solution. The problem of development of reliable AI methods is still open. In this paper we have shown how it can be accomplished for some diagnostic, design, control and scheduling tasks.

Acknowledgments

The authors gratefully acknowledge the support from the Office of Naval Research (ONR) Grant N0014-95-1-0639 and CAST-COBASE Grants from the US National Research Council.

References

1. Beale, M., and Demuth, H. 1994. *Fuzzy systems toolbox for use with MATLAB.* Boston, Mass.:PWS Publishing Co,.

2. *International journal of pattern recognition. and artificial intelligence*. 1989. 3(1).

3. Kipersztok, O., and Patterson, J. 1995. Monitoring network load for fuzzy scheduling of jobs submitted to clusters of workstations, In *ISUMA-NAFIPS'95*, 553-558. College Park, MD.

4. Kovalerchuk B., Triantaphyllou, E., Despande, A., and Vityaev, E. 1996a. Learning of monotone Boolean functions, *Information Sciences.* Forthcoming.

5. Kovalerchuk B. 1996b. Second interpolation for fuzzy control. In *Fifth IEEE international conference on fuzzy systems*. New Orleans. Forthcoming.

6. Kovalerchuk B. 1996b. Linguistic context spaces: necessary frames for correct approximate reasoning. *Int. Journal of General Systems* 24(4). Forthcoming.

7. Kovalerchuk, B. 1994a. Advantages of exact complete context for fuzzy control. In *Proceedings of the Joint conference on Information Sciences,* 448-449. Pinehurst, NC.

8. Kovalerchuk B., Yusupov H., and Kovalerchuk N. 1994b. Comparison of interpolations in fuzzy control. In *2nd IFAC workshop on computer software structures integrating AI/KBS Systems in Process Control,* 76-81. Lund, Sweden.

9. Kovalerchuk, B. and Klir, G. 1995. Linguistic context spaces and modal logic for approximate reasoning and fuzzy-probability comparison. In *ISUMA-NAFIPS'95, a23-a28.* College Park, MD.

10.Kovalerchuk, B. 1975. On a method for decision rule selection. *Computing systems* 61:43-50. Novosibirsk. (in Russian).

11. Krantz, D.H., Luce, R.D., Suppes, P., and Tversky A. 1971. *Foundations of measurement,* v.1. AP, NY,

12. Mouzouris, G., and Mendel, J. 1996. Designing fuzzy logic systems for uncertain environments using a singular-value-QR decomposition method. *Fifth IEEE international conference on fuzzy systems. New Orleans. Forthcoming.*

13. Vityaev, E., and Moskvitin, A. 1993. Introduction to discovery theory. *Computational Systems* 148: 117-163. (in Russian).

14. Vityaev, E.E. 1976. Methods for discovery of regularities and forecasting. *Computing Systems* 67:54-68. (in Russian)

15. Vityaev, E.E., Kolchanov, N.A., and Solov'ev S.S. 1980. Discovery of regularities in secondary structures of proteins. In *Computers in molecular and chemical spectroscopy*, 133-134. Institute of Cytology and Genetics, Novosibirsk. (in Russian)

16. Vityaev, E.E. 1992. Semantic approach to knowledge base systems: semantic probabilistic inference of the best predictive PROLOG-programs. *Computing Systems* 146:19-49. Novosibirsk. (in Russian).

17. Zagoruiko, N.G. 1981. *Methods for discovering regularities*. Moscow, Znanie. (in Russian)

18. Zagoruiko, N.G., Samokhvalov K.F., and Sviridenko D.I. 1978. Logic of empirical research. Novosibirsk. (in Russian)