9-1989

Pattern Classification in Dynamic Environments: Tagged Feature-Class Representation and the Classifiers

Qiuming Zhu
University of Nebraska at Omaha, qzhu@unomaha.edu

Pattern Classification in Dynamic Environments: Tagged Feature-Class Representation and the Classifiers

QIUMING ZHU, MEMBER, IEEE

Abstract—**The classifiers characterized by a tagged feature-class representation, a univariate discrimination approach, a cooperative classification scheme, and a logic-based learning strategy are discussed. Neither of the classifiers bears the constraints to the fixed sets of features and classes. Concepts of the tagged feature-class representation and the properties of feature matching in the dynamic environment are studied. Experimental tests and results of the classifiers are illustrated.**

I. Introduction

Traditional statistical pattern recognition system has the following attributes of the classifier determined during the process: 1) a set of classes to which sample patterns are to be assigned; and 2) a set of features by which sample patterns are evaluated and categorized. Outcome of a classification in such system is a unique assignment of the sample pattern to one of the known classes [1], [3], [5], [6].

The environments of classification, however, do not possess such static behavior in many real world problems. Answers to questions of how many classes and what classes the problem has, how many features and what features are presented in the problem are not always predefinable. Examples can be found in visual perception of unexplored scenes, trouble shooting and medical diagnosis, speech and natural language processing, etc. In a broader sense, many rule-based expert systems behave the same way. The activation of the decision rules in "condition ⇒ action" form can be viewed as the consequences of the recognition of the condition patterns, as the features, to the rules, as the classes. The decisionmaking or problem solving is then a sequence of such pattern matching and classification processes. It can not be expected that a perfect satisfaction of the condition set of the to be activated rule will always be granted by the fact set presented. There are many occasions where the knowledge built in the system demands continual up-dating and improvement. Therefore both the classes and features must undergo continual changes. We call those attribute varying situations the dynamic environment of pattern recognition. Pattern classification systems operating in these environments must accommodate the incompleteness and uncertainties of those class and feature presentations, and be able to adapt to the changes of the class and feature attributes. The manipulation must be made by gaining knowledge of the environment from the classification practice rather than a prescheduled scheme.

In traditional statistical pattern recognition, features to distinguish various classes are represented as a feature vector, denoted as $\underline{X}$ [3], [5], [6]. The multidimension space spanned by the possible occurrences of the feature vectors is called the feature space, denoted as $\Omega(\underline{X})$. Clusters of feature vectors form a partition of this feature space. These partitions, designated as classes, are collections of objects with high intra-class and low interclass similarities of the evaluation of these feature vectors. The surfaces, called decision boundaries, of making these partitions are represented by the discriminant function, $g_i(\underline{X})$s. The $g_i(\underline{X})$ can be linear, piecewise linear, or nonlinear. The central problem of those systems is to find and formulate such functions [7], [11]. An optimal classifier that minimizes the probability of error can be obtained by applying the Bayes decision theory.

Manuscript received November 3, 1987; revised November 29, 1988 and March 18, 1989.

Q. Zhu is with the School of Engineering and Computer Science, Oakland University, Rochester, MI 48309-4401.

However the techniques do not allow to be applied to the situations where the sets of classes and features have changes.

A bit-mapped classifier is studied by Frey [4]. The work is originated on Holland's introduction of bit-tagging notation of classifier systems [9]. In Holland's work, a bit defined over the alphabet $\{1, 0, \#\}$ signals the presence, absence, and the "don't care" of the features (messages). Frey demonstrated that the idea is applicable to general pattern classification systems in [4]. It allows the variation of appearance of the number of features to a classifier. In this correspondence, we extend the notion to the domain where features possess statistical distributions. The tag attached to each feature represents the availability and the reliability of the feature value or distribution function. We show that these tagged features and the associated classes form a tagged feature-class space for the classification [12]. A univariate- and a cooperative-classifier working on the tagged feature-class representation are developed respectively. Instead of categorizing samples into classes by the discriminant functions defined on the fixed sets of classes and features, classifications are made from rejecting the inconsistent classes out of a candidate set according to the high intraclass and low interclass measurements. The most significant characteristic of these classifiers is that they bear no constriction to the variations of the sets of classes and features in the system.

The following sections are so organized. Section II introduces the notations of the tagged feature-class representation. Section III discusses the univariate distinguishability of classes and presents an univariate-classifier. A cooperative-classifier is described in Section IV. The natural of the dynamic environment makes the system strongly learning oriented. The learning procedures for the classifiers are presented in Section V. Section VI illustrates the experiment results for the univariate and cooperative-classifiers in tagged feature-class representation. Section VII is conclusion remark.

II. Tagged Feature-Class Representation

A. Feature Characteristics

In a dynamic environment, features in the feature vector $\boldsymbol{f}_s$ of a sample s may have the situations of:

1) a feature f^i is presented in sample s. It is tagged as 1. We use f_s^i to denote the feature and its value. $|f_s^i|$ denotes the tag, therefore, $|f_s^i| = 1$.
2) a feature f^i is not presented in sample s. We do not have value of this feature. When it is referred to, however, it is tagged as 0, i.e., $|f_s^i| = 0$.

A class c is called an established class such that it has been entered to and defined in the system from the previous classification process. We also call it old class to distinguish from the new class that was just introduced by a sample pattern but has not been verified by the system yet.

The appearance of features in feature vector f_c of an established class c has the cases of:

1) a feature f^i is presented for classification in class c. It is tagged as 1. We use f_c^i to denote the feature and $p(f_c^i)$ its probability density function. $|f_c^i|$ denotes the tag. Therefore, $|f_c^i| = 1$.
2) a feature f^i is not presented in class c. We have no probability density function for this feature. When it is referred, however, it is tagged as 0, i.e., $|f_c^i| = 0$.
3) a feature f^i is presented in class c but is declared as an uncertain feature. It is tagged as $\#$. We use f_c^i to denote this feature and $p(f_c^i)$ its probability density function. $|f_c^i|$ denotes the tag. Therefore $|f_c^i| = \#$.

A feature tagged $\#$ represents the uncertainty of its role in the classification. It may be a new feature just introduced without verification yet, or an old feature but its strength is too weak to be used for confirmative classifications.

The $p(f_c^i)$ is established sequentially in the system running process by a learning algorithm. Use μ_c^i and σ_c^i to denote its parameters. A Gaussian density function is assumed when no *a priori* knowledge about the form of the distribution is available. That is

$$p(f_c^i) = G(\mu_c^i, \sigma_c^i) = \frac{1}{\sqrt{2\pi\sigma_c^i}} \exp\left(-\frac{1}{2}\left(\frac{f_c^i - \mu_c^i}{\sigma_c^i}\right)^2\right).$$

We assume that the f_c^i's are mutally independent. The multidimensional probability density function of $\boldsymbol{f}_c$ is then the multiplication of the densities:

$$p(\boldsymbol{f}_c) = \Pi\, p(f_c^i).$$

A strength measure is associated with each f_c^i, denoted as $\mathrm{str}(f_c^i)$. It records and indicates the validity and usefulness of the feature in the classification process.

B. Univariate Discrimination

We call it a correspondence from f_s^i, feature f^i of sample s, to f_c^i, feature f^i of class c, when f_s^i is tagged 1 and f_c^i is tagged 1 or $\#$. Denote the correspondence by $=$, then

$$\forall i \forall c\left[\left((|f_s^i| = 1) \quad \text{and} \quad ((|f_c^i| = 1) \vee (|f_c^i| = \#))\right) \Rightarrow (f_s^i = f_c^i)\right]$$

When $f_s^i = f_c^i$, a matching degree of them, denoted as $\boldsymbol{d}_m(f_s^i, f_c^i)$, is defined as the value of f_s^i on $p(f_c^i)$. That is

$$d_m(f_s^i, f_c^i) = p(f_c^i = f_s^i).$$

Normalize the $d_m(f_s^i, f_c^i)$ to range 0 to 1, we get

$$d_m(f_s^i, f_c^i) = 1 \qquad \text{if } f_s^i = \mu_c^i;$$
$$0 \leqslant d_m(f_s^i, f_c^i) < 1, \qquad \text{otherwise.}$$

The matching degree $d_m(f_s^i, f_c^i)$ measures the certainty of the sample feature f_s^i falling in the distribution region of class feature f_c^i. It can also be viewed as a membership measurement of f_s^i with respect to the category of f_c^i, as that developed in fuzzy set theory [8], [10]. To the Gaussian density of $p(f_c^i)$, the $d_m(f_s^i, f_c^i)$ then is

$$d_m(f_s^i, f_c^i) = \sqrt{2\pi}\,\sigma_c^i p(f_c^i = f_s^i) = \exp\left(-\frac{1}{2}\left(\frac{f_c^i - \mu_c^i}{\sigma_c^i}\right)^2\right).$$

We call it an *inclusion*, denoted by $\propto$, such that f_s^i and f_c^i are correspondent and the $d_m(f_s^i, f_c^i)$ is greater than a specified threshold. That is

$$\forall i \forall c\left[\left((f_s^i = f_c^i) \quad \text{and} \quad (d_m(f_s^i, f_c^i) \geqslant \xi_c^i)\right) \Rightarrow (f_s^i \propto f_c^i)\right]$$

where ξ_c^i is called the inclusion threshold.

On the other hand we have an exclusion, denoted as $\tilde{\propto}$, such that

$$\forall i \forall c\left[\left((f_s^i = f_c^i) \quad \text{and} \quad (d_m(f_s^i, f_c^i) < \xi_c^i)\right) \Rightarrow (f_s^i \,\tilde{\propto}\, f_c^i)\right].$$

The value of $d_m(f_s^i, f_c^i)$ can be attached to the inclusion measurement when it is necessary. Such as

$$f_s^i \overset{0.6}{\propto} f_c^i$$

denotes the inclusion of f_s^i to f_c^i with $d_m(f_s^i, f_c^i) = 0.6$.

The value range of f_c^i on which $f_s^i \propto f_c^i$ is called the discriminant scope of feature f_c^i, denoted as $\mathrm{dis}(f_c^i)$. The $\mathrm{dis}(f_c^i)$'s of a class c regulate the decision region of that class. The probability of error of a classification on class c is then also regulated and monitored by the combination of the ξ_c^i's set to each features of the class c. (An illustration is shown in Fig. 1.)

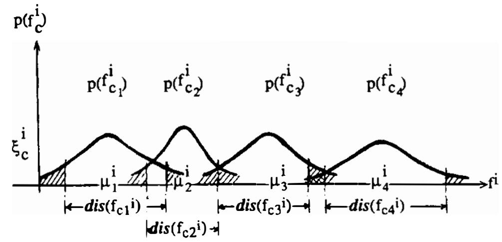


Fig. 1. Illustration of discriminant scopes.

When the probability density functions $p(f^i_{c_k})$ and $p(f^i_{c_j})$ for feature f^i of class c_j and c_k are available, the inclusion threshold $\xi^i_{c_j}$ and $\xi^i_{c_k}$ can be determined by the way such that: for feature value f^i,

$$p\left(f^i_{c_j}=f^i\right)=p\left(f^i_{c_k}=f^i\right)=\xi^i \quad \text{and} \quad \xi^i=\xi^i_{c_j}=\xi^i_{c_k}.$$

III. Univariate Distinguishability and the Univariate Classifier

Conventional pattern classification process can be viewed as a series of transformations that convert the feature vector $\boldsymbol{f}_s$ from n-dimensional feature space $\Omega(\underline{\boldsymbol{X}})$ to a one-dimensional decision surface. Such as

$$g_i=T_i(\boldsymbol{X}); \qquad \boldsymbol{X}=(x_1,x_2,x_3,\cdots,x_n)$$

$$\forall j \exists i\left[(g_i>g_j)\Rightarrow(\boldsymbol{X}\rightarrow c)\right]$$

where c_i denotes a class i. The symbol $\rightarrow$ stands for "is assigned to." The classification relies on the evaluation of the prespecified set of features presented in the sample. Such transformations can not be established in dynamic environment because the variables of the transformation are not able to be specified due to the unpredictable appearance of the x_i's in the input pattern.

One way for pattern classification in dynamic environment is to apply an univariate sequential classifier [12]. The principle of the classifier bases on the univariate distinguishability of the tagged feature-class representation.

We say that two classes c_j and c_k are univariately distinguishable if there exists one feature f^i in $\boldsymbol{f}_{c_j}$ and $\boldsymbol{f}_{c_k}$ such that

$$\left(|f^i_{c_j}|=1\right) \quad \text{and} \quad \left(|f^i_{c_k}|=1\right) \quad \text{and} \quad \left(\operatorname{dis}\left(f^i_{c_j}\right)\cap\operatorname{dis}\left(f^i_{c_k}\right)=0\right).$$

Classes in class set $\{C\}$ are said to be univariately distinguishable if for any pair of classes in $\{C\}$ there exists one feature f^i presented in both classes that makes them univariately distinguishable. On the other hand, if two classes are univariately distinguishable, than a sample pattern belonging to one of the two classes can be uniquely classified by using only that one feature. We call the feature that makes two classes c_j and c_k univariately distinguishable the discriminant feature of these two classes, denoted as $f^d(c_j,c_k)$. A geometric interpretation of univariately distinguishability in the feature-class space is that the decision boundary between the two classes is perpendicular to one feature axis.

Problem is how to find the discriminant features for any given class pair. For the task of distinguishing a sample from classes, fortunately, the explicit identification of these discriminant features is not necessary. The problem is solved by the univariate-classification procedure stated as the following.

A. Procedure Univariate-Classification

1) Form a candidate set $\{C\}=\{c_1,c_2,\cdots,c_n\}$, which contains all classes so far established in the system.
2) For every feature f^i_s of sample s and corresponding feature f^i_c of class c in $\{C\}$, if
$$\left(|f^i_s|=1\right) \quad \text{and} \quad \left(|f^i_c|=1\right) \quad \text{and} \quad \left(f^i_s \tilde{\propto} f^i_c\right)$$
then remove c from $\{C\}$.
3) Assign sample s to the remaining classes in $\{C\}$.

The univariate-classification is a linear operation, which means that it is additive. Therefore we can apply it successively to the candidate set with the use of different features to classify the sample pattern. The univariate discriminant rule is formally stated as

$$\exists i\left[\left(f^i_s \tilde{\propto} f^i_c\right)\right]\Rightarrow\left(s\tilde{\rightarrow}c\right)$$

where $\tilde{\rightarrow}$ stands for not assigned to. On the other hand we have the Theorem 1.

Theorem 1

$$\forall j \exists i\left[\left(f^i_s \propto f^i_{c_k}\right) \quad \left(\operatorname{dis}\left(f^i_{c_j}\right)\cap\operatorname{dis}\left(f^i_{c_k}\right)=0\right)\right.$$
$$\left.\left(|f^i_{c_j}|=1\right)\right] \quad \forall i\left[\left(f^i_s \propto f^i_{c_k}\right)\right]\Rightarrow\left(s\rightarrow c_k\right).$$

Proof: Consider that each feature f^i defines a one-dimensional subspace $\Omega(f^i)$ in the n-dimensional class-feature space $\Omega(\boldsymbol{f})$, where n is the dimension of the feature vector. First we have in every $\Omega(f^i)$ $f^i_s \propto f^i_{c_k}$. Secondly let us take any class $c_j \in \{C\})$ and $(c_j \neq c_k)$, since $\operatorname{dis}(f^i_{c_j})\cap\operatorname{dis}(f^i_{c_k})=0$, $f^i_s \tilde{\propto} f^i_{c_j}$ must be true. That is, c_j is rejected from $\{C\}$. In this case, we say f^i is discovered as a discriminant feature of c_k and c_j. Continuing this process, classes except c_k therefore will be rejected from $\{C\}$ after examining all the features presented. Thus a correct assignment is made to c_k.

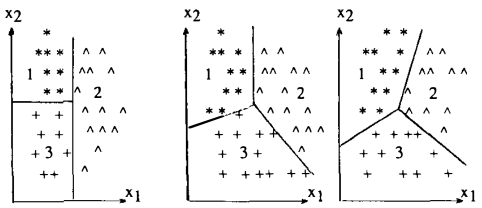
The critical condition for an unique classification to be made by the univariate-classifier is that the discriminant features of the assigned class against all others must be presented in the sample feature vector. Classes are said that are paritally univariately distinguishable if there exists some discriminant features that make a subset of the classes univariately distinguishable. In many real situations, classes are only partially univariately distinguishable.

The univariate-classifier is rejection-natured. Note that in general the previous univariate-classification procedure will be terminated with the outcomes of following three cases.

1) Only one class remaining in $\{C\}$—Sample s is assigned to this class and this class is univariately distinguishable from all others with respect to the features in sample s.
2) No class remain in $\{C\}$—In this case the sample s may be declared as a new class.
3) More than one classes remaining in $\{C\}$—Classes remaining are not univariately distinguishable with respect to the features in sample s.

The sequential evaluation of the features in the univariate-classifier forms a feature chain. When a class is rejected from $\{C\}$ by the evaluation of feature f^i, we say that this class is discriminated at level i. The remaining features of that class is no longer necessary to be evaluated in the process. When a sample s is uniquely assigned to one class by evaluating up to feature f^j in the feature chain, we say that this sample is classified at level j. The efficiency of the procedure can be improved by a frequent reordering of the features in the chain as in the sequential classifiers [3]. Generally features having better discriminant capacity should be evaluated first.

The univariate-classifier is simple and effective in many situations where the classes in system possess the property of univariate distinguishability. It imposes a decision boundary that is perpendicular to one feature axis for any two classes in $\Omega(\boldsymbol{f})$.



(a) (b) (c)

Fig. 2. In 2-D feature space: (a) class 1, 2, and 3 are univariately distinguishable; (b) class 1 and 2 are univariately distinguishable; (c) class 1, 2, and 3 are not univariately distinguishable.

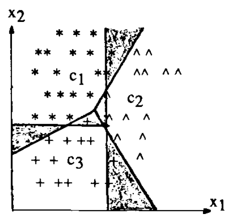


Fig. 3. Errors of classification (marked by dark area) by use of univariate classifier on classes that are not univariately distinguishable.

However such imposition is not generally consistent with the nature of the class-feature distributions for the majority of classification problems. Examples in Fig. 2 show some of the cases. When univariate classifier is applied to the classes that are not univariately distinguishable, larger error rate of classification will be resulted, as it is illustrated in Fig. 3.

The most significant feature of the univariate-classifier is that the classification is made based on the individual appearance of the features. Thus it does not depend on the availability of a prespecified set of features as the g_i does.

IV. The Cooperative Classifier

A cooperative classification method apples a set of measurement of feature matching between $\boldsymbol{f}_s$ and $\boldsymbol{f}_c$ in the classifier. The assignment is not dependent upon the utilization of the discriminant feature and the univariate distinguishability, but rather on the combination of the measurements of the entire features in the feature vector. Following in this section we first describe the measurements and then present the cooperative classification procedure that makes used of these measurements.

A. Sample-Class Consistencies

We use the consistency to measure the matching of feature vector $\boldsymbol{f}_s$ with the feature vector $\boldsymbol{f}_c$. A general-consistency of $\boldsymbol{f}_s$ to $\boldsymbol{f}_c$, denoted as GC, is defined as

$$\forall i\left[\left(|f_c^i|=1\right) \Rightarrow \left(f_s^i \propto f_c^i\right)\right].$$

A general-consistency requires the feature vector $\boldsymbol{f}_s$ to carry all feature f^i's tagged "1" in $\boldsymbol{f}_c$. The inclusion conditions must be satisfied by those features. Sample s is very possibly to be classified as class c when a general-consistency is exhibited. One example is

$\lvert\boldsymbol{f}_c\rvert$:	1	1	1	1	1	1	#	#	#	0	0	0
$\lvert\boldsymbol{f}_s\rvert$:	1	1	1	1	1	1	1	1	0	1	1	0
$\propto$	i	i	i	i	i	i	i	e				

where the first and second row represents the tags of the features in $\boldsymbol{f}_s$ and $\boldsymbol{f}_c$, respectively. On the third row, an i indicates an inclusion of the corresponding features in the column, e stands for an exclusion. Blank means that inclusion measure is not applied.

Let $\|\boldsymbol{f}_s\|$ denote the number of features tagged 1 in $\boldsymbol{f}_s$ and $\|\boldsymbol{f}_c\|$ the number of features tagged 1 in $\boldsymbol{f}_c$. $\|\boldsymbol{f}_c\|^+$ is the number of features tagged 1 or # in $\boldsymbol{f}_c$. We define the consistency value, denoted as C_v, to be the number of inclusions on features $|f_c^i|$ tagged 1. The augmented consistency value, denoted as C_v^+, is the number of inclusions on features $|f_c^i|$ tagged either 1 or #. In the previous example $C_v = \|\boldsymbol{f}_c\| = 6$ and $C_v^+ = 7$. When we have a general consistency, it must have $\|\boldsymbol{f}_s\| \geqslant \|\boldsymbol{f}_c\|$.

The consistency rate of $\boldsymbol{f}_c$ with respect to $\boldsymbol{f}_s$ is defined as $C_r = C_v / \|\boldsymbol{f}_c\|$. The augmented consistency rate of $\boldsymbol{f}_c$ with respect to $\boldsymbol{f}_s$ is defined as $C_r^+ = C_v^+ / \|\boldsymbol{f}_c\|^+$. In a general consistency, C_r always equals to 1 and $C_r^+ \leqslant 1$.

A complete-consistency of $\boldsymbol{f}_s$ to $\boldsymbol{f}_c$, denoted as CC, is defined as

$$\forall i\left[\left(\left(|f_c^i|=1\right) \vee \left(|f_s^i|=1\right)\right) \Rightarrow \left(f_s^i \propto f_c^i\right)\right].$$

A complete-consistency exhibits strong evidence that sample s should be classified as class c. One example is

$\lvert\boldsymbol{f}_c\rvert$:	1	1	1	1	1	1	#	0	1	#	#	0
$\lvert\boldsymbol{f}_s\rvert$:	1	1	1	1	1	1	0	0	1	1	0	0
$\propto$	i	i	i	i	i	i			i	i.		

In this example, $C_v = \|\boldsymbol{f}_c\| = 7$ and $C_v^+ = 8$. In a complete-consistency, C_r always equals to 1 and $C_r^+ \leqslant 1$. The main difference of a complete-consistency from the general-consistency is that no exclusion is allowed for any feature presented.

A *semi-consistency* of $\boldsymbol{f}_s$ to $\boldsymbol{f}_c$, denoted as SC, is defined as

$$\forall i\left[\left(|f_c^i|=1\right) \Rightarrow \left(\left(|f_s^i|=1\right) \Rightarrow \left(f_s^i \propto f_c^i\right)\right)\right].$$

A semiconsistency means that feature vector $\boldsymbol{f}_s$ does not carry all features tagged 1 in $\boldsymbol{f}_c$, neither does the $\boldsymbol{f}_c$ present all features tagged 1 in $\boldsymbol{f}_s$. But inclusion holds if both are tagged 1. Sample s is possible to be classified as class c when a semiconsistency is exhibited. One example is

$\lvert\boldsymbol{f}_c\rvert$:	1	1	1	1	1	0	1	#	#	#	0	0
$\lvert\boldsymbol{f}_s\rvert$:	1	1	1	1	1	1	0	1	1	0	1	0
$\propto$	i	i	i	i	i			i	e.			

In this example, $C_v = 5$ and $C_v^+ = 6$. $C_r = 5/6$ and $C_r^+ = 6/8$. The main feature of a semiconsistency is that if a feature is tagged 1 in both $\boldsymbol{f}_c$ and $\boldsymbol{f}_s$ then the inclusion condition must be satisfied.

Other cases involve with exclusions for features tagged 1 in both $\boldsymbol{f}_c$ and $\boldsymbol{f}_s$. Those cases are called missing-consistency, denoted as MC. One example is

$\lvert\boldsymbol{f}_c\rvert$:	1	1	1	1	1	1	#	#	#	1	0	0
$\lvert\boldsymbol{f}_s\rvert$:	1	1	1	1	1	1	0	1	1	0	1	0
$\propto$	i	i	i	i	e	e		i	e			

We define the missing-consistency value, denoted as MC_v, is the number of exclusions for the features tagged 1 in $\boldsymbol{f}_c$. The augmented missing-consistency value, denoted as MC_v^+, is the number of exclusions for the features tagged either 1 or # in $\boldsymbol{f}_c$. In this example, $MC_v = 2$ and $MC_v^+ = 3$. MC_v always equals 0 for the three consistencies, GC, CC, SC, defined previously. $MC_v^+ = 1$ for both examples in the GC and SC. MC_v^+ always equals 0 for CC.

The missing-consistency rate is defined as $MC_r = MC_v / \|\boldsymbol{f}_c\|$. The augmented missing-consistency rate is defined as $MC_r^+ = MC_v^+ / \|\boldsymbol{f}_c\|^+$. In this example, $MC_r = 2/7$ and $MC_r^+ = 3/10$.

The general situations of the types of consistency with respect to the measurements are listed in Table I.

B. The Cooperative-Classification Procedure

The cooperative-classification procedure uses the measurements previously defined to determine the assignment of sample s. Let $\{C\}$ be the set of all established classes. Subsets of $\{C\}$,

TABLE I

	C_v	C_r	C_v^+	C_r^+	MC_v	MC_r	MC_v^+	MC_r^+
GC	$\|f_c\|$	1	$\geqslant \|f_c\|$ $\leqslant \|f_c\|^+$	$\leqslant 1$	0	0	$\leqslant (\|f_c\|^+ - \|f_c\|)$	<1
CC	$\|f_c\|$	1	$\geqslant \|f_c\|$ $\leqslant \|f_c\|^+$	$\leqslant 1$	0	0	0	0
SC	$\leqslant \|f_c\|$	<1	$\leqslant \|f_c\|^+$	<1	0	0	$\leqslant (\|f_c\|^+ - \|f_c\|)$	<1
MC	$\leqslant \|f_c\|$	<1	$\leqslant \|f_c\|^+$	<1	$\leqslant \|f_c\|$	<1	$\leqslant \|f_c\|^+$	<1

named $\{GC\}$, $\{CC\}$, $\{SC\}$, $\{MC\}$ will be constructed that contain the classes having GC, CC, SC, and MC matching with the sample s, respectively. It is noted that

$$\{GC\} \cup \{CC\} \cup \{SC\} \cup \{MC\} = \{C\}.$$

An active set, denoted as $\{AC\}$, will be used in the procedure. $\|AC\|$ will be used to denote the number of elements in the set AC.

Procedure Coop-Classifier

1) IF not empty of $\{CC\}$, THEN $\{AC\} = \{CC\}$,
   ELSE IF not empty of $\{GC\}$, THEN $\{AC\} = \{GC\}$,
   ELSE IF not empty of $\{SC\}$, THEN $\{AC\} = \{SC\}$,
   ELSE $\{AC\} = \{MC\}$.
2) Apply following rules on $\{AC\}$:
   $\forall c_j[\exists c_i[(c_j \neq c_i)$ and $(C_r(c_j) < C_r(c_i))] \Rightarrow (c_j \notin \{AC\})]$,
   $\forall c_j[\exists c_i[c_j \neq c_i)$ and $(C_v(c_j) * C_r(c_j) < C_v(c_i) * C_r(c_j))] \Rightarrow (c_j \notin \{AC\})]$,
   $\forall c_j[\exists c_i[(c_j \neq c_i)$ and $(C_r^+(c_j) < C_r^+(c_i))] \Rightarrow (c_j \notin \{AC\})]$,
   $\forall c_j[\exists c_i[(c_j \neq c_i)$ and $(C_v^+(c_j) * C_r^+(c_j) < C_v^+(c_i) * C_r^+(c_i))] \Rightarrow (c_j \notin \{AC\})]$,
   $\forall c_j[\exists c_i[(c_j \neq c_i)$ and $(C_v^+(c_j) * C_r^+(c_j) < C_v^+(c_i) * C_r^+(c_i))] \Rightarrow (c_j \notin \{AC\})]$,
   $\forall c_j[\exists c_i[c_j \neq c_i)$ and $(MC_r(c_j) > MC_r(c_i))] \Rightarrow (c_j \notin \{AC\})]$,
   $\forall c_j[\exists c_i[c_j \neq c_i)$ and $(MC_v(c_j) * MC_r(c_j) > MC_v(c_i) * MC_r(c_i))] \Rightarrow (c_j \notin \{AC\})]$,
   $\forall c_j[\exists c_i[(c_j \neq c_i)$ and $(MC_r^+(c_j) > MC_r^+(c_i))] \Rightarrow (c_j \notin \{AC\})]$,
   $\forall c_j[\exists_{c_i}[(c_j \neq c_i)$ and $(MC_v^+(c_j) * MC_r(c_j) > MC_v^+(c_i) * MC_r^+(c_i))] \Rightarrow (c_j \notin \{AC\})]$.
3) If $\{AC\} = \{MC\}$, THEN for all $c_j[MC_r(c_j) < T_{MC_r})] \Rightarrow (c_j \notin \{AC\})]$, where T_{MC_r} is a threshold of missing-consistency rate.
4) Sample s is assigned to $\{AC\}$.

Again we see that the cooperative-classification procedure is rejection-natured. The outcome of the procedure will also have three different situations:

1) $\|AC\| = 1$,
2) $\|AC\| = 0$,
3) $\|AC\| > 1$.

For the Case 1), sample s is uniquely assigned to a class. Case 2) usually results a new class being declared. Case 3) needs more treatment. We may declare that those classes are all possible assignments of s. To make an unique assignment, other measurements, such as the $\boldsymbol{dm}(f_s', f_c')$ can be used to further discriminate the classes in $\{AC\}$.

V. Learning Procedures of the Classifiers

The quality of the classification in dynamic environment heavily depends on the ability of the learning procedures that make the system be able to adjust to the feature and class variations. The following learning procedures are applicable to both the univariate-classifier and the cooperative classifier. Before getting into the description of the learning procedures, we define the concept of mismatching of the classification that is to be used in our learning algorithms.

A. Sample-Class Mismatching (MM's)

When a sample s comes to the classification system in the dynamic environment, an assignment is attempted to be made by the classifier towards the classes established. We call it a mismatching between the f_s and the f_c's, denotes as MM, when the sample is assigned incorrectly. The mismatching can be categorized into following three types.

1) MMI—Maladapted-matching. Sample s is mistakenly declared as a new class n but actually it belongs to an old class c.
2) MMII—Malapropos-matching: Sample s is mistakenly assigned to an old class k but it actually belongs to a) another old class c, or b) a new class n which has not been established yet.
3) MMIII—Maladroit-matching: Sample s is assigned to a subset $\{Ck\}$ that contains more than one class. The actual class of sample s may belong to a) an old class c in the $\{Ck\}$, b) an old class c out of the $\{Ck\}$, c) a new class n not established in the system yet.

The key issue for the system to adapt to the feature and class variations in the dynamic environment is to detect, identify, and eliminate these mismatchings. The patterns of mismatching can be identified by the analysis of the current classification result with the history of the classification process and assisted by a supervised learning process [2], [5]. In our learning processes for the classifiers operating in dynamic environment, the task control is directed to the corresponding procedures for each mismatching pattern identified.

B. Exploration of the Classification Space

Two underlying procedures are used in the learning procedures that handle individual patterns of mismatching.

1) An inclusion procedure on feature f' that achieves the result of $[(s \rightarrow c) \Rightarrow (f_s' \propto f_c')]$.
2) An exclusion procedure on feature f' that achieves the results of $[(s \tilde{\rightarrow} c) \Rightarrow (f_s^i \bar{\propto} f_c')]$.

We use the value of $\mathrm{str}(f_c')$ to signal the switch of feature tags among 0, 1 and # in the inclusion and exclusion procedures. The two procedures are described next.

Procedure Inclusion (f_s', f_c')

IF $(|f_c'| <> 0)$ AND $(|f_s'|| <> 0)$ THEN
   $\mathrm{str}(f_c') = \mathrm{str}(f_c') + 1$,
   REPEAT
      $\mu_c' = \mu_c' + d_m(f_s', f_c') * (f_s' - \mu_c')$,
      $\sigma_c' = \sigma_c' + d_m(f_s', f_c')) * (f_s' - \mu_c')^2$
   UNTIL $p(f_s') \geqslant \xi_c'$
ELSE IF $(|f_s'| = 0)$ THEN
      $\mu_c' = f_s'$,
      $\sigma_c^i = M_\sigma$,
      $|f_c'| = \#$,
      $\mathrm{str}(f_c') = 1$;
IF $(|f_c'| = \#)$ and $(\mathrm{str}(f_c') > M_{\mathrm{strength}})$ THEN $|f_c'| = 1$.

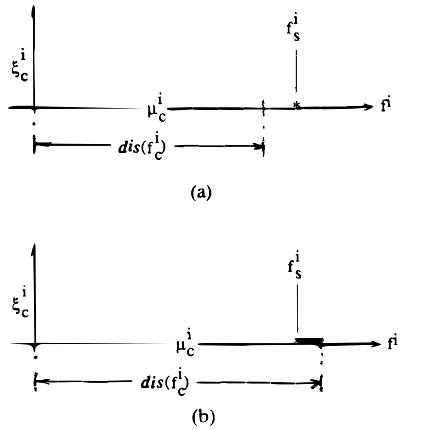


Fig. 4. Feature density function adjusted in procedure Inclusion. (a) Before adjustment. (b) After adjustment.

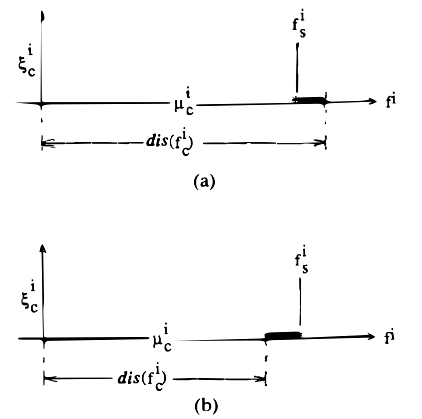


Fig. 5. Feature density function adjusted in procedure Exclusion. (a) Before adjustment. (b) After adjustment.

The M_{strength} in the procedure is a specified threshold for switching the tag between 1 and #. M_σ is a prespecified constant.

An illustration of the Inclusion process is shown in Fig. 4. Fig. 4(a) depicts the situation of sample feature f_s^i and the probability density of f_c^i before the call of the procedure and Fig. 4(b) depicts the situation after the execution of the procedure.

Procedure Exclusion (f_s^i, f_c^i)

IF $(|f_c^i| <> 0)$ AND $(|f_s^i|| <> 0)$ THEN

$\text{str}(f_c^i) = \text{str}(f_c^i) - 1,$

REPEAT

$\mu_c^i = \mu_c^i - d_m(f_s^i, f_c^i) * (f_s^i - \mu_c^i),$

$\sigma_c^i = \sigma_c^i - d_m(f_s^i, f_c^i)) * (f_s^i - \mu_c^i)^2$

UNTIL $p(f_s^i) < \xi_c^i$

IF $(|f_c^i| = \#)$ and $(\text{str}(f_c^i) < -M_{\text{strength}})$ THEN $|f_c^i| = 0$, $\text{str}(f_c^i) = 1$,

IF $(|f_c^i = 1)$ and $(\text{str}(f_c^i) < -M_{\text{strength}})$ THEN $|f_c^i| = \#$, $\text{str}(f_c^i) = 1$.

An illustration of the exclusion procedure is shown in Fig. 5. Fig. 5(a) depicts the situation before the call of the procedure, and Fig. 5(b) depicts the situation after the execution of the procedure.

The following are descriptions of the learning procedures for cases distinguished by the *MM*'s.

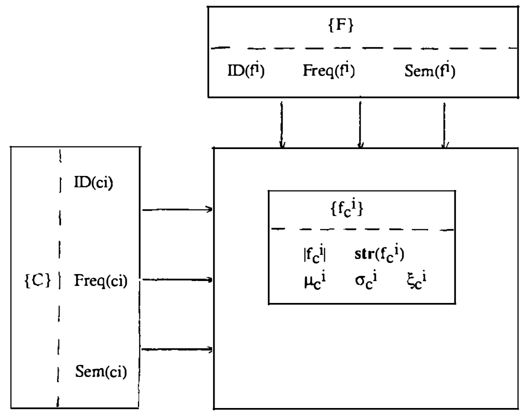


Fig. 6. Matrix representation of feature-class space.

1) An MMI is identified such that sample s is recognized as class c rather than a new class n:
   for all features with $|f_s^i| = 1$, call Inclusion (f_s^i, f_c^i).
2) An MMII is identified such that the sample s is recognized as a new class n rather than an old class k.
   - Add n to $\{C\}$. For all features with $|f_s^i| = 1$, call Inclusion(f_s^i, f_n^i).
   - Choose feature f_s^i such that $(f_s^i \propto f_k^i)$ and for all $j[\xi_c^i < d_m(f_s^i, f_k^i) < d_m(f_s^j, f_k^j)]$, call Exclusion$(f_s^i, f_k^i)$.
3) An MMIII is identified such that sample s is recognized as an old class c rather than an old class k.
   - For all features with $|f_s^i| = 1$, call Inclusion(f_s^i, f_c^i).
   - Choose feature f_s^i such that $(f_s^i \propto f_k^i)$ and for all $j[\xi_c^i < d_m(f_s^i, f_k^i) < d_m(f_s^j, f_k^j)]$, call Exclusion$(f_s^i, f_k^i)$.
4) An MMIII is identified such that sample s is recognized as a new class n.
   - Add n to $\{C\}$. For all features with $|f_s^i| = 1$, call Inclusion(f_s^i, f_n^i).
   - For every class k in $\{Ck\}$: Choose feature f_s^i such that $(f_s^i \propto f_k^i)$ and for all $j[\xi_c^i < d_m(f_s^i, f_k^i) < d_m(f_s^j, f_k^j)]$, call Exclusion$(f_s^i, f_k^i)$.
5) An MMIII is identified such that sample s is recognized as an old class c in $\{Ck\}$.
   - For all features with $|f_s^i| = 1$, call Inclusion(f_s^i, f_c^i).
   - For every other class k in $\{Ck\}$ except c. Choose feature f_s^i such that $(f_s^i \propto f_k^i)$ and for all $j[\xi_c^i < d_m(f_s^i, f_k^i) < d_m(f_s^j, f_k^j)]$, call Exclusion$(f_s^i, f_k^i)$.
6) An MMIII is identified such that sample s is recognized as an old class c out of $\{Ck\}$.
   - For all features with $|f_s^i| = 1$, call Inclusion(f_s^i, f_n^i).
   - For every class k in $\{Ck\}$: Choose feature f_s^i such that $(f_s^i \propto f_k^i)$ and for all $j[\xi_c^i < d_m(f_s^i, f_k^i) < d_m(f_s^j, f_k^j)]$, call Exclusion$(f_s^i, f_k^i)$.

VI. Experimentation

Experiments are conducted on the tagged feature-class representation and the two classifiers for pattern classification in dynamic environment we described previously.

We maintain a feature set $\{F\}$ and a class set $\{C\}$ for learning in the classification system. Element in $\{F\}$ is a 3-tuple that specifies the identification of the feature $\text{ID}(f^i)$, the frequency of the appearance $\text{Freq}(f^i)$, and the semantics of the feature $\text{Sem}(f^i)$. Element in $\{C\}$ contains also a 3-tuple that specifies the identification of the class $\text{ID}(c_i)$, the frequency of appearance $\text{Freq}(c_i)$, and the semantics of the class $\text{Sem}(c_i)$. The product of $\{F\}$ and $\{C\}$ forms the set $\{f_c^i\}$ that specifies the

TABLE II
EXPERIMENT RESULTS USING UNIVARIATE CLASSIFIER

	Test 1 4 Classes 4 Features	Test 2 4 Classes 8 Features	Test 3 8 Classes 4 Features	Test 4 8 Classes 8 Features	Test 5 4 Classes 16 Features	Test 6 8 Classes 16 Features
Group 1	4	10	8	4	1	0
Group 2	9	8	11	12	3	2
Group 3	4	10	8	9	4	4
Group 4	8	6	4	9	4	1
Group 5	7	7	1	5	3	3
Group 6	4	4	3	6	0	4
Group 7	3	5	2	7	2	2
Group 8	6	4	8	6	0	2
Group 9	7	3	4	5	0	2
Group 10	2	3	3	4	2	0
Group 11	1	4	3	6	1	1
Group 12	3	2	2	4	0	2
Group 13	2	2	3	3	1	0
Group 14	3	3	2	2	1	1
Group 15	2	2	3	2	0	0
Group 16	2	2	3	1	0	0

TABLE III
EXPERIMENT RESULTS USING COOPERATIVE CLASSIFIER

	Test 1 4 Classes 4 Features	Test 2 4 Classes 8 Features	Test 3 8 Classes 4 Features	Test 4 8 Classes 8 Features	Test 5 4 Classes 16 Features	Test 6 8 Classes 16 Features
Group 1	4	10	8	4	1	0
Group 2	9	8	11	13	3	2
Group 3	4	8	8	9	0	2
Group 4	7	3	4	9	1	1
Group 5	6	6	1	5	2	0
Group 6	3	2	3	6	0	1
Group 7	2	5	2	6	2	1
Group 8	3	4	2	5	0	2
Group 9	4	1	0	4	0	0
Group 10	2	3	3	4	1	0
Group 11	1	2	1	3	1	1
Group 12	2	2	2	5	0	2
Group 13	2	1	2	3	1	1
Group 14	1	1	1	4	1	1
Group 15	2	1	2	2	0	0
Group 16	1	0	1	2	0	0

features of each class. We call the $\{C\}$, $\{F\}$, and $\{f_c^i\}$ together the feature-class $(F-C)$ space. Each element in $\{f_c^i\}$ has a 5-tuple that specifies the tag of the feature $|f_c^i|$, the strength of the feature $\mathrm{str}(f_c^i)$, the parameters μ_c^i and σ_c^i of $p(f_c^i)$, and the inclusion threshold ξ_c^i. The feature-class space is structurally conformed as a two-dimensional matrix, as it is shown in Fig. 6.

At the initial stage of the classification, the $\{C\}$, $\{F\}$ and $\{f_c^i\}$ are all empty. To do the test, a set of background classes and their feature distributions are randomly generated in the experiments. There is no *a priori* knowledge about these background classes available to the $F-C$ space and the classifiers.

A sample generator is designed to provide inputs to the classifiers. First, a background class, say c_i, is chosen randomly. Sample features are then generated with respect to the prespecified feature distributions of c_i. Noises are added to the sample. A $\{0,1\}$ random number is used to set the tags of the sample features, therefore changes the presences of the sample feature set. These patterns are then classified by the attributes of the classes and features established so far in the $F-C$ space.

The contents of $\{C\}$, $\{F\}$, and $\{f_c^i\}$ are established and augmented in terms of the information carried by the samples when the classification process proceeds. Generally, an introduction of a new class in the sample pattern leads to a new row added to the $F-C$ space; and introduction of a new feature in the sample pattern leads to a new column added to the $F-C$ space. To avoid the overgrowth of the $F-C$ space, obsolete classes must be detected and deleted from $\{C\}$ periodically in a long run of the system. It is done by referring to the frequency measure $\mathrm{Freq}(c_i)$ of the classes. Occasionally two classes may need to be merged to one single-class according to the feature distributions and the situation of the classification. It is the same for the feature set $\{F\}$. These problems are not to be discussed in this correspondence.

Several test results for the univariate and cooperative classifiers are shown in tables II and III, respectively. Each test is organized into 16 groups. Sixteen samples are generated in each group randomly. The contents of the tables list the total number of mismatchings for the sample patterns classified. The maximum number of classes and features in each test is indicated in the tables.

Since the number of misclassifications for the test cases depends on how the background class and feature distributions are set and valued, therefore it does not necessary stand for the precision or accuracy of the classifications. However from the results we can see that the trend of the decreasing of misclassification rate is obvious when the classification process proceeds. It signals the effectiveness of the classifiers along with the use of the learning procedures in such feature and class sets variant environment. Some higher rates in first few groups are caused because most of the classes are just introduced to the classifier at that stage. Note that the set of features available to the classifier is randomly decided. Therefore both the number of features and the appearnace of the features vary continually. The tables also exhibit that the cooperative-classifier has better performance than the univariate-classifier in terms of the error rate of the classifications. However the univariate-classifier needs less computation than the cooperative-classifier.

VII. Conclusion

Pattern recognition is a general purpose task underlying for many application systems. The incompleteness and uncertainty of the feature and class presentation in dynamic environment makes the system be difficult from applying traditional statistical pattern recognition techniques. The configuration of the classifiers operating in such environment must possesses the properties that are distinct from the traditional techniques. In this correspondence we have discussed 1) a tagged feature and class representation of the pattern recognition problem in the dynamic environment; 2) the statistical feature evaluation based univariate- and cooperative-classifiers that bear no constrains to the variations of the sets of classes and features; and 3) the inductive learning procedures that are used to the creation of a class-feature space adaptive to the variations of the dynamic environment.

Rather than trying to formulate the discriminant functions that are defined on the fixed sets of classes and features, the univariate classifier and the cooperative classifier discussed in this paper applies a classify-by-rejection approach on a candidate class set. The classification is based on the individual evaluation of the features presented in the sample patterns and the classes. Statistical distribution properties of the features are retained in the so developed processes. There are many other techniques that can be combined to make the process of pattern classification in dynamic environment more subtle. For example, when the distributions of the class features are all settled or available, discriminant functions defined on the subset of the features can be constructed according to the feature set presented in the sample pattern. These functions have to be reconstructed every time to classify a sample.

The tagged feature-class space permits the building of an hierarchical structure of the classifications conveniently. Because both the classes and features are tagged, they are not necessary to be distinct every where. The outcome of a classification at one level of the process therefore can be coupled to the feature set at another designated level of the hierarchy for making further decisions. When viewing each row of the $F-C$ space as a production rule where the features are the conditions and the classes distinguished as the conclusions, the structure of $F-C$ space permits both value tuning and rule constructions from the learning processes.

A content-addressable data retrieve characteristic is also possessed by the univariate- and cooperative-classifiers. A complete set of features of a class can be recovered from a partial presentation of the features in the sample pattern by accessing the contents of the $F-C$ space. The sample pattern formed by the partial features acts as an index to the complete set of the contents of a class. The application of this property can be found in the database retrieving, the prediction of occluding or missing part of objects in an image, natural language understanding, and various of other applications.

References

[1] P. A. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach.* Englewood Cliffs, NJ: Prentice-Hall, Inc., 1982.

[2] T. G. Dietterich and R. S. Michalski, "Learning to Predict Sequences," in *Machine Learning II.* Los Altos, CA: Morgan Kaufmann Publishers, 1986, pp. 63–106.

[3] O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis.* New York: John Wiley, 1973.

[4] P. W. Frey, "A bit-mapped classifier," *BYTE*, pp. 161–172, Nov. 1986.

[5] K. S. Fu, *Sequential Methods in Pattern Recognition and Machine Learning.* New York: Academic Press, 1968.

[6] K. Fukunaga, *Introduction to Statistical Pattern Recognition.* New York: Academic Press, 1972.

[7] I. Foroutan and J. Sklansky, "Feature selection for automatic classification of non-gaussian data," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-17, no. 2, pp. 187–198, Mar./Apr. 1987

[8] J. M. Keller and D. J. Hunt, "Incorporating fuzzy membership functions into the perceptron algorithm," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-7, no. 6, pp. 693–699, Nov. 1985.

[9] J. H. Holland, "Escaping brittleness: The possibilities of general-purpose learning algorithms applied to parrallel rule-based systems," in *Machine Learning II.* Los Altos, CA: Morgan Kaufmann Publishers, 1986, pp. 593–623.

[10] S. K. Pal, D. K. Dutta, and D. Majumder, *Fuzzy Mathematical Approach to Pattern Recognition.* New Delhi, India: Wiley Eastern Limited 1986.

[11] M. A. L. Thathacher and P. S. Sastery, "Learning optimal discriminant functions through a cooperative game of automata," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-17, no. 1, pp. 73–85, Jan./Feb. 1987.

[12] Q. M. Zhu, "Pattern classification in dynamic environment: Tagged feature-class space and univariate classifier," *Pattern Recognition 1988 (BAPR 4th Int. Conf.*, J. Kittler, (Ed.), Cambridge, England, pp. 517–526.