

# A Mathematically Rigorous Foundation for Supervised Learning \* \*\*

Eugene M. Kleinberg

Department of Mathematics, The State University of New York, Buffalo NY 14214.  
kleinbrg@math.buffalo.edu

**Abstract.** This paper consists of two parts, one theoretical, and one experimental. And while its primary focus is the development of a mathematically rigorous, theoretical foundation for the field of supervised learning, including a discussion of what constitutes a “solvable pattern recognition problem”, it will also provide some algorithmic detail for implementing the general classification method derived from the theory, a method based on classifier combination, and will discuss experimental results comparing its performance to other well-known methods on standard benchmark problems from the U.C. Irvine, and Statlog, collections. The practical consequences of this work are consistent with the mathematical predictions. Comparing our experimental results on 24 standard benchmark problems taken from the U.C. Irvine, and Statlog, collections, with those reported in the literature for other well-known methods, our method placed 1st on 19 problems, 2nd on 2 others, 4th on another, and 5th on the remaining 2.

*Keywords:* machine learning, pattern recognition, classification algorithms, stochastic discrimination, SD, boosting.

## 1 Introduction

We are about to develop the ideas behind a particular approach to solving problems in supervised learning. The method derived from this approach is very general, and algorithmic implementations have produced results which, in most observed cases, are superior to those produced by any other method of which we are aware. And while this should certainly be an important consideration for interest here, we feel that it is the underlying mathematical theory, and the implications of this theory providing a perspective underlying existing work in the field in general, as well as a basis for future work, which merits the greatest attention. As one example of this, we might note that the mathematics we are about to develop provides a complete theoretical explanation for the experimentally observed success of the method of boosting, including the ability of boosting to generalize to unseen data; and, based on this theoretical understanding, provides a clear direction for improvement for future boosting algorithms (see [7, 8]).

---

\* This work uses software copyrighted by K Square Inc

\*\* ©Springer-Verlag

Although we will not present explicit pseudo-code for an algorithmic implementation of our method, we will provide a description sufficient for creating such an implementation. As motivation for the mathematics which will be presented in later sections of this paper, we begin with a discussion of experimental results, comparing our particular algorithmic implementation of the method, henceforth referred to as *SDK*, to other well-know pattern recognition methods. Our use of the word “motivation” here is somewhat nonstandard. It is our hope that readers will find the experimental results for SDK sufficiently promising that they are motivated to thoroughly read the mathematical theory which follows, and use their understanding of it to create their own, hopefully superior, implementations.

Detail concerning our implementation, SDK, can be found in [7]. However, we feel that it might be useful, at this time, to point out the following: SDK operates by first (pseudo) randomly sampling (with replacement) from a space of *subsets* of the feature space underlying a given problem, and then combining these subsets to form a final classifier. There are many ways to contrast this approach with other classification methods, but perhaps the most striking deals with the perspective from which one attacks the problem of establishing theoretical bounds on classifier performance. For when proving theorems concerning the accuracy of classifiers built using SDK, we initially consider probabilities *with respect to the sample space of subsets of the given feature space*, rather than with respect to the feature space itself. It is only by appealing to something know as the *duality lemma* (see [6]), that one can translate these accuracy estimates into standard error rates over the feature space.

## 2 Experimental Results

*The Datasets* We worked with datasets from two major sites containing sets of standardized problems in machine learning, the repository at the University of California at Irvine, and the repository (of Statlog problems) at the University of Porto in Portugal.

We carried out experiments with 17 datasets from the Irvine collection, datasets which seemed to be the most popular appearing in the recent literature dealing with comparative studies of pattern recognition methods. The sets we used were, Australian credit (henceforth abbreviated “crx”), Pima diabetes (dia), glass (gls), Cleveland heart (hrt), hepatitis (hep), ionosphere (ion), iris (iri), labor (lab), letter (let), satimage (sat), segment (seg), sonar (son), soybean-large (soy), splice (spl), vehicle (veh), vote (vot), and Wisconsin breast cancer (wsc). In [3], Freund and Schapire report on experimental results they derived for these problems using 9 different classification methods, namely, three underlying “weak learning algorithms” FindAttrTest (henceforth denoted, “FIA”), FindDecRule (FID), and Quinlan’s C4.5 (C45) (see [10]), the boosted ([3]) versions of these algorithms, denoted ABO, DBO, and 5BO, respectively, and the bagged ([1]) versions, denoted ABA, DBA, and 5BA, respectively. Our learning runs on these datasets used the same study methodologies (either 10-fold cross

validation, or training/test set, depending on the dataset) as used by Freund and Schapire, with the sole change (due to time constraints) of running two of the training/test problems (letter and satimage) only once, using the default seed of 1 in each case.

Of the 10 Statlog sets publicly available from Porto, we eliminated two from consideration (heart and German credit) since they involved nontrivial cost matrices, something SDK is not designed to deal with, and eliminated a third (shuttle) since it was extremely underrepresented in some classes (class seven contained 2 test points out of a sample containing 58,000 training and test examples). On the remaining 7 sets, we carried out training runs using the same study methodologies (either a cross validation, or a training/test set, depending on the dataset) as [9].

*The Results* We compare our results on the Irvine problems with those reported in [3] in Figures 1 and 2. The table shows error rates for each method on each problem, with the *italicized* entry in each row belonging to the method which produced the lowest error rate. And in the graph, we produce for each method, a bar ranging from the best rank to the worst rank for that method across all problems, and place a left tic at the method's average rank, and a right tic at the method's mode. The methods are listed in order of average rank, and we superimpose a line graph showing these average ranks.

In Figures 3 and 4, we basically do the same thing, as we compare our results on the Statlog datasets from Porto with those reported in [9]. (Note the row/column switch in the table.)

Note that the data in Figures 1 and 3 shows that SDK was the best performing method in 14 of the 17 U.C. Irvine experiments, and in 5 of the 7 Statlog experiments.

### 3 The Theory

*The Prototypical Problem* Our first goal is to try to formalize from a foundational mathematical point of view the notion of “building classifiers based on the study of training data”. We assume we are at a point in the process where data has already passed through an initial feature extraction stage and that there exists a fixed positive integer  $n$  such that the objects among which we are interested in discriminating have all been reduced to numeric records of length  $n$ . Conforming to standard practice, we refer to the subspace of Euclidean  $n$ -space in which these records reside as the “feature space” of the problem.

The prototypical supervised learning problem in pattern recognition asks one to build a classifier from “representative” examples. From a mathematical perspective, what does “representative” mean here? Clearly, it would be impossible to proceed with any rigorous development of the theory underlying supervised learning without first answering this question.

	FIA	ABO	ABA	FID	DBO	DBA	C45	5BO	5BA	SDK
crx	14.5	14.4	14.5	14.5	13.5	14.5	15.8	13.8	13.6	12.4
dia	26.1	24.4	26.1	27.8	25.3	26.4	28.4	25.7	24.4	25.5
gls	51.5	51.1	50.9	49.7	48.5	47.2	31.7	22.7	25.7	20.3
hrt	27.8	18.8	22.4	27.4	19.7	20.3	26.6	21.7	20.9	17.4
hep	19.7	18.6	16.8	21.6	18.0	20.1	21.2	16.3	17.5	16.2
ion	17.8	8.5	17.3	10.3	6.6	9.3	8.9	5.8	6.2	6.2
iri	35.2	4.7	28.4	38.3	4.3	18.8	5.9	5.0	5.0	4.2
lab	25.1	8.8	19.1	24.0	7.3	14.6	15.8	13.1	11.3	6.1
let	92.9	92.9	91.9	92.3	91.8	91.8	13.8	3.3	6.8	3.3
sat	58.3	58.3	58.3	57.6	56.5	56.7	14.8	8.9	10.6	8.7
seg	75.8	75.8	54.5	73.7	53.3	54.3	3.6	1.4	2.7	1.9
son	25.9	16.5	25.9	31.4	15.2	26.1	28.9	19.0	24.3	10.6
soy	64.8	64.5	59.0	73.6	73.6	73.6	13.3	6.8	12.2	5.9
spl	37.0	9.2	35.6	29.5	8.0	29.5	5.8	4.9	5.2	4.9
veh	64.3	64.4	57.6	61.3	61.2	61.0	29.9	22.6	26.1	22.1
vot	4.4	3.7	4.4	4.0	4.4	4.4	3.5	5.1	3.6	3.5
wsc	8.4	4.4	6.7	8.1	4.1	5.3	5.0	3.3	3.2	2.6

Fig 1. Experimental Results - Error Rates on Irvine Problems

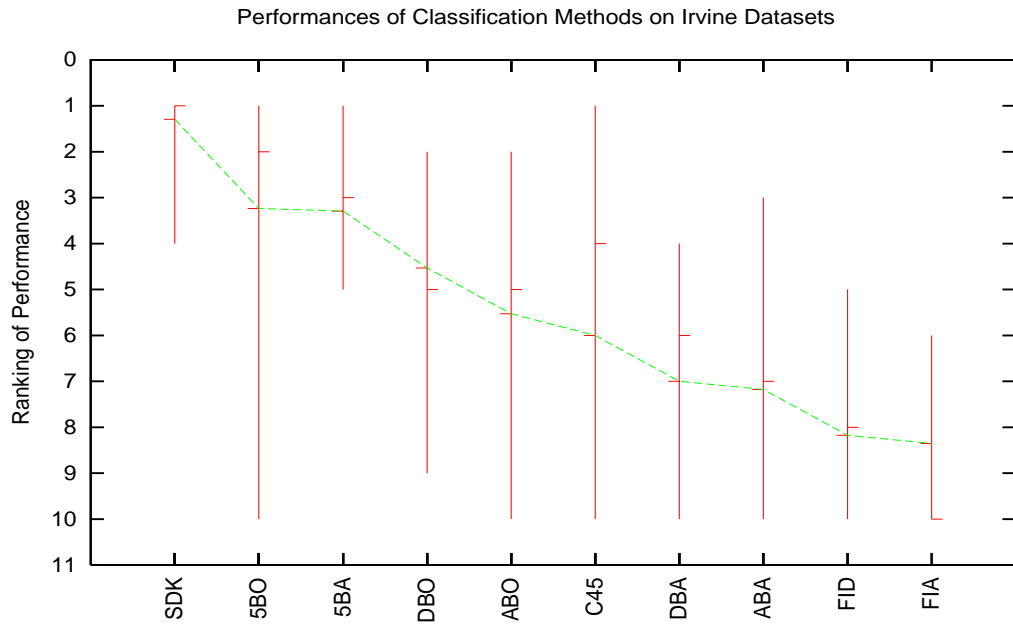


Fig 2. Relative Performance Ranks - Irvine Problems

	crx	dia	dna	let	sat	seg	veh
Ac2	0.181	0.276	0.245	0.245	0.157	0.031	0.296
Alloc80	0.201	0.301	0.064	0.064	0.132	0.030	0.173
BackProp	0.154	0.248	0.327	0.327	0.139	0.054	0.207
BayTree	0.171	0.271	0.124	0.124	0.147	0.033	0.271
Bayes	0.151	0.262	0.529	0.529	0.287	0.265	0.558
C4.5	0.155	0.270	0.132	0.132	0.150	0.040	0.266
Cal5	0.131	0.250	0.253	0.253	0.151	0.062	0.279
Cart	0.145	0.255	NA	NA	0.138	0.040	0.235
Castle	0.148	0.258	0.245	0.245	0.194	0.112	0.505
Cn2	0.204	0.289	0.115	0.115	0.150	0.043	0.314
Default	0.440	0.350	0.960	0.960	0.769	0.760	0.750
Dipol92	0.141	0.224	0.176	0.176	0.111	0.039	0.151

	crx	dia	dna	let	sat	seg	veh
Discrim	0.141	0.225	0.302	0.302	0.171	0.116	0.216
IndCart	0.152	0.271	0.130	0.130	0.138	0.045	0.298
ltrule	0.137	0.245	0.594	0.594	NA	0.455	0.324
KNN	0.181	0.324	0.068	0.068	0.094	0.077	0.275
Kohonen	NA	0.273	0.252	0.252	0.179	0.067	0.340
LVQ	0.197	0.272	0.079	0.079	0.105	0.046	0.287
LogDisc	0.141	0.223	0.234	0.234	0.163	0.109	0.192
NewId	0.181	0.289	0.128	0.128	0.150	0.034	0.298
QuaDisc	0.207	0.262	0.113	0.113	0.155	0.157	0.150
Radial	0.145	0.243	0.233	0.233	0.121	0.069	0.307
SDK	0.126	0.233	0.033	0.038	0.0865	0.021	0.201
Smart	0.158	0.232	0.295	0.295	0.159	0.052	0.217

Fig 3. Experimental Results - Error Rates on Statlog Problems



Fig 4. Relative Performance Ranks - Statlog Problems

In practice, a set  $A$  is usually viewed as being representative of a set  $B$  if it is spatially distributed throughout the region of the feature space occupied by  $B$ . This is a simple, pragmatic, derivative thesis which tends to work to greater or lesser extents based on the specifics of any particular problem being considered. But it is not really what one means by the notion “representative”. Most people

would agree that, from a more fundamental perspective, the intuition is that a set  $A$  is representative of a set  $B$  if given any classifier  $\mathcal{C}$ , the error rate of  $\mathcal{C}$  (for its task of recognition) when measured on members of  $A$  is equal to, or at least close to, its error rate when measured on members of  $B$ . Another way to express this intuition in a more operational, but less precise, way is to simply say that a set  $A$  is representative of a set  $B$  if any classifier  $\mathcal{C}$  built using  $A$  generalizes to  $B$ .

Needless to say this description has serious flaws. For if  $\mathcal{C}$  were the classifier which simply cataloged the points in  $A$ , and then classified any (new) point based on whether or not it sat in this list, then the error rate of that classifier when measured on  $A$  would be 0, yet, assuming  $A$  were substantially smaller than  $B$ , would be substantially larger than 0 when measured on  $B$ .

Thus the notion “ $A$  is representative of  $B$ ” must be dependent on both the sets  $A$  and  $B$ , *and on some expectation concerning the nature of the classifier itself*. In other words, the notion “representative” can never be an absolute; when one is given a particular pattern recognition problem through a training set of examples which are declared to be “representative”, the understanding *must* be that the examples are “representative” only so long as possible classifiers derived as solutions to the problem are restricted to satisfy certain additional requirements.

In most practical applications, there is an implicit assumption that if training sets are sufficiently densely distributed throughout class regions in the feature space, then by seeking classifiers which are restricted to carve out sufficiently “thick” subsets of the feature, such training sets are “representative”. In effect, the assumption is one of spatial proximity of like points between training and test sets.

However, given our desire for generality, we feel that there is a far more elegant, and natural, way to formalize the notion “representative”. We will simply define what it means, given some collection  $\mathbf{M}$  of subsets of the feature space, (intuitively, the building blocks of allowed, possible classifiers) for a subset  $A$  of the feature space to be  $\mathbf{M}$ -representative of another subset  $B$  of the feature space. In this way, although we can encompass the usual proximity-based approach as a special case, we don’t require any topological relationship between training and test sets, and as such allow for a number of interesting alternative possibilities. Most important we feel that this definition constitutes the minimal requirement for “representativeness”.

The underlying idea is very simple. In order for a set  $A$  to be  $\mathbf{M}$ -representative of a set  $B$ , it must be impossible to tell the difference between points in  $A$  and points in  $B$  using the expressive power inherent in the sets of  $\mathbf{M}$ . There is a slight irony here. In pattern recognition one tries to find a solution which must succeed in discriminating between points of different classes, yet one which must simultaneously fail to discriminate between training and test subsets of a given class.

It is this “indiscernibility” between training and test sets modulo the expressive power of sets in  $\mathbf{M}$  which serves as the basis for our development here.

*Indiscernibility and Representativeness* Let  $n$  be a fixed positive integer, and assume that our feature space  $F$  is some fixed, finite subset of Euclidean  $n$ -space. Since  $F$  is finite we can consider it to be a measure space under the counting measure  $\mu$ .

Let us denote by  $\mathbf{F}$ , the power set of  $F$ , that is, the collection of all subsets of  $F$ .

**Definition 1.** For a given collection  $\mathbf{M}$  of subsets of  $F$ , we define a binary relation  $\sim_{\mathbf{M}}$  on the collection of nonempty subsets of  $F$  as follows: for any sets  $A$  and  $B$  contained in  $F$ ,  $A \sim_{\mathbf{M}} B$  iff for every  $M$  in  $\mathbf{M}$ ,  $Pr(M|A) = Pr(M|B)$

(Viewing  $F$  as a sample space,  $Pr(M|A)$  denotes the probability of  $M$  given  $A$ .)

Having  $A \sim_{\mathbf{M}} B$  would certainly appear to be a necessary condition for  $B$  being  $\mathbf{M}$ -indiscernible from  $A$ . But it is easy to construct examples showing that it is not sufficient, that is, examples where  $A \sim_{\mathbf{M}} B$  for distinct sets  $A$  and  $B$ , yet  $\mathbf{M}$  contains information capable of showing  $A \neq B$ .

In order to have true  $\mathbf{M}$ -indiscernibility, it must be the case that any “profile” of  $A$  which can be deduced using information from  $\mathbf{M}$  is identical with a similarly deduced “profile” of  $B$ . Thus consider the following function  $f_{\mathbf{M},x,A}$ , defined for any subset  $\mathbf{M}$  of  $\mathbf{F}$ , any nonempty subset  $A$  of  $F$ , and any real  $x$  for which there exist  $M$  in  $\mathbf{M}$  such that  $Pr(M|A) = x$ , which maps  $A$  into the reals: given any member  $q$  of  $A$ ,

$$f_{\mathbf{M},x,A}(q) = Pr_{\mathbf{M}}(q \in M | Pr(M|A) = x).$$

(Since we will be dealing with several different probability spaces in what follows, there might be times when confusion could arise as to just which space we are taking probabilities with respect to. At times of such potential ambiguity, we will use  $Pr_T$  to denote probabilities taken with respect to the space  $T$ .)

In some sense, the random variable  $f_{\mathbf{M},x,A}$  defines a profile of the coverage of points in  $A$  by those members  $M$  of  $\mathbf{M}$  such that  $Pr(M|A) = x$ . We restrict to those  $M$  such that  $Pr(M|A) = x$  for the sake of simplicity, for there is often a clear expectation of coverage for such  $M$ . For example, if  $\mathbf{M}$  were equal to the full power set of  $F$ , it is fairly easy to see that for any  $q$  in  $A$ ,  $f_{\mathbf{M},x,A}(q) = x$  for any  $x$ .

Using this notation, we are now in a position to precisely define the notion of indiscernibility:

**Definition 2.** Given sets  $A$  and  $B$  contained in  $F$ , and given a collection  $\mathbf{M}$  of subsets of  $F$ , we say that  $A$  is  $\mathbf{M}$ -indiscernible from  $B$  if

- (a)  $A \sim_{\mathbf{M}} B$ ;
- (b) for every  $x$ , the random variables  $f_{\mathbf{M},x,A}$  and  $f_{\mathbf{M},x,B}$  have the same probability mass functions.

Let us now rigorously define the notion “representative”. Since, in a typical pattern recognition problem, we are given, for some positive integer  $m$  (the

number of classes), training subsets  $TR_1, TR_2, \dots, TR_m$  which are supposed to be “representative” of the available sets  $A_1, A_2, \dots, A_m$ , we wish to define, in general, what it means for a sequence of subsets  $\mathbf{C} = (C_1, C_2, \dots, C_m)$  of a feature space  $F$  to be  $\mathbf{M}$ -representative of another sequence of subsets  $\mathbf{D} = (D_1, D_2, \dots, D_m)$ . We start with natural generalizations of concepts given above.

**Definition 3.** *Given a positive integer  $m$ , a sequence  $C = (C_1, C_2, \dots, C_m)$  of subsets of  $F$ , and a sequence  $x = (x_1, x_2, \dots, x_m)$  of reals,  $\mathbf{M}_{x,C}$  denotes the set of those  $M$  in  $\mathbf{M}$  such that for each  $j$ ,  $1 \leq j \leq m$ ,  $\Pr(M|C_j) = x_j$ .*

**Definition 4.** *Given any subset  $\mathbf{M}$  of  $\mathbf{F}$ , any positive integer  $m$ , any sequence  $C = (C_1, C_2, \dots, C_m)$  of subsets of  $F$ , and any sequence  $x = (x_1, x_2, \dots, x_m)$  of reals such that  $\mathbf{M}_{x,C}$  is nonempty, for any  $j$ ,  $1 \leq j \leq m$ ,  $f_{\mathbf{M},x,C}^j$  is the random variable defined on  $C_j$  whose value at any  $q$  (in  $C_j$ ) is given by*

$$f_{\mathbf{M},x,C}^j(q) = \Pr_{\mathbf{M}}(q \in M | M \in \mathbf{M}_{x,C}).$$

The definition of “representative” is now completely natural.

**Definition 5.** *Given any subset  $\mathbf{M}$  of  $\mathbf{F}$ , any positive integer  $m$ , and any two sequences  $\mathbf{D} = (D_1, D_2, \dots, D_m)$  and  $\mathbf{C} = (C_1, C_2, \dots, C_m)$  of subsets of  $F$ , we say that  $\mathbf{C}$  is  $\mathbf{M}$ -representative of  $\mathbf{D}$  if*

- (a) for any  $j$ ,  $1 \leq j \leq m$ ,  $C_j \subseteq D_j$ ,
- (b) for any  $j$ ,  $1 \leq j \leq m$ ,  $C_j \sim_{\mathbf{M}} D_j$ ,
- (c) for any sequence  $\mathbf{x} = (x_1, x_2, \dots, x_m)$  of reals, and for any  $j$ ,  $1 \leq j \leq m$ , the random variables  $f_{\mathbf{M},x,C}^j$  and  $f_{\mathbf{M},x,D}^j$  have the same probability density functions.

*Enrichment and Uniformity* Simply having an  $\mathbf{M}$ -representative set of training examples could not possibly guarantee one’s ability to find a classifier which accurately solved the given problem. For example, if  $\mathbf{M}$  consisted of the single set  $F$ , the feature space itself, then given any two sequences  $\mathbf{D} = (D_1, D_2, \dots, D_m)$  and  $\mathbf{C} = (C_1, C_2, \dots, C_m)$  of subsets of  $F$  such that for any  $j$ ,  $1 \leq j \leq m$ ,  $C_j \subseteq D_j$ ,  $\mathbf{C}$  is  $\mathbf{M}$ -representative of  $\mathbf{D}$ .

There are actually two, natural requirements a collection  $\mathbf{M}$  must satisfy in order to have any chance of building a reasonable classifier from an  $\mathbf{M}$ -representative set of training examples.

The first of these, called uniformity, whose formal definition will be given shortly, basically requires that the members of  $\mathbf{M}$  uniformly cover all regions of the feature space where training examples are present. This is clearly an essential requirement, for otherwise  $\mathbf{M}$ -representative would really amount to “representative in this region of the feature space but not in this other region”. Trivially, since  $\{F\}$  uniformly covers  $F$ , if  $\mathbf{M}$  were equal to  $\{F\}$ ,  $\mathbf{M}$  would be uniform.

The second requirement, called enrichment, would not be satisfied were  $\mathbf{M}$  equal to  $\{F\}$ . Here we basically require that the different (training) classes constituting the given problem not be  $\mathbf{M}$ -indiscernible from one another. Again,

this requirement, whose formal definition will be given shortly, is both reasonable and essential. For if we created a problem by distributing all points in a sufficiently complex feature space among two classes at random, and then specified training sets by random sampling, the training sets would certainly be  $\{F\}$ -representative, and  $\{F\}$  would be uniform, but the classification problem would (and should), by any reasonable standard, be unsolvable.

Motivated by this discussion, we now present the formal definitions:

**Definition 6.** For a given sequence of subsets  $\mathbf{C} = (C_1, C_2, \dots, C_m)$  of  $F$ ,  $\mathbf{M}$  is said to be  $\mathbf{C}$ -uniform if for every  $j$ ,  $1 \leq j \leq m$ , every member  $q$  of  $C_j$ , and every sequence  $\mathbf{x} = (x_1, x_2, \dots, x_m)$  of real numbers such that  $\mathbf{M}_{\mathbf{x}, \mathbf{C}}$  is nonempty,

$$Pr_{\mathbf{M}}(q \in M | M \in \mathbf{M}_{\mathbf{x}, \mathbf{C}}) = x_j.$$

While it may not be apparent that this definition formalizes the intuitive description of uniformity given above, in [6] we prove, mathematically, that it does.

Now for the issue of enrichment.

**Definition 7.** Given a sequence  $\mathbf{C} = (C_1, C_2, \dots, C_m)$ , the  $\mathbf{C}$ -enrichment degree of  $\mathbf{M}$  (written  $e(\mathbf{C}, \mathbf{M})$ ) is defined to be

$$\inf\{|Pr(M|C_i) - Pr(M|C_j)| \mid M \in \mathbf{M}, 1 \leq i \leq m, 1 \leq j \leq m\}.$$

$\mathbf{M}$  is said to be  $\mathbf{C}$ -enriched if  $e(\mathbf{C}, \mathbf{M}) > 0$ .

This definition clearly does formalize the intuitive description of enrichment given above.

*The Solvability Theorem* We are now in a position to give the central definition of this paper. In light of the development above, this definition is completely natural, and seems to constitute the minimal condition appropriate to the concept of solvability.

**Definition 8.** An  $m$ -class problem in supervised learning, presented as two finite sequences  $\mathbf{E} = (E_1, E_2, \dots, E_m)$  and  $\mathbf{T} = (T_1, T_2, \dots, T_m)$  of classes in a finite feature space (intuitively, the examples and the training examples, respectively), is said to be solvable if there exists a collection  $\mathbf{M}$  of subsets of the feature space such that  $\mathbf{T}$  is  $\mathbf{M}$ -representative of  $\mathbf{E}$ , and such that  $\mathbf{M}$  is  $\mathbf{T}$ -enriched and  $\mathbf{T}$ -uniform.

The following theorem says, in essence, that any solvable problem in supervised learning can actually be solved:

**Theorem 1.** There exists an algorithm  $\mathcal{A}$  with the following property: given any solvable problem,  $\mathbf{E}$ ,  $\mathbf{T}$ , in supervised learning, if  $\mathbf{M}$  is a collection of subsets of the feature space such that  $\mathbf{T}$  is  $\mathbf{M}$ -representative of  $\mathbf{E}$ , and if  $\mathbf{M}$  is  $\mathbf{T}$ -enriched and  $\mathbf{T}$ -uniform, then given any desired upper bound  $u$  on error rate,  $\mathcal{A}$  will output, within time proportional to  $1/u$  and inversely proportional to the square of  $e(\mathbf{T}, \mathbf{M})$ , a classifier whose expected error rate on  $\mathbf{E}$  is less than  $u$ .

The algorithm  $\mathcal{A}$  builds classifiers by sampling, with replacement, from the set  $\mathbf{M}$ , and then combining the “weak classifiers” in the resulting sample. We reduce  $n$ -class problems to  $n$ -many 2-class problems; given a training pair  $(T_1, T_2)$  for any such 2-class problem, a sample  $\mathbf{S}$  of size  $t$  produces the classifier which assigns any given example  $q$  to class 1 if

$$\frac{1}{t} \sum_{S \in \mathbf{S}} \frac{\chi_S(q) - Pr(S|T_2)}{Pr(S|T_1) - Pr(S|T_2)} > 0.5$$

(where  $\chi_S$  is the characteristic function of  $S$ ). For a rigorous proof of this theorem, and related results, see [5–7].

Let us also note that the estimate given in the statement of the theorem for run time of the algorithm  $\mathcal{A}$  is intentionally crude, and is provided solely for the purpose of indicating computational feasibility. For more useful statistical estimates, we refer the reader to [2].

## 4 Conclusions

Our intention in this paper was to examine, from a purely mathematical perspective, fundamental issues in the field of supervised learning; and to then explore the usefulness of such a perspective in practical application. The results we derived show a good deal of promise for the general approach, and our hope is that as new algorithmic implementations are developed, results will improve even further.

## References

1. L. Breiman, Bagging Predictors, *Machine Learning*, **24**, 1996, pp. 123-140.
2. D. Chen, Statistical Estimates for Kleinberg’s Method of Stochastic Discrimination, Ph.D. Thesis, SUNY/Buffalo, 1998.
3. Y. Freund, R. E. Schapire, Experiments with a New Boosting Algorithm, *Proceedings of the Thirteenth International Conference on Machine Learning*, Bari, Italy, July 3-6, 1996, pp. 148-156.
4. T. K. Ho, Random Decision Forests, *Proc. of the 3rd Int’l Conference on Document Analysis and Recognition*, Montreal, Canada, 1995, pp. 278-282.
5. E. M. Kleinberg, Stochastic Discrimination, *Annals of Mathematics and Artificial Intelligence*, 1990, pp. 207-239.
6. E. M. Kleinberg, An Overtraining-Resistant Stochastic Modeling Method for Pattern Recognition, *Annals of Statistics*, 1996, pp. 2319-2349.
7. E. M. Kleinberg, On the Algorithmic Implementation of Stochastic Discrimination, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, to appear.
8. E. M. Kleinberg, A Note on the Mathematics Underlying Boosting, preprint, to appear.
9. D. Michie, D. Spiegelhalter, C. C. Taylor, *Machine Learning, Neural and Statistical Classification*, Ellis Horwood, 1994.
10. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, Oct 1993.