Building Projectable Classifiers of Arbitrary Complexity

Tin Kam Ho

Bell Laboratories, Lucent Technologies
700 Mountain Avenue
Murray Hill, NJ 07974, USA
tkh@research.att.com

Eugene M. Kleinberg

Department of Mathematics
State University of New York at Buffalo
Buffalo, NY 14214, USA
kleinbrg@cs.buffalo.edu

Abstract

Conventional methods for classifier design often suffer from having two conflicting goals – to develop arbitrarily complex decision boundaries to suit a given problem, and at the same time to constrain the complexity of those boundaries to avoid overfitting given training data. A recent analysis reveals that the conflict is resolvable by building classifiers based on projectable elements, which are weak discriminators that perform equally well for both training and testing data. Based on this analysis, we present a method that constructs a classifier up to arbitrary complexity while preserving generalization accuracy.

1. Introduction

Classes in typical pattern recognition problems are often defined implicitly by given training data. In most cases the training set is so sparse and limited that the existence of models capable of projecting (generalizing the classification) from these implicit definitions to unseen samples has to be assumed. The degree to which a classifier does, in fact, project satisfactorily, is often determined empirically by the accuracy of the model on a testing set.

The problem of overtraining a classifier usually occurs when the underlying generic nature of such presumably projectable model forms is allowed to be indefinitely specialized. This is often a side-effect of tailoring the decision boundaries to suit the contour of the class boundaries implicit in the training data. It appears as if there is no way one can avoid this problem in order to develop a solution that is sufficiently complex for a given problem. Our study reveals that by exercising some control over the process by which the classifier increases its complexity (number of free parameters), one need not be trapped in this dilemma.

In this paper we describe a method – *stochastic discrimination* – that constructs a classifier to arbitrary complexity while preserving its projectability. The essence of the idea is that the classifier is constructed by combining many simple discriminators that are referred to as *weak models*. Given their simplicity, weak models tend to be very projectable, i.e., their generalization error tend to be low as determined by their limited capacity [9]. While any single weak model will not usually be very accurate as a classifier, the combined classifier's complexity and accuracy rise monotonically as more weak models are included. The key point is that the projectability of the combined classifier is comparable to that of the weak models from which it is built, and so, while complexity and accuracy are increasing, projectability is not decreasing. This phenomenon is predicted in the theory and confirmed in our experiments.

Stochastic discrimination was first proposed in [5] and the mathematical theory is given in detail in [6]. An application to handwritten digits is reported in [7]. Some variants of the method have been studied in [1], [3], and [4]. We will summarize the theory and the method, and then details of an implementation and results of a series of experiments.

2. Stochastic Discrimination

The method of stochastic discrimination constructs a classifier by combining a large number of simple discriminators that are called *weak models*. A weak model is a subset of the feature space, i.e., a set of data points. An example of a weak model is $\{(x, y)|x > 0\}$ in a two-dimensional space. Conceptually, the classifier is constructed by a three-step process: (1) weak model generation, (2) weak model evaluation, and (3) weak model combination. The generator enumerates weak models in an arbitrary order and passes them on to the evaluator. The evaluator has access to the training set. It rates and filters the weak models according to their capability in capturing points of each class and satisfying certain required conditions. The combinator then produces a discriminant function for each class that depends on a point's membership status with respect to each model, and the models' ratings. At classification, a point is assigned to the class whose discriminant function has the highest value.

Informally, the method captures the intuition of gaining wisdom from graded random guesses. And while it is superficially reminiscent of the notion of combining multiple classifiers trained for the same task

(see, for example, [2]), stochastic discrimination differs fundamentally from previous work in that it is based on an analysis of the space of all possible classifiers. Here classifiers are identified with their decision regions, and not with any specific algorithms which might have been employed to construct such regions. In other words, two classifiers are considered equivalent if their decision regions are identical. In this way, algorithmic details in the classification procedure are omitted, and it becomes possible to talk about the entire population of classifiers, or all ways to partition and interpret the feature space.

The theory predicts that if the retained weak models satisfy certain conditions, namely, enrichment and uniformity, the accuracy of the classifier increases monotonically with the number of weak models. The rate of increase is such that very high degree of accuracies can be achieved far before the source of weak models is exhausted. Moreover, the difference in accuracy of the classifier on the training and testing sets is no more than the difference of the ratings of the individual weak models on the two sets. In practice, this difference can be kept to a minimum by using weak models of simple form.

2.1. Enrichment and Uniformity Conditions

It is unlikely that a randomly chosen subset of the feature space will coincide precisely with the set of all points of any one single class. It is also unlikely that it will contain an equal fraction of points of each of the classes. Mostly likely, it will capture, unevenly, fractions of one or more classes. If the fraction of points from a particular class captured by a particular weak model is larger than the fraction of any other class captured by that weak model, then that weak model is said to be *enriched* with respect to that class. Assuming a two-class problem with classes i and j, we can formally define the degree of enrichment of a model m with respect to class i versus class j as

$$d_{ij}(m) = r_i(m) - r_j(m),$$

where

$$r_k(m) = \frac{|m \cap TR_k|}{|TR_k|}$$

for $k = i, j$. TR_i and TR_j are the training sets for classes i and j, respectively, and $|A|$ denotes the cardinality of the set A. In the discussion below, we will informally refer to the quantities $r_i(m)$ and $r_j(m)$ as *ratings* of model m.

At a minimum, we require that $|d_{i,j}(m)|$ be greater than 0, that is, any weak model chosen from our random stream must be *nontrivial* in the sense that it is enriched with respect to either class i or class j. However, a larger difference $|d_{i,j}|$ will speed up the convergence of the discriminant.

The rating r_i measures how well a model is able to capture points of class i, but it tells nothing about which points are being captured. If the weak models simply capture the same points again and again, they contain virtually no information concerning those points which are not captured. Therefore in addition to the enrichment criterion, another condition that is referred to as *uniformity* has to be satisfied by the collection of weak models. Intuitively, we require that the selected models must cover evenly all parts of the feature space where training samples are given. Specifically, we require that the set of selected models should be *unbiased* for any particular point, in other words, the number and quality (given by the enrichment ratings) of models covering each point of a given class should be the same. This uniformity condition guarantees a duality between the probability distribution of the discriminant function in the feature and model spaces, and is critical in the proof of convergence of the classifier's accuracy [6]. Given the need for projectability of the models, strict uniformity is often difficult to obtain. However, empirical evidence shows that satisfactory results can be obtained with approximate uniformity, a fact which is also supported by theoretical study [1].

2.2. Combining the Weak Models

We now show how to combine a set of weak models to yield a discriminant function. Let q be a point in the feature space F, and let m be a weak model from a given collection M. Given any two classes i and j, a random variable $X_{ij}(q, m)$ is defined as follows:

$$X_{ij}(q,m) = \begin{cases} 2(\frac{C_m(q) - r_j(m)}{r_i(m) - r_j(m)}) - 1 & \text{if } r_i(m) \neq r_j(m), \\ 0 & \text{if } r_i(m) = r_j(m). \end{cases}$$

where $C_m(q)$ is the characteristic function of m evaluated at q ($C_m(q)$ is 1 if $q \in m$, and 0 if $q \notin m$). The variable $X_{ij}(q, m)$ has a value for each point q and each model m. It can be shown [6] that, as long as enrichment and uniformity conditions are satisfied, for each point q, the expectation of X_{ij} is 1 if q is a member of class i, and is -1 if q is a member of class j; furthermore, if for each point q we let

$$Y_{ij}(q) = \frac{1}{p_{ij}} \sum_{k=1,\ldots,p_{ij}} X_{ij}(q, m_k),$$

where p_{ij} is the number of models accumulated for the class pair i and j, then the expectation of Y_{ij} is 1 if q is a member of class i, is -1 if q is a member of class j, and the variance of Y_{ij} approaches 0 as the number of models p_{ij} increases.

An n-class ($n > 2$) problem is solved by decomposing the problem into $n(n-1)$ 2-class discrimination problems. For each point q, and each class i, let

$$Z_i(q) = \sum_{j=1,\ldots,n, j \neq i} Y_{ij}(q).$$

$Z_i(q)$ can then be used as a discriminant function for class i. A point q is assigned to that class i for which $Z_i(q)$ is the maximum.

The complexity of the resultant classifier is determined by the number of weak models used. Given the ability to improve by accumulating additional weak models, the classifier's complexity can be enhanced to suit a given problem.

3. Algorithmic Choices

The method outlined above allows for rich possibilities of variation at the algorithmic level. The variations may be in the design of the weak model generator, or in ways to enforce the enrichment and uniformity conditions. It is also possible to change the definition of the discriminant, or to use different kinds of ratings. The effects of these variations are often difficult to analyze. The rest of this paper is devoted to a discussion of algorithmic implementations, and to results obtained from the application of such implementations to concrete pattern recognition problems.

3.1. Types of Weak Models

In a finite space of l dimensions each having a distinct levels, there are a^l-many points in the space, and 2^{a^l}-many subsets of the space. In an infinite space, there are uncountably many possible sets of points. Each of these sets is a candidate to be a weak model. An exhaustive enumeration is usually infeasible, so random samples from such a population are used instead. Two guidelines should be observed in drawing the samples:

(1) *projectability*: A weak model should contain a sufficient number of points so that the solution can be projectable. Geometrically, this means that a useful model must be of certain minimum size, and it should be able to capture points that are considered *neighbors* of one another. To guarantee similar accuracies of the classifier (based on similar ratings of the weak models) on both training and testing data, one also needs an assumption that the training data are *representative*. Data representativeness and model projectability are two sides of the same question. A weak model defines a *neighborhood* in the space, and we need a training sample in a neighborhood of every unseen sample. Otherwise, since our only knowledge of the class boundaries is from the given training set, there can be no basis for any inference concerning regions of the feature space where no training samples are given.

(2) *simplicity of representation*: A weak model should have a simple representation. That means, the characteristic function of the model must be cheaply computable. For this reason, a model represented as a listing of all the points it contains is practically useless since the resultant solution could be as expensive as an exhaustive template matching. An example model with a simple representation is a half-plane bounded by a straight line in a two-dimensional feature space.

Conditions (1) and (2) restrict the type of weak models but by no means reduce the number of candidates to any tangible limit. To obtain an unbiased collection of the candidates with minimum effort, random sampling with replacement is useful. The training of the method thus relies on a stochastic process which, at each iteration, generates a weak model that satisfies the above conditions.

A convenient way to generate weak models randomly is to use a type of model that can be described by a small number of parameters. The values of the parameters can be chosen pseudo-randomly. Some example types of models that can be generated this way include (1) half-spaces bounded by a threshold on a randomly selected feature dimension; (2) half-spaces bounded by a hyperplane equi-distant from two randomly selected points; (3) regions bounded by two parallel hyperplanes perpendicular to a randomly selected axis. (4) hypercubes centered at randomly selected points with edges of varying lengths; (5) balls (based on a chosen metric) centered at randomly selected points with randomly selected radii. A model can also be a union or intersection of several regions of these types.

In creating weak models certain heuristics may be used to specify the way parameters are chosen. For instance, random points may be chosen among those whose coordinates fall inside the range of those of the training samples. Or, radii of the balls could be restricted to a fraction of the range of values in a particular feature dimension. These heuristics are intended to speed up the search for acceptable models by confining the search to the most interesting regions, or by guaranteeing some minimum model size.

3.2. Enrichment Enforcement

The enrichment condition is relatively easy to enforce, as models biased towards one class are most common. But since the strength of the biases ($|d_{ij}(m)|$) determines the rate at which accuracy increases, we tend to prefer to use models with an enrichment degree further away from zero.

One way to implement this is to use a threshold on the enrichment degree to select weak models from the random stream so that they are of some minimum quality. In this way, one will be able to use a smaller collection of models to yield a classifier of the same level of accuracy. However, there are tradeoffs involved in doing this. For one thing, models of higher rating are less likely to appear in the stream, and so more random models have to be explored in order to find sufficient numbers of higher quality weak models. And once the type of model is fixed and the value of the threshold is set, there is a risk that such models may never be found.

Alternatively, one can use the most enriched model found in a pre-determined number of trials. This also makes the time needed for training more predictable, and it permits a tradeoff between training time and model quality.

In enriching the model stream by either the thresholding or maximizing method, it is important to remember that if the quality of weak models selected is allowed to get too high, there is a risk that they will become training set specific, that is, less likely to be projectable to unseen samples. This could present a problem since the projectability of the final classifier is directly based on the projectability of its component weak models. However, it is very improbable that any weak model generated by the method outlined above, would be of such a high enrichment degree that training set specificity becomes an issue.

Since each multi-class problem is decomposed into a set of two-class problems, a model that is good for a particular two-class problem may also be useful for

others. Also, when a model is enriched for problem i versus j by having $d_{ij}(m) > 0$, its complement is also enriched for problem j versus i. Therefore one can use both the model and its complement on the subproblems (i, j) and (j, i) respectively.

3.3. Uniformity Promotion

The uniformity condition is much more difficult to satisfy. Strict uniformity requires that every point be covered by the same number of weak models of every combination of per-class ratings. This is rather infeasible for continuous and unconstrained ratings.

One useful strategy is to use only weak models of a particular rating. Since $r_i(m)$ and $r_j(m)$ will be the same for all m enriched for classes i and j, we need only to make sure that the each point is in the same number of models. This can be enforced by creating models in groups such that each group partitions the space into non-overlapping regions. An example is to use leaves of a fully-split decision tree, where each leave is perfectly enriched for one class, and each point is covered by exactly one leave of each tree. For a pairwise discrimination between classes i and j, we can use only those leaves that contain only class i, such that $r_i(m)$ is always 1 and $r_j(m)$ is always 0. Constraints are put in the tree-construction process to guarantee some minimum projectability. Multiple trees are created and combined, raising the classification accuracy [4].

With other types of models, a first step to promote uniformity is to use models that are unions of small regions with simple boundaries. The component regions may be scattered throughout the space. This way, models have simple representations but can describe complicated class boundaries. Moreover, the scattered locations of component regions tend to cover large areas more evenly.

A more sophisticated way to promote uniformity involves defining a measure of the lack of uniformity and an algorithm to minimize such a measure. The goal is to create or retain more models located in areas where the coverage is thinner. One such measure we used is the count of points that are covered by less-than-average number of previously retained models. For each point x in the class c_0 to be positively enriched, we calculate, out of all previous models used for that class, how many of them have covered x. If the coverage is less than the average for class c_0, we call x a weak point. Each model is checked against how many weak points it covers. We take the model that covers the most weak points among a pre-set number of trials. One can go further to introduce a bias to the model generator so that models covering the weak points are more likely to be created.

3.4. Alternative Discriminants

The general framework of stochastic discrimination works with other discriminants as well. A variant has been proposed in [3] and studied in detail in [1], there we define the ratings differently by

$$r'_i(m) = \frac{|m \cap TR_i|}{|m \cap TR|},$$

for all i. The discriminant for class i is

$$W_i(q) = \frac{\sum_{k=1,\ldots,p_i} C_m(q) r'_i(m)}{\sum_{k=1,\ldots,p_i} C_m(q)}.$$

where p_i is the number of models used for class i. This discriminant is more useful for problems involving many classes. It turns out that, with this discriminant, the classifier also approaches perfection asymptotically provided that an additional *symmetry* condition is satisfied. The symmetry condition requires that the same number of models is used for all permutations of $(r'_1, r'_2, ..., r'_n)$. It prevents biases created by using more (i, j)-enriched models than (j, i)-enriched models for all pairs (i, j) [1]. Again, this condition may be enforced by using only certain particular permutations of the r' ratings [4].

4. Experimental Results

We carried out a series of experiments to determine the effects of using different weak model types, and various enrichment and uniformity enforcement algorithms. In all these experiments the implementation of the method remained the same. Our intention was to explore the generality of the algorithm and its easy adaptability to drastically different domains.

4.1. Approaching the Bayes error rate

Our first results show that on a well-posed problem, a classifier built by stochastic discrimination (*sd*) is able to get close to the Bayes error rate. Here samples are drawn from two overlapping two-dimensional Gaussian joint distributions with centers at (80,80) for class 0, and (120,120) for class 1, respectively. In each distribution both x and y vary independently with a standard deviation of 20. The Bayes error rate calculated by numerical integration is 0.07866, which implies a maximum correct rate of 92.13%. Both the training and testing sets contain 1000 pseudo-random samples of each class.

Each weak model is the union of 20 rectangles at randomly generated locations and of variable sizes. We retain only those models that have an enrichment degree over 0.1359, which is set by taking the average of the first 100 tried models. No other uniformity promotion algorithm is used. Bayes correct rate is reached with less than 1000 models. Accuracies on both training and testing sets remain similar at all times (Figure 1).

4.2. Four classes that are not linearly separable

With this example, we intend to show that sd is able to increase the complexity of the classifier to suit the class boundaries. Here we artificially created points of four classes in a two-dimensional, bounded space. The classes have irregular spreads over the space including disconnected regions and are not linearly separable. This problem is difficult for classifiers of limited complexity.

We randomly take 5% of the points for training and 50% for testing, making certain training and test sets

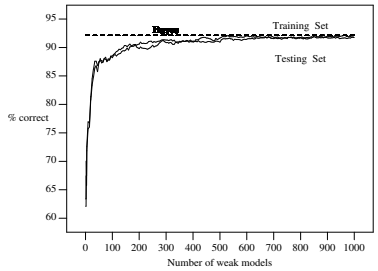
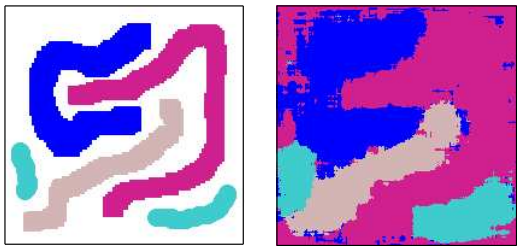


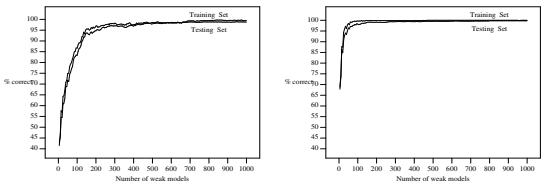
Figure 1. Accuracies on the overlapping Gaussian data.

are disjoint. Models are unions of 20 rectangles just as before. Figure 2 (a) shows the true distributions and (b) shows the classification using 500 weak models. Accuracy increases faster when uniformity promotion is in place, but both classifiers remain highly projectable (Figure 3).



(a) (b)

Figure 2. (a) True distributions and (b) classification with 500 weak models.



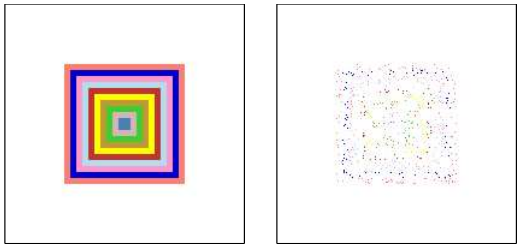
(a) (b)

Figure 3. Accuracy (a) without and (b) with uniformity promotion.

4.3. When the training set is sparse

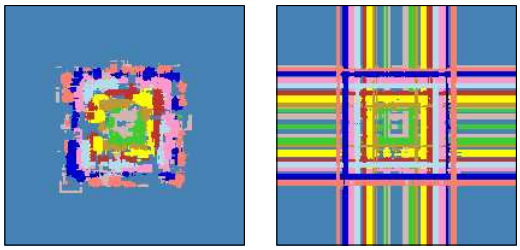
Here we will show the important effect of model types when there is not sufficient training data. Again, the data was created in a two-dimensional, bounded space. The true distributions can be described by ten squares nested within one another, and each class is distributed along a border of width 5 (Figure 4 (a)). Again, we take only 5% of all the points for training (Figure 4 (b)) and 50% for testing. This means that some classes will be represented by very few training samples.

We constructed two classifiers by using two different types of weak models. Models of the first type are same as before: unions of 20 randomly placed rectangles; and those of the second type are unions of 10 axis-parallel strips with variable widths. Figures 5 (a) and (b) show the classification of the space with 100 models of each type.



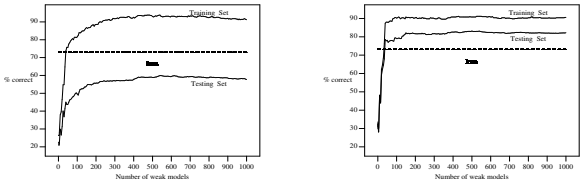
(a) (b)

Figure 4. Distributions of (a) all and (b) the training samples in the nested squares example.



(a) (b)

Figure 5. Classification of space with 100 of (a) type 1 and (b) type 2 models.



(a) (b)

Figure 6. Accuracies with (a) type 1 and (b) type 2 models.

It is easy to see that models of the second type project much better to areas that are underrepresented by the training set. As can be expected, not only does the classifier constructed with the strips perform better than that with the rectangles, but the difference between the training and testing set accuracies is considerably smaller (Figures 5). We also measured the accuracy of a nearest-neighbor classifier using Euclidean distance for comparison. The nearest-neighbor classifier is 73.10% correct on the testing set. With 1000 models of the second type, our classifier is considerably better (82.26%).

It can be argued that the design of the second type

of weak model just happens to fit the true distributions well. This is true, but it must be emphasized that this difference is largely due to the sparseness of the training set. This example shows clearly that the representativeness of a set of data is relative to the assumed projection of the chosen type of model. To the second type of model, the training set is far more representative than to the first type.

While this is an artificial example, we believe that it reveals important characteristics of most practical recognition problems. This is when domain knowledge should help: good heuristics about the problem can lead to better choice of models. With very sparse training sets, there is no replacement for feature transformations that can gather points of the same class into neighborhoods of simple representations. However, it is not necessary that very well-designed models are needed to achieve any satisfactory results, as would be shown in the next example.

4.4. Identifying boundaries in DNA sequences

In this experiment, we use data from samples of DNA sequences that were compiled and made publicly accessible by the Project StatLog [8]. The problem is to identify two kinds of boundaries in DNA sequences. The boundaries specify where portions of a sequence are removed in the process of protein creation in higher organisms. The 'exon to intron' and 'intron to exon' boundaries are coded as two classes respectively, and a 'neither' response is coded as the third class. Feature vectors are derived from contiguous blocks of 60 elements (nucleotides) from the sequences. 2000 samples are used for training and 1186 for testing.

Each weak model is the union of 5 halfspaces, each of which is chosen by randomly selecting a feature dimension and thresholding the value. Given that the feature vector has 180 dimensions, there are a large number of possible choices. However, very good results are achieved with fewer than 500 models. In fact, with fewer than 500 models the classifier does better on the test set than both the k-nearest-neighbor classifier and a feedforward neural-network trained with back-error-propagation. Figure 7 shows the accuracy of the classifier and the results of 'knn' and 'backprop' as reported in the literature [8].

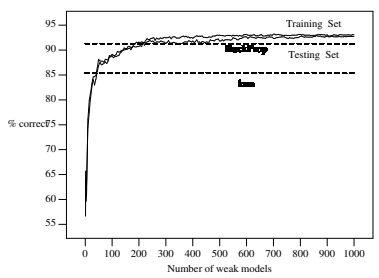


Figure 7. Accuracies on the DNA data.

5. Conclusions

Stochastic discrimination offers a way to construct classifiers which approach perfection asymptotically with both domain-specific or generally applicable heuristics. The heuristics are embedded in the forms of the weak models, and isolated from the process of classifier construction. More sophisticated models are used to compensate for the lack of sufficient and representative training data. With dense training samples, simple models are expected to suffice. Models can also be grouped as partitions of the space, which is an important special case as it includes essentially all conventional classifiers.

Projectability is maintained by the form of the weak models one chooses to use. With that, the combined classifier can be extended to arbitrary complexity without loss of generalizability. It is a distinctive advantage of the method. Practically, it permits continuous tradeoff between space-time complexity and recognition accuracy. Since the weak models can be made independently and in parallel, the method provides a parallel learning algorithm that does not have the problem of falling into local optima. Practically, the parallelism can also be exploited for efficient classifier construction.

Acknowledgements

We are grateful to Henry Baird, Dar-Shyang Lee, and George Nagy for many helpful discussions.

References

[1] R. Berlind, *An Alternative Method of Stochastic Discrimination with Applications to Pattern Recognition*, Doctoral Dissertation, Department of Mathematics, SUNY at Buffalo, 1994.

[2] T.K. Ho, *A Theory of Multiple Classifier Systems and Its Application to Visual Word Recognition*, Ph.D. Thesis, Dept. of Computer Science, State University of New York at Buffalo, 1992.

[3] T.K. Ho, Recognition of Handwritten Digits by Combining Independent Learning Vector Quantizations, *Proceedings of the Second International Conference on Document Analysis and Recognition*, Tsukuba Science City, Japan, October 20-22, 1993, 818-821.

[4] T.K. Ho, Random Decision Forests, *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, Montreal, Canada, August 14-18, 1995, 278-282.

[5] E.M. Kleinberg, Stochastic Discrimination, *Annals of Mathematics and Artificial Intelligence*, **1**, 1990, 207-239.

[6] E.M. Kleinberg, An overtraining-resistant stochastic modeling method for pattern recognition, Annals of Statistics, to appear.

[7] E.M. Kleinberg, T.K. Ho, Pattern Recognition by Stochastic Modeling, *Proceedings of the Third International Workshop on Frontiers in Handwriting Recognition*, Buffalo, May 1993, 175-183.

[8] Project StatLog, LIACC, University of Porto, *internet address:* ftp.ncc.up.pt (*directory:* pub/statlog/datasets).

[9] V. Vapnik, *Estimation of Dependences Based on Empirical Data*, Springer-Verlag, 1982.