Poor Man vote with M-ary classifiers. Application to Iris recognition.

Vincent Vigneron, Hichem Maaref, Sylvie Lelandais

"Poor man" vote with M-ary classifiers

V. Vigneron[1,2], H. Maaref[2], and S. Lelandais[2]

[1] LIS,
avenue Félix Viallet,
38031 Grenoble cedex, France
vvigne@lis.inpg.fr
[2] LSC FRE 2494
40 rue du Pelvoux
91020 Evry Courcouronnes
vvigne,maaref@cemif.univ-evry.fr

Abstract. Achieving good performance in biometrics requires matching the capacity of the classifier or a set of classifiers to the size of the available training set. A classifier with too many adjustable parameters (large capacity) is likely to learn the training set without difficulty but be unable to generalize properly to new patterns. If the capacity is too small, the training set might not be learned without appreciable error. There is thus advantage to control the capacity through a variety of methods involving not only the structure of the classifiers themselves, but also the property of the input space. Ths paper proposes an original non parametric method to combine optimaly multiple classifier responses. Highly favorable results have been obtained using the above method.

1 Ensemble of classifiers

One recent trend in computational learning looks at what Valiant called "theory of learnable" [24] . Suppose we have a set of n samples and use these to fit a finite number M of classifiers from a family $\mathcal{F}$ of possible classifiers. Then the probability that a classifier g chosen is *consistent* with a training set yet having overall error rate at least E_n, is at most $M(1-E_n)^n$. Using a single classifier has shown a certain limitation in achieving satisfactory recognition performance and this leads us to use multiple classifiers, which is now a common practice [15]. Readers can find surveys in Ripley [21] and in Devroye *et al.* [5]. Some recent papers include those proposing directional: mixtures of experts [12], boosting methods [6], bagging methods [2], *query by committee* [8], stacked regression [25], distributed estimation for data fusion [1, 9, 23]. These papers prove that the approach of multiple classifiers produced a promising improvement in recognition performance. The efficacy of the method is explained by the following argument: for instance, most classifiers share the feature that the solution space is highly degenerate. The post-training distribution of classifiers trained on different training sets chosen according to the density of the samples $p(x)$ will be spread out over a multitude of nearly equivalent solutions. The ensemble is a particular sample from the set of these solutions. The basic idea of the ensemble approach is to eliminate some of the generalization errors using the differentiation within the realized solutions of the learning problem. The variability of the errors made by the classifiers of the ensemble has shown that the *consensus* improves significantly on the performance of the best individual in the ensemble**. In [11], Hansen *et al.* used a digit recogntion problem to illustrate how

** This analysis is certainly true for situations described here wherein the classifiers of the ensemble see different training patterns, it can be effective even when all the classifiers are trained using the same training set [10].

the ensemble consensus outperformed the best individuals by 25%. The marginal benefit obtained by increasing the ensemble size is usualy low due to correlation among errors made by participating classifiers on an input x [10]: most classifiers will get the right answer on *easy* inputs while many classifiers will make mistakes on "difficult" inputs.
The number of classifiers can be very high (some hundreds) so it is difficult to understand their decision characteristics. Some of the above referenced papers used a simple scheme of combination, which just *cascade* multiple classifiers. This scheme results in a less larger robustness of the agregated classifier. This is due to classifiers interdependencies that may reduce the recognition performance.
Our present interest is a new scheme to improve class separation performances in combining multiple classifiers using information theoric learning.
The paper is organizes as follows. Section 2 explains the basic idea of wavelet denoising. Section 3 presents a theoretical sketch of bins-based classification (BBC). Section 4 provides an independence measure based on mutual information to evaluate the classifier combination. Section 5 presents an experiment where few samples are available in the case of iris recognition and further investigations are finally discussed.

2 Relations between independence and M-ary classifiers collective decision

One can often read in literature the following sentence "*to combined a set of classifiers, it is better to choose independent classifiers*", *e.g.* in Nadal *et al.* [17]. This idea is vague and this is illustrated on the following example.

Example 1. Consider a binary classification problem (with equiprobable classes w_1 and w_2) with two classifiers $\mathcal{C}_1$ and $\mathcal{C}_2$ whose outputs are ς_1 and ς_2. Suppose there is no reject mechanism and that their performances are homogeneous for the two classes, *i.e.* their probabilities of correct classification λ_1 and λ_2 are equal: $p(\varsigma_1 = w_1|w_1) = p(\varsigma_1 = w_2|w_2) = \lambda_1$ and $p(\varsigma_2 = w_1|w_1) = p(\varsigma_2 = w_2|w_2) = \lambda_2$. By evidence, $p(\varsigma_1 = w_1|w_2) = p(\varsigma_1 = w_2|w_1) = 1 - \lambda_1$ and $p(\varsigma_2 = w_1|w_2) = p(\varsigma_2 = w_2|w_1) = 1 - \lambda_2$. The probabilities of the outputs ς_1 and ς_2 are

$$p(\varsigma_1 = w_1) = p(\varsigma_1 = w_1|w_1)p(w_1) + p(\varsigma_1 = w_1|w_2)p(w_2) = \lambda_1\tfrac{1}{2} + (1-\lambda_1)\tfrac{1}{2} = \tfrac{1}{2}$$
$$p(\varsigma_2 = w_1) = p(\varsigma_2 = w_1|w_1)p(w_1) + p(\varsigma_2 = w_1|w_2)p(w_2) = \lambda_2\tfrac{1}{2} + (1-\lambda_2)\tfrac{1}{2} = \tfrac{1}{2}$$

Suppose the two classifiers are independent, then we have $p(\varsigma_1 = w_1, \varsigma_2 = w_1) = p(\varsigma_1 = w_1)p(\varsigma_2 = w_1) = \frac{1}{2}\frac{1}{2} = \frac{1}{4}$. Similarly, $p(\varsigma_1 = w_1, \varsigma_2 = w_2) = p(\varsigma_1 = w_2, \varsigma_2 = w_1) = p(\varsigma_1 = w_1)p(\varsigma_2 = w_2) = \frac{1}{2}\frac{1}{2} = \frac{1}{4}$. Thus the performance of the ensemble is independent of the performance of the individuals. This can be possible only if $\lambda_1 = \lambda_2 = \frac{1}{2}$, *i.e.* two classifiers are independent if their are random (recognition rate at 50%) ! □

This example suggest that interesting classifiers should not be independent in the classical sense. Still one class of probabilities consists of conditional probabilities [20], *e.g.* we have to impose that classifiers be independent for each class. Conditional densities of two random variables ς_1 and ς_2 conditioned by the class arise in the following case :

$$p(\varsigma_1 = w_j, \varsigma_2 = w_\ell|w_i) = p(\varsigma_1 = w_j|w_i)p(\varsigma_2 = w_\ell|w_i), \forall 1 \leq i, j, \ell \leq K. \quad (1)$$

Suppose that prior probabilities $p(w_i) = 1/K, \forall i$ and that both classifiers admit a conditional probability and satisfy Eq. (1). Then,

$$p(\varsigma_1 = w_j, \varsigma_2 = w_\ell) = p(\varsigma_1 = w_\ell)p(\varsigma_2 = w_\ell), \forall 1 \leq j, \ell \leq K. \quad (2)$$

where $p(\varsigma_1 = w_j) = \sum_{i=1}^{K} p(\varsigma_1 = w_j|w_i)p(w_i) = \frac{1}{K}\sum_{i=1}^{K} p(\varsigma_1 = w_j|w_i)$ and $p(\varsigma_2 = w_\ell) = \sum_{i=1}^{K} p(\varsigma_1 = w_\ell|w_i)p(w_i) = \frac{1}{K}\sum_{i=1}^{K} p(\varsigma_2 = w_\ell|w_i)$. Hence, $p(\varsigma_1 = w_j, \varsigma_2 = w_\ell) = \sum_{i=1}^{K} p(\varsigma_1 = w_j, \varsigma_2 = w_\ell|w_i)p(w_i) = \frac{1}{K}\sum_{i=1}^{K} p(\varsigma_1 = w_j, \varsigma_2 = w_\ell|w_i)$. Eq. (2) becomes:

$$\frac{1}{K}\sum_{i=1}^{K} p(\varsigma_1 = w_j, \varsigma_2 = w_\ell|w_i) = \frac{1}{K}\sum_{i=1}^{K} p(\varsigma_1 = w_j|w_i)\frac{1}{K}\sum_{i'=1}^{K} p(\varsigma_2 = w_\ell|w_{i'}), \quad (3)$$

$$= \frac{1}{K^2}\sum_{i=1}^{K}\sum_{i'=1}^{K} p(\varsigma_1 = w_j|w_i)p(\varsigma_2 = w_\ell|w_{i'}), \quad (4)$$

Replacing (1) in (4) give $\frac{1}{K}\sum_{i=1}^{K} p(\varsigma_1 = w_j, \varsigma_2 = w_\ell|w_i) = \frac{1}{K^2}\sum_{i=1}^{K} p(\varsigma_1 = w_j|w_i)p(\varsigma_2 = w_\ell|w_i) + \frac{1}{K^2}\sum_{i=1}^{K}\sum_{i'\neq i}^{K} p(\varsigma_1 = w_j|w_i)p(\varsigma_2 = w_\ell|w_{i'})$. At the end :

$$(K-1)\sum_{i=1}^{K} p(\varsigma_1 = w_j, \varsigma_2 = w_\ell|w_i) = \sum_{i=1}^{K}\sum_{i'\neq i}^{K} p(\varsigma_1 = w_j|w_i)p(\varsigma_2 = w_\ell|w_{i'}) \quad (5)$$

Eq. (5) can be satisfied only if at least one of both conditions is true: $p(\varsigma_1 = w_j|w_i) = 1/K, \forall 1 \leq i, j \leq K$ or $p(\varsigma_1 = w_\ell|w_i) = 1/K, \forall 1 \leq i, \ell \leq K$.

Example 2. Consider once again our toy example restricted to binary classification and decide that $\lambda_1 = \lambda_2 \approx 1$. Similarly, $p(\varsigma_1 = w_2, \varsigma_2 = w_1|w_1) \approx 1$ and $p(\varsigma_1 = w_1|w_1)p(\varsigma_2 = w_1|w_1) \approx \lambda_1\lambda_2 \approx 1$. We conclude that the two classifiers performed well even they are conditionaly independent.

Further, conditional independence will be the evaluation criterion of the classifiers.

3 Bins-based classification

Suppose that the classifiers of the ensemble are each trained on *independently chosen* training sets of s samples selected according to $p(x)$. We have K classes, each represented as $w_i, 1 \leq i \leq K$. Let $p(w_i|x)$ be the probability that x comes from w_i. The classifier has K possible categories to choose. It is well known that the Bayes classifier represents the optimum measure of performance in the sense of the minimal classification error [26]. The Bayes classifier selects the class $w^*(x) = w_j$ if $p(w_j|x) = \arg\max_{w_i \in \Omega} p(w_i|x)$, where Ω is a partition of the feature space. Using Bayes' formula, this *posterior* probability is $p(w_j|x) = \frac{p(x|w_j)p(w_j)}{\sum_{i=1}^{K} p(x|w_i)p(w_i)}$, where $p(x|w_i)$ (the *likelihood function* of class w_i) and $p(w_i)$ are usually unknown but can be evaluated using the learning data set.
Suppose there are a number of classifiers $\mathcal{C}_k, 1 \leq k \leq M$, each of which produces the output $\varsigma_k(x)$ (simplified for readibility in ς_k). Then the bayes formula gives

$$p(w_j|\varsigma_1, \ldots, \varsigma_M) = \frac{p(\varsigma_1, \ldots, \varsigma_M|w_j)p(w_j)}{\sum_{i=1}^{K} p(\varsigma_1, \ldots, \varsigma_M|w_i)p(w_i)}. \quad (6)$$

Estimating the unknown likelihood $p(\varsigma_1, \ldots, \varsigma_M|w_j)$ is of primordial importance. They could be estimated using some parametric model, but we have no prior information on the collective functionning of the classifiers. This motivates us to search for distribution-free performance of error estimation using a *histogram-based* rule with a fixed partition, the number of bins in the partition being "not too large". Such learning algorithm is given in Table 1. The interesting point sets in the fact that no prior knowledge is needed by the classifiers.
The output space of the classifiers is a discrete space with $K + 1$ distinct points

Learning Algorithm
Inputs: $\{x_1, \ldots, x_n\}$;
Init. Partition output space of $\mathcal{C}_k$ in q_k bins $\mathcal{L}_1^k, \ldots, \mathcal{L}_M^k$ $Q = \prod_{k=1}^K q_k$ sets of exactly K counters associated to the K classes, denoted[a] $n_{i_1,\ldots,i_M}^{w_i}, 1 \leq i \leq K$.
Repeat
1 Compute the outputs $\varsigma_1(x), \ldots, \varsigma_M(x)$ of the classifiers
2 For each $\mathcal{C}_k$, find i_k such that $\varsigma_k(x) \in \mathcal{L}_{i_k}^k, 1 \leq i_k \leq q_k$.
3 Increment the counter of those indices $n_{i_1,\ldots,i_M}^{w_i=w_{\text{true}}}$ matching the true class w_{true} of x
4 For each possible combination of indices $i_1, \ldots, i_K$, $1 \leq i_k \leq q_k$, collect the agregated classification response $y_{i_1,\ldots,i_K}(x)$ as $y_{i_1,\ldots,i_K}(x) = \begin{cases} w_j & \text{if } n_{i_1,\ldots,i_M}^{w_j} = \arg\max_{1 \leq \ell \leq K} n_{i_1,\ldots,i_M}^{w_\ell} > 0 \\ w_0 & \text{in the other case.} \end{cases}$
For each $x \in \{x_1, \ldots, x_n\}$.
output: $\{y_{i_1,\ldots,i_K}(x_1), \ldots, y_{i_1,\ldots,i_K}(x_n)\}$;

[a] $i_1, \ldots, i_M$ are the indices of the set of counters.

Table 1. "Poor man" BBC learning algorithm.

corresponding to the K possible classes, the supplementary one being the rejection class***. The most natural division for such a space consists to consider each point as a division, *i.e.*

$$\mathcal{L}_i = \{w_i\}, 1 \leq i \leq K, \text{ et } \mathcal{L}_0 = \{w_0\}(\text{rejection class}). \tag{7}$$

The classification algorithm is given in Tab.2. If the objective is to provide subjective probabilities and not to classify, the step 3 changes. Then, subjective probabilities can be computed from $n_{i_1,\ldots,i_M}^{w_j}$ for each class by:

$$P(w_i) = \begin{cases} \frac{n_{i_1,\ldots,i_M}^{w_j}}{\sum_{j=1}^M n_{i_1,\ldots,i_M}^{w_j}} & \text{if } \exists n_{i_1,\ldots,i_M}^{w_j} \neq 0, \\ 0 & \text{in the other case.} \end{cases} \tag{8}$$

In the case of a *rank-classifier*, the output is a $K-$dimensional vector $\varsigma_k = (r_1^k, \ldots, r_K^k)$, where r_j^k is the output of the classifier k corresponding to the class w_j. Each axe r_j^k is subdivided in q_{kj} intervals $\mathcal{L}_\ell^{kj}$ which can be of different size, *e.g.* $\mathcal{L}_\ell^{kj}, 1 \leq \ell \leq q_{kj}$ is an interval defined by its lower and upper bounds by $[L_{\ell-1}^{kj}, L_\ell^{kj}]$. The index i_k in the algorithm is computed by

$$i_k = \sum_{j=1}^K \left(i_{kj} \prod_{\ell=1}^{j-1} q_{k\ell} \right), \tag{9}$$

where the indices i_{kj} are such that $r_j^k \in \mathcal{L}_{i_{kj}}^{kj}$.

4 Independence measure of the classifiers ensemble

In section 2, we prove that conditional independence of two classifiers $\mathcal{C}_1$ and $\mathcal{C}_2$ requires that $p(\varsigma_1, \varsigma_2|w_i) = p(\varsigma_1|w_i)p(\varsigma_2|w_i)$. This is a desired property of the ensemble

*** If the objective is just to give subjective probabilities and not to classify, we just need to collect the counter values $n_{i_1,\ldots,i_M}^{w_{\text{true}}}$.

Classification Algorithm
1 Compute the outputs $\varsigma_1(x), \ldots, \varsigma_M(x)$ for a given form x
2 For each $\varsigma_k(x)$, find i_k such that $\varsigma_k(x) \in \mathcal{L}_{i_k}^k$.
3 Compute $y_{i_1,\ldots,i_K}(x)$ as the agregated classification response.

Table 2. Application of the BBC algorithm for a given input x.

performances. The Kullback-Leibler (KL) *divergence* can be considered as a kind of distance[†] between two probability densities, because it is always non negative and it is equal to zero iff the two distributions are equal[‡] (see Hyvrinen *et al.* [19] for more details). This is defined between two probability density functions (pdf's) g and f as

$$D(f\|g) = \sum_x f(x) \log \frac{f(x)}{g(x)}. \tag{10}$$

To apply the Kullback-Leibler divergence here, one might measure the independence between the joint density $p(\varsigma_1, \varsigma_2|w_i)$ and the factorized density $p(\varsigma_1|w_i)p(\varsigma_2|w_i)$, *i.e.*

$$D(p(\varsigma_1, \varsigma_2|w_i)\|p(\varsigma_1|w_i)p(\varsigma_2|w_i)) = \sum_{\varsigma_1,\varsigma_2} p(\varsigma_1, \varsigma_2|w_i) \log \frac{p(\varsigma_1, \varsigma_2|w_i)}{p(\varsigma_1|w_i)p(\varsigma_2|w_i)}. \tag{11}$$

The more $D(p(\varsigma_1, \varsigma_2|w_i)\|p(\varsigma_1|w_i)p(\varsigma_2|w_i))$ is small, the more the classifiers are independent given the class w_i. For a finite number of class $\{w_0, w_1, \ldots, w_K\}$:

$$D(x) = \sum_{w_i} p(w_i) D(p(\varsigma_1, \varsigma_2|w_i)\|p(\varsigma_1|w_i)p(\varsigma_2|w_i)) \tag{12}$$

$$= \sum_{w_i} p(w_i) \sum_{\varsigma_1,\varsigma_2} p(\varsigma_1, \varsigma_2|w_i) \log \frac{p(\varsigma_1, \varsigma_2|w_i)}{p(\varsigma_1|w_i)p(\varsigma_2|w_i)} \tag{13}$$

$$= \sum_{w_i} \sum_{\varsigma_1,\varsigma_2} p(\varsigma_1, \varsigma_2, w_i) \log \frac{p(\varsigma_1, \varsigma_2|w_i)}{p(\varsigma_1|w_i)p(\varsigma_2|w_i)}, \tag{14}$$

Rearranging (14) we find for a given x:

$$D(x) = -\sum_{w_i} p(\varsigma_1, \varsigma_2, w_i) \log \prod_{j=1}^{2} p(\varsigma_j|w_i) + \sum_{w_i} p(\varsigma_1, \varsigma_2, w_i) \log p(\varsigma_1, \varsigma_2|w_i) \tag{15}$$

$$= \sum_{j=1}^{2} \sum_{w_i} H_j - H(\varsigma_1, \varsigma_2; w_0, \ldots, w_K), \tag{16}$$

where $H_j = -\sum_{w_i} \sum_j \log p(\varsigma_j|w_i)p(\varsigma_1, \varsigma_2, w_i)$ is the entropy of the $j-$th classifier and $H(\varsigma_1, \varsigma_2; w_0, \ldots, w_K)$ denotes the *total entropy* onto the total set of classes $\{w_0, \ldots, w_K\}$. From (16) and noting that $D(x) \geq 0$, we see that the inequality (17) holds

$$H(\varsigma_1, \varsigma_2; w_0, \ldots, w_K) \leq \sum_j \sum_{w_i} H_j, \tag{17}$$

with equality iff the *conditional mutual information* equals zero $D(x) = 0$, *i.e.* the multivariate probabilities is fully factorized $p(\varsigma_1, \varsigma_2|w_i) = p(\varsigma_1|w_i)p(\varsigma_2|w_i)$. The mutual information between the two classifiers $\mathcal{C}_1$ and $\mathcal{C}_2$ measures the quantity of

[†] Although KL divergence does not satisfy the axiom of symmetry [16].

[‡] This is a direct consequence of the (strict) convexity of the negative logarithm. This is not a proper distance measure because it is not symmetric.

information that each classifier conveys about the other; this can be considered as a measure of statistical correlation between the classifiers (see Cardoso [3] and Oja [19]). In other words, suppose ς_2 and $D(x)$ are important, then the knowledge of ς_2 does not give us much more information (most information in ς_1 is already in ς_2). Hence, a small value of $D(x)$ is preferable for combination of classifiers.

4.1 Estimation of the independence measure

Let us denote $n^{w_i}_{i_1,i_2} \triangleq \sum_{i_3} \dots \sum_{i_K} n^{w_i}_{i_1,i_2,\dots,i_M}$ for simplicity. Note that such calculus requires only few arrangement from the proposed algorithm in section 3. The various probabilities given in equation (14) can be estimated by the following marginals:

$$p(\varsigma_1, \varsigma_2, w_i) = \frac{n^{w_i}_{i_1,i_2}}{\sum_{w_i} \sum_{i_1,i_2} n^{w_i}_{i_1,i_2}}, \tag{18}$$

$$p(\varsigma_1, \varsigma_2 | w_i) = \frac{n^{w_i}_{i_1,i_2}}{\sum_{i_1,i_2} n^{w_i}_{i_1,i_2}}, \tag{19}$$

$$p(\varsigma_1 | w_i) = \frac{\sum_{i_2} n^{w_i}_{i_1,i_2}}{\sum_{i_1,i_2} n^{w_i}_{i_1,i_2}}, \tag{20}$$

$$p(\varsigma_2 | w_i) = \frac{\sum_{i_1} n^{w_i}_{i_1,i_2}}{\sum_{i_1,i_2} n^{w_i}_{i_1,i_2}}. \tag{21}$$

By inserting Eqs. (18-21) in (14), we obtain

$$D(x) = \sum_{w_i} \sum_{i_1,i_2} \frac{n^{w_i}_{i_1,i_2}}{\sum_{w_i} \sum_{\alpha,\beta} n^{w_i}_{\alpha,\beta}} \log \frac{\frac{n^{w_i}_{i_1,i_2}}{\sum_{w_i} \sum_{\alpha,\beta} n^{w_i}_{\alpha,\beta}}}{\frac{\sum_{\alpha} n^{w_i}_{i_1,\alpha}}{\sum_{\alpha,\beta} n^{w_i}_{\alpha,\beta}} \frac{\sum_{\alpha} n^{w_i}_{\alpha,i_2}}{\sum_{\alpha,\beta} n^{w_i}_{\alpha,\beta}}}, \tag{22}$$

and, after some algebra manipulation:

$$D(x) = \frac{1}{\sum_{\alpha,\beta} \sum_{w_i} n^{w_i}_{\alpha,\beta}} \sum_{i_1,i_2} \sum_{w_i} n^{w_i}_{i_1,i_2} \log \frac{n^{w_i}_{i_1,i_2} \sum_{\alpha,\beta} n^{w_i}_{\alpha,\beta}}{(\sum_{\alpha} n^{w_i}_{\alpha,i_2})(\sum_{\alpha} n^{w_i}_{i_1,\alpha})}. \tag{23}$$

5 Application to M-ary classifier to iris recognition

Iris recognition combines computer vision, pattern recognition, statistics. The purpose is real-time, high confidence recognition of a person's identity by mathematical analysis of the random patterns that are visible within the iris of an eye from some distance. Because the randomness of iris patterns has very high dimensionality, recognition decisions are made with confidence levels high enough to support rapid and reliable exhaustive searches through large-sized databases. Iris recognition technology identifies people by the unique patterns of the iris - the colored ring around the pupil of the eye. Features extraction used a 4 step protocole: (*i*) localisation of the eye regions in the face, (*ii*) detection of the outlines of the eyes, (*iii*) localisation of the pupils, (*iv*) extraction of the gradient vector field.

The pupil detection used the luminance image of the face: the image of the face is divided in 4 almost equal rectangles of size $\ell/2 \times L/2$ where ℓ is the width of the face and L its height. On each eye region, the luminance image is then computed and filtered by a retinian filter (to correct local variations of light), from which one extract the norm and the direction of the gradient. Conversion of an iris image into a numeric code that can be easily manipulated is essential to its use. A first process developed by Daugman [4], permits efficient comparisons based on information from a set of gabor wavelets, which

are specialized filters bank that extract information from a signal at various locations and scales. Once the image have been obtained, an *iris code* is computed based on information from a gradient field. Iris code derived from this process are compared with previously learned iris code.

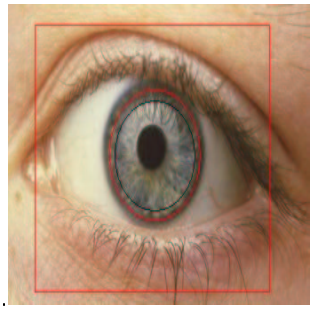
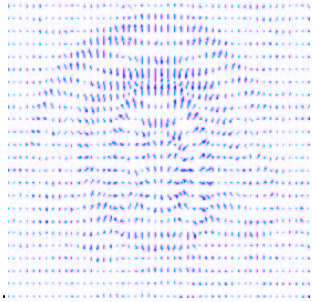
a.  b.  c. 

Fig. 1. (a) Eye region detection. (b) Iris outlines extraction. (c) Iris vector field computed.

5.1 Data collection

A serie of experiments[§] was performed for both tasks (i) evaluating features, (ii) evaluating recognition performance by combining multiple classifiers. The database and features to be used are as follows: let $\mathcal{B}_0 = \{(x_1, w_1), \ldots, (x_n, w_n)\}$ be a $n = 30$ eye-images database, x_i a N-dimensional vector composed of a set of *feature cells* x_{ij} identified by the pixel $P_j = (a_j, b_j)$, $1 \leq j \leq N$, w_i the label attached to the i-th image. Two numerical features that have good recognition performance in practice are used in this experiment: the first feature called CGB (Contour-Based Gradient Distribution) [13] is computed by computing the Sobel operator onto the normalized mesh R and computing the gradient direction distribution map. The second feature, called DDD (Directional Distance Distribution) [18], is computed using distance information. Each pixel in the binary map R shoots rays in eight directions and each ray computes the distance to the pixel with opposite color (black or right). Both map CGB and DDD can be represented with a $N = 256-$dimensional feature vector[¶] $x = (x_1, \ldots, x_N)$. Due to the small size of the dataset $n \ll N$, performance evaluation of the agregated classifier is done by bootstrap (see Kallel *et al.* [14] for details). In this experiment, a M-ary classifier ($M = 100$) is trained on the basis of the algorithm reported in Tab. 1: one half with CGD-based inputs, one half with DDD-based inputs. A *partial classifier* is a classifier which takes into considerations only $N'-$inputs, $N' \ll N$ (see [22]).
For a given classifier, N' randomly chosen positions in the gradient vector field of the mesh R are memorized. For a new image, gradient values are collected at the same positions. Other positions are randomly selected for a new classifier. Hence, we evaluate the $M-$ary classifiers in supposing a "*degenerate*" feature space.

5.2 Method

Classically, if there are numerous data, the first step consists in the division of the supplied data into two sets : a *test set* and a *training set*. This is not possible here due to the small size of the dataset. Our last resort since no classical inference is possible due to the intrinsic

[§] Private dataset, property iof the LIS.

[¶] CGB contains the local information about the image because the edge operator can extract only the local gradient direction information. On the contrary, DDD capture the global information since directional distance information provides a rough sketch of the global pattern.

complexity of the problem is to construct an estimate of the density function without imposing structural assumptions. Using *resampling methods* such as bootstrap [7], the information contained in the observed dataset $\mathcal{B}_0$, drawn from the empirical distribution $\mathcal{F}_0$ such that $\{(x_1, w_1), \ldots, (x_n, w_n)\} \overset{iid}{\sim} \mathcal{F}_0$, is extended to many typical generated data sets $\mathcal{B}^{*b}, 1 \leq b \leq B$ such that $\{(x_1^{*b}, w_1^{*b}), \ldots, (x_n^{*b}, w_n^{*b})\} \overset{iid}{\sim} \mathcal{F}^{*b}$. These samples are called bootstrapped samples (see [2]).
In our framework, $B = 200$ replications randomly drawn from the initial sample by resampling *with repetitions*. Tab. 3 describes the training algorithm. We look then for the winner class j represented by the code vector $z_j \in \mathbb{R}^{R'}$,

$$j = \arg \min_{1 \leq i \leq K} \|x - z_i\| \tag{24}$$

The agregated classification response $y^{*b}_{i_1,\ldots,i_K}(x)$ is then updated according to Tab. 1.

TRAINING ALGORITHM
input 30 Initial CGD and DDD R meshes
init. $\mathcal{B}_0$=Empty list; $\mathcal{B}_b^*$=Empty lists $(1 \leq b \leq B)$;
FOR EACH $\mathcal{C}_k, 1 \leq k \leq M$
1 Choose pixel $P_i^{(k)} = (a_i^{(k)}, b_i^{(k)}), 1 \leq i \leq N' \ll N$ randomly in the mesh R.
2 Get $\{x_1^0, \ldots, x_{N'}^0\}$ with $x_i^0 \leftarrow P_i^{(k)}$;
ENDFOR
3 Draw random samples $\mathcal{B}_b^*, 1 \leq b \leq B$ with replacement from $\mathcal{B}_0$.
FOR EACH $\mathcal{B}_b^*, 1 \leq b \leq B$
4 Construct the classifier $\mathcal{C}_k^{*b}, 1 \leq k \leq M$ (see Tab.1)[a] using the bootstrap sample.
5 Collect the agregated classification responses $y^{*b}_{i_1,\ldots,i_K}(x)$.
ENDFOR

Table 3. Experimental protocole including bootstrap resampling method.

[a] The choice of classifier type has few impact on the results.

At the end, in Tab.3, a new observation x is classified by majority voting using the predictions of all classifiers. Tab. 4 compares the recognition rates based on DDD, CDG and mixed CGD-DDD based classifiers using the original samples $\mathcal{B}_0$. It is apparent that the CGD feature has a better recognition performance than the DDD feature, this means a better discriminating power. Combination of both CGD-based and DDD-based classifiers was also tested and show some improved performance. Further test show that these recognition rates are improved in all case when the size of the pattern vector is larger. Although 3% may appear to be a small increase, it should be borne in mind that even small percentage increases are difficult to generate when the overall classification accuracy level exceeds 80%. We can therefore conclude that bootstrap is a useful technique for improving the performance of classifier. Thus, this algorithm forces the classification to concentrate on those observations that are more difficult to classify. We can also conclude that the rank-based classifiers produce significantly improvement over the class-based classifiers. Note that no test step is necesary with bootstrap.

6 Conclusion

A non parametric method for multiple classifiers fusion was describe and evaluated on a biometrics data. As our experimental results indicate good performance classification is archieved although there is room for improvement and although these results are not comparable with the recognition rate proposed with industrial devices (see [4]). The aggregated classifier scheme proposed exploits the simple fact

classifier	RANK-BASED (%)		CLASS-BASED (%)	
CGD	99,521	$\pm 3,07$	96,65	$\pm 6,40$
DDD	99,492	$\pm 22,36$	96,55	$\pm 6,40$
CGD+DDD	99,638	$\pm 11,0$	96,98	$\pm 7,34$

Table 4. Comparison of recognition rate (%) with boostrap standard deviation (No rejection).

that consensus decision produce a significant improvement over the original classifiers. An important merit lies in the low computational cost. The method could ten be used to speed up processing of larger datasets consisting of 100.000 cases or more. The price paid, in terms of drop of accuracy was rather small.
It is by no means clear, whether the same method could also be used in conjunction with other types of classifiers. It could be expected that the BBC method when used in conjunction with a k-NN classifer would not only deliver faster results, but also archieve higher accuracies.

References

1. U. Beyer and F. Smieja. Learning from exemples, agent teams, and the concept of reflection. *International Journal of Pattern Recognition and Artificial Intelligence*, 1994.
2. L. Breiman. Bagging predictors. Rapport technique TR-421, Statistics Department, University of California, Berkeley, 1994.for our model.
3. J.F. Cardoso. Multidimensional independent component analysis, *Proc. ICASSP'98*, 4, pp. 1941-19944, 1998.
4. J. Daugman. high confidence visual recognition of persons by a test of statsitical independence", IEEE Trans. PAMI, 1, Nov. 1993, pp. 148-160.
5. L. Devroye, L. Gyrfi and G. Lugosi. *A probabilistic theory of pattern recognition*, Application of Mathematics, Springer, 1991.
6. H. Drker, C. Cortes, L.Jackel, Y. Lecun and V. Vapnik. Boosting and other ensemble methodes. *Neural Computation*, 6(**6**), pp. 1289-1301, 1994.
7. B. Efron (1979) The convex hull of a random set of points. *Biometrika*, **52**, p 331–342.
8. Y. Freund, H. Seung, E. Shamir and N. Tishby. Information, prediction and query commitee. *Neural Information Processing Systems*, S. Hanson, J. Cowan, C. Giles ed., pp. 483-490, Denver, 1993.
9. J. Gubner. Distributed estimation and quantization. *IEEE Trans. on Information Theory*, 39(**4**), pp. 1456-1459, 1993.
10. L.K. Hansen and P. Salamon. Neural network ensembles. *IEEE Trans. on PAMI*, 12, pp. 993-1001, 1990.
11. L.K. Hansen, C. Lsberg and P. Salamon. Ensemble methods for recognition of handwritten digits. *Proc. IEEE Signal Processing Workshop*, S.Y. Kung, F. Fallside, J.A. Sorensen and C.A. Kamm Ed.; Piscataway, NJ, pp. 540-549, 1992.
12. M.I. Jordan and R.A. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6, pp. 181-214, 1994.
13. A.K. Jain. *Fundamentals of digital image processing*, Prentice-Hall Inc., 1989.
14. R. Kallel, M. Cottrell and V. Vigneron. Bootstrap for neural model selection. *Proc. of the* 8^{th} *European Symposium on Artificial Neural Networks*, Bruges, 22-24 Avril, 2000.
15. J. Kittler and M. Hatef. Improving recognition rates by classifier combination. *Proc. IWFHR'96*, pp. 81-101, 1996.
16. A.N. Kolmogorov and S.V. Fomin. *Elements of the theory of functions and functional analysis*, Vol. 1, Graylock Press, 1957.
17. J.P. Nadal, R. Legault et C. Suen. Complementary algorithms for the recognition of totally unconstrained handwritten numerals. 10^{th} *International Conference on Pattern Recognition*, pp. 443-446, Atlantic cit, 1990.

18. I.S. Oh, J.S. Lee and C.Y. Suen. Analysis of class separation and combination of class dependent features for handwriting recognition, *IEEE Trans. on PAMI*, 21(**10**), pp. 1099-1125.
19. A. Hyvrinen, J.Karhunen and E. Oja. *Independent component analysis*, John Wiley & Sons, 2001.
20. J. Pearl , D. Geiger and T. Verma. Conditional independence and its representations. *Kybernetika*, 25(**2**), 1989.
21. B.D. Ripley. *Pattern recognition and neural networks*, Cambridge University Press, 1996.
22. C.M. Soares, C.L. Fróes da Silva, M. De Gregorio and F.M.G. França. A software implmentation of the WISARD classifier, *Proceeding of Brasilian Symposium on Artificial Neural Network*, Belo Horizonte, MG, december 9-11, 1998, vol. II, 225-229.
23. K. Tumer and J. Ghosh. A framework for estimating performance improvements in hybrid pattern classifiers. *Proc. World Congress on Neural Networks*, San Diego, pp. 220-225, 1994.
24. L.G. Valiant. A theory of the learnable. *Communications of the association for computing machinery*, 27, pp. 1134-1142, reprint in Shavlik & Dietterich, 1990.
25. D. Wolpert. Stacked regression. *Neural Networks*, 5(**2**), pp. 241-260, 1992.
26. J.T. Tou and R.C. Gonzalez. *Pattern recognition principles*, Addison-Wesley, 1974.
27. A. Zapranis, A.-P. Refenes (1999) *Principles of Neural Model Identification, Selection and Adequacy*, Springer, London.