Information Theoretic Combination of Classifiers With Application to Multiple SVMs

Julien Meynet and Jean-Philippe Thiran
Ecole Polytechnique Fédérale de Lausanne (EPFL)
Signal Processing Institute
CH-1015 Lausanne, Switzerland.

Technical report TR-ITS-2006.15

October 11, 2006

Abstract

Combining several classifiers has proved to be an effective machine learning technique. Two concepts clearly influence the efficiency of an ensemble: the diversity between classifiers and the individual accuracies of the classifiers. In this paper we propose an information theoretic framework to establish a link between these quantities and, as they appear to be contradictory, we propose an information theoretic measure that expresses a trade-off between individual accuracy and diversity. This technique can be directly adapted for the selection of an ensemble in a pool of classifiers. We then apply this theory to the particular case of multiple Support Vector Machines using this new measure. We propose an adaptation of the Kernel-Adatron algorithm for learning online multiple SVMs. The results are compared to standard multiple SVMs techniques on reference datasets.

keywords: Combination of Classifiers, Information Theory, SVMs

1 Introduction

In many pattern recognition tasks, combining the decisions of several classifiers has shown to be an effective technique for improving the classification performances. Dietterich gives in [1] three main reasons why an ensemble of classifiers may be a better choice than a *monolithic* classifier. First, when the same learning accuracies can be achieved by several classifiers, there is more probability being close to the optimal solution by averaging all the decisions than by just picking one of them randomly. Then, many learning techniques use local searches to converge toward a solution (e.g. neural networks techniques), with the risk of staying stacked in local optima. Running several searches and combining the solutions can improve the performances. Finally, from a representational point of view, it is possible that the class of functions chosen for learning the classifier does not contain the optimal solution (e.g. restrict the use of a certain type of kernels in suport vector optimization). Combining several functions of this class allows to reach solutions outside of this class. In [2], Freund and Shapire also discuss why averaging classifiers can avoid overfitting.

Many techniques have been proposed in the past few years for combining classifiers. On the first hand, the classifiers can be either trained on different sample subsets or different

feature sets or use various learning algorithms. Then the combination itself can be performed according to two strategies: non trainable combiners (majority vote, probability rules: sum, product, mean, median, etc.) and trainable combiners (weighted majority vote, classifier as combiner, etc.). More detailed surveys on classifier combination can be found in [3] and [4]. The use of an ensemble is only justified if it becomes better than its best individual member. To achieve this requirement, classifiers need to commit errors on different new data. This concept refers to the notion of diversity which will be discussed widely throughout this paper. An overview of the different diversity measures is given in [5].

In this paper, we analyse the classifier combination process in an information theoretic framework. We define a new measure of the goodness of an ensemble of classifiers which is based on the trade-off between the individual accuracies and the diversity between the classifiers. This new measure is then taken into account for learning ensembles of Support Vector Machines (SVMs).

The paper is organized as follows: after reviewing in section 2 some basic notions of information theoretic classification, we present information theoretic combination of classifiers in Section 3. Section 4 presents techniques for learning ensembles of classifiers in this information theoretic framework as well as an experimental setup for evaluating these algorithms and comparing the results to the state-of-the-art. Finally, we draw some conclusions in Section 6.

2 Introduction to Information Theoretic Classification

Information theoretic classification was first introduced by Principe et al. in [6]. We summarize its concept here: the classification problem is formulated through the following first order Markov chain:

$$C \rightarrow \hat{C} \rightarrow E, \tag{1}$$

where C represents the true class labels defined over the set Ω_c, $\hat{C}$ models the classification steps through the decided class labels (feature extraction, feature selection and classification) and E is the error random variable taking values into $\{1, 0\}$. The probability of making an error during the classification process is thus:

$$P_e = P(E = 1) = P(\hat{C} \neq C). \tag{2}$$

Fano's inequality [7] gives a lower bound on this probability of error:

$$P_e \geq \frac{H_S(C|\hat{C}) - 1}{\log |\Omega_c|} = \frac{H_S(C) - I_S(C;\hat{C}) - 1}{\log |\Omega_c|}, \tag{3}$$

where $H_S(C) = -\sum_{k \in \Omega_c} p(C_k) \log p(C_k)$ is Shannon's entropy [8] of C, $I_S(C;\hat{C}) = \sum_{k,j \in \Omega_c^2} p(C_k, \hat{C}_j) \log \frac{p(C_k,\hat{C}_j)}{p(C_k)p(\hat{C}_j)}$ is Shannon's Mutual Information (MI) between C and $\hat{C}$ and $|\Omega_c|$ is the number of classes.

From this lower bound Erdogmus et al. [9] also derived an upper bound using Jensen's inequality described in [8]:

$$\frac{H_S(C) - I_\alpha(C;\hat{C}) - h_S(P_e)}{\log |\Omega_c| - 1} \leq P_e \leq \frac{H_S(C) - I_\beta(C;\hat{C}) - h_S(P_e)}{\min_k H_S(C|e, \hat{c}_k)}, \tag{4}$$

where $h_S(P_e) = -P_e \log P_e - (1-P_e) \log (1 - P_e)$ is the binary Shannon's entropy and $I_\alpha(C;\hat{C})$ represents Renyi's definition of the mutual information with $\alpha \in \mathbb{R}^+ \setminus \{1\}$. The tightest

bounds are obtained when the Renyi's entropy coefficients (α, β) tend to 1 in which case Renyi's definitions correspond to Shannon's ones. As the number of classes $|\Omega_c|$ is fixed, $H_S(C)$ does not depend on the classification process, bounds in Eq. 4 point out that maximizing the MI between the two random variables C and $\hat{C}$ will tend to minimize the probability of making an error Pe.

This formulation of the classification problem has been extended to feature extraction (Fisher et al. [10], Hild et al. [11]) and processing of multimodal signals (Butz et al. [12]). Sindhwani et al. [13] also proposed a feature selection technique for support vector machines and neural networks based on similar information theoretic considerations. In the next section we will extend these properties to the framework of multiple classifiers.

3 Information Theoretic Combination of Classifiers

The information theoretic framework given in the previous section was referring to the general classification task. This section shows how it can be extended when the classification problem is more specifically a combination of several classifiers.

Let us assume that we have a team of given classifiers. The aim is to find the best combination of members in the sense that it will maximize $I(C;\hat{C})$. For simplification and without loss of generality, let us consider a two class problem with labels $\{-1, 1\}$ and three classifiers to be combined. We denote by $\hat{C}_i$, $i \in \{1, 2, 3\}$ the random variables representing the output labels of classifier i, $I_{C;\hat{C}_i}$, $i \in \{1, 2, 3\}$ the MI between the output of individual classifier i and the true labels and finally $I_{i,j} = I_{\hat{C}_i;\hat{C}_j}$, $i, j \in \{1, 2, 3\}$, $j \geq i$ the MI between two classifiers. The quantity to be maximized is:

$$I_{C;\hat{C}} = \sum_{k=-1,1} \sum_{j=-1,1} P_{C;\hat{C}}(k, j) \log \frac{P_{C,\hat{C}}(k, j)}{P_C(k) P_{\hat{C}}(j)}. \tag{5}$$

As described in the introduction, the combination can be implemented using variety of strategies. The simplest rule and the most widespread is the majority voting. Majority voting simply considers the output label of each member, whereas other probability rules take into account output confidences of the classifiers. Despite its simplicity, it has proven to be an effective rule for many combination tasks. Moreover, majority vote can easily be extended to weighted majority vote which is widely used in the multiple classifiers community. For example, the decision of AdaBoost [14] is a weighted majority vote of weak classifiers. Many studies [15, 16, 17, 18] have focused on analysing why majority voting was as effective as more complicated schemes in improving the recognition results. In the remaining of the paper we restrict the combination rule to majority voting.

3.1 Majority Vote for Combining Classifiers

Considering a majority voting scheme, the probability $P_{\hat{C}}(i)$ that $\hat{C}$ outputs i is is related to each voter classifier:

$$P_{\hat{C}}(i) \leq P_{C_1,C_2}(i) + P_{C_1,C_3}(i) + P_{C_2,C_3}(i). \tag{6}$$

Then, considering an odd number of independent classifiers $N > 1$ and assuming that they

all have the same accuracy denoted p, the accuracy of the ensemble is:

$$P_{maj} = \sum_{m=\lfloor N/2 \rfloor+1}^{N} \binom{N}{m} p^m (1-p)^{N-m}. \tag{7}$$

The following result is known as the Condorcet Jury Theorem (1785):

Theorem 3.1 *[3]*

1. *If $p > 0.5$ then P_{maj} is monotically increasing and $P_{maj} \longrightarrow 1$ as $N \longrightarrow \infty$*
2. *If $p < 0.5$ then P_{maj} is monotically decreasing and $P_{maj} \longrightarrow 0$ as $N \longrightarrow \infty$*
3. *If $p = 0.5$ then $P_{maj} = 0.5$.*

The assumptions of equal accuracy and independence of the classifiers are of course too strong in our framework as each classifier is trained using features extracted from the same data. Moreover, this theorem does not tell about the non asymptotic behavior of the majority rule. How does majority voting behave with a small number of classifiers N? To address this ambiguity, Shapley et al. in [19] give the following lemma:

Theorem 3.2 *[19]*
Consider a group of odd size N with any competence structure $(p_1, \ldots, p_N)$, where $p_i > 0.5$ $\forall i$. The probability to reach the correct decision, when utilizing the simple majority rule, is larger than the probability $p = \frac{1}{N}\sum_{i=1}^{N} p_i$ of a random group member to do so.

In our case this theorem leads to:

$$P_{C,\hat{C}}(i) \geq \frac{1}{3}(P_{C,C_1}(i) + P_{C,C_2}(i) + P_{C,C_3}(i)). \tag{8}$$

Considering bounds (6) and (8) on each term of the mutual information in Eq. (5), we show that minimizing the MI between each pair of classifier $I_{\hat{C}_i;\hat{C}_j}, i \neq j$ and maximizing the MI between each single classifier and the true class labels $I_{C;\hat{C}_i}$ will tend to maximize $I_{C;\hat{C}}$.

As introduced in section 2, $I_{C;\hat{C}_i}$ represents the accuracy of classifier i, while $I_{\hat{C}_i;\hat{C}_j}$ measures the similarity between the two classifiers i and j. Thus, by minimizing $I_{\hat{C}_i;\hat{C}_j}$, we maximize the diversity between the two classifiers.

It is important to note that we proved a sufficient condition for maximazing $I(C;\hat{C})$, but it is clearly not a necessary condition. It is possible to have a good combiner accuracy which does not maximize the ratio between the classifiers accuracies and the diversity. This will be discussed experimentally in section 4.

3.2 Diversity

Diversity appears to be a key feature in obtaining an effective combination process. Clearly, in order to be efficient, an ensemble needs to contain classifiers that are complementary, in

the sense that they commit errors on different objects. That is why many papers proposed to directly exploit diversity for finding good ensembles [20, 21, 22, 23, 24]. However, this section will show that diversity is a more ambiguous concept than it seems to be. In the past years, various diversity measures have been preposed. They can be splitted into pairwise and non pairwise diversites, the most widespread being the Q statistic [25], Double fault [26] and the Disagreement Measure [27]. Let us define the Q-statistics between two classifiers C_1, C_2 with the notations of Section 2. Denote $a = P(C_1 = C, C_2 = C)$, $b = P(C_1 \neq C, C_2 = C)$, $c = P(C_1 = C, C_2 \neq C)$ and $d = P(C_1 \neq C, C_2 \neq C)$, then $Q_{C_1,C_2} = \frac{ad-bc}{ad+bc}$.

We define the following experimental setups to evaluate how the diversity influences the performances of an ensemble of classifiers, using two diversity measures: the Q-statistics and our information theoretic measure.

A first experiment imposes the classifiers to have strictly equal accuracies $p \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$. Considering this constraint, 1000 binary outputs were randomly generated for each classifier. For each trial we measured the ensemble accuracy *pvote* and we compared two diversity measures: the average Q-statistics and the average MI. Results are reported in Figure 1.

In the second experiment p is randomly distributed in the interval $p \in [0.7\,, 0.8]$. Results are reported in Figure 2.

It turns out that the diversity (both Q-statistics and MI) seems to be a relevant feature when the classifiers have similar individual accuracies, supposing that this accuracy is not too low (Figure 1 with $p \geq 0.7$). When they have equal but low accuracies, large diversity does not imply necessarily improvements (Figure 1 with $p < 0.7$). But Figure 2 shows that even if the classifiers have only slight differences in terms of individual accuracy, diversity between them is not a highly discriminant feature for choosing the best ensemble (This remark is important as in practical applications we cannot ensure that the classifiers have exactly the same individual accuracy).

These considerations explain why diversity is usually only used for visualization (plot pairs of classifiers according to their diversity), or overproduction and selection of classifiers. Conceptually, forcing diversity in an ensemble seems to encourage the global accuracy of the ensemble. However, Kuncheva reported in [3] that the improvement on the best individual accuracy by forcing diversity is negligible. More details about diversity and how to create diversity in ensemble are given in [28]. To understand this problem from an information theoretic point of view, the following section will discuss the accuracy/diversity dilemma.

3.3 Diversity/Accuracy Dilemma

We pointed out in Section 3 that our aim is to maximize both the average individual accuracy and the diversity between classifiers. However these two measures are somehow contradictory. In fact two very good classifiers will clearly have very low diversity and vice-versa. We discuss this phenomenon by the following formalism.

Consider two random variables C_1, C_2 representing two classifiers. Let C be the true class labels.

To establish a probabilistic link between the 2 classifiers, a parallel is made with the work of Butz et al in [12] concerning processing of multi-modal signals. First recall some pattern recognition definitions. We consider that the training and testing examples are generated from an unknown but fixed probability distribution function (*pdf*) and the task is to find a

function that minimizes the risk of misclassifying new vectors drawn from the same *pdf*. We can consider that the inputs of both classifiers C_1 and C_2 come from this *pdf*. Two coupled Markov chains can be built:

$$\begin{cases} C \rightarrow C_1 \rightarrow \hat{C}_2 \rightarrow \hat{C} \rightarrow E \\ C \rightarrow C_2 \rightarrow \hat{C}_1 \rightarrow \hat{C} \rightarrow E. \end{cases} \tag{9}$$

These coupled Markov chains are depicted in Figure 3. The probability densities of C_1 and $\hat{C}_1$, resp. C_2 and $\hat{C}_2$, are both estimated from the same data sequences. Therefore we can write $I(C_1;\hat{C}_2) \approx I(C_2;\hat{C}_1) \approx I(C_1;C_2)$. Then, the data processing inequality as defined in [8] gives: $I(C_1;C_2) \geq I(C;C_2)$ and $I(C_1;C_2) \geq I(C;C_1)$. This implies that:

$$I(C_1;C_2) \geq \frac{I(C;C_1) + I(C;C_2)}{2}. \tag{10}$$

Maximizing the individual accuracies represented by $I(C;C_1), I(C;C_2)$ will consequently maximize $I(C_1;C_2)$, the diversity between the classifiers. Inversely, minimizing $I(C_1;C_2)$ (maximizing the diversity) will tend to minimize the classifiers accuracy. To adress the contradiction presented here, a trade-off needs to be introduced. A study of how the diversity evolves depending on the classifiers accuracies is given in the next section.

4 Information Theoretic Score

4.1 Estimation of the Relationships Between Diversity and Classifiers Accuracy

This section estimates experimentally the relationship between diversity and accuracy in order to give a computable measure of the ensemble performance. This link is estimated with the following experiment. Outputs of two classifiers (C_1, C_2) with equal accuracies are iteratively simulated. We report in Figure 4 the similarity between output labels $I(C_1;C_2)$ for each trial as a function of the individual accuracy $\frac{I(C;C_1)+I(C;C_2)}{2}$.

We propose to approximate the similarity by a quadratic function of the average individual accuracy. Figure 5 gives a graphical interpretation of this approximation. A classifier is represented by a vector. Its projection onto the horizontal axis measures its individual accuracy while the difference between vertical projections of two vectors measures the diversity between them. The dash line represents the maximal diversity allowed between two classifiers with identical accuracy. It appears that two poor classifiers can have large diversity while two accurate classifiers cannot be so diverse.

In the following, we will consider two terms based on the mutual information between classifiers: the average accuracy of the K classifiers:

$$ITA = \frac{\sum_{i=1}^{K} I(C;C_i)}{K}, \tag{11}$$

and the diversity between the classifiers:

$$ITD = \frac{\binom{K}{2}}{\sum_{i=1}^{K-1} \sum_{j=i+1}^{K} I(C_i;C_j)}. \tag{12}$$

Taking into account the second order approximation of the similarity between the classifiers and the average accuracy, we propose the Information Theoretic Score (ITS) as:

$$ITS = (1 + ITA)^3.(1 + ITD). \tag{13}$$

This model is a choice and other similar modeling could be chosen. The next section tries to validate this definition in the context of overproduction and selection of classifiers.

4.2 Validation of the ITS

To evaluate the intrinsic behavior of the ITS, we first consider artificial classifier outputs. By generating random outputs we can explore the complete space of output labels. It presents the advantage of being completely independent of the process of feature selection and independent of the learning algorithm. We can thus perform an unbiased evaluation of the ITS. Let us consider the following simple experimental setup. We generate randomly outputs for three classifiers. For each run, we measure the accuracy of the ensemble and the ITS. The results are shown in Figure 6. Note that in this experiment we do not impose the individual accuracies to be identical.

As expected, ensembles with high ITS are accurate. Moreover, an ensemble can be accurate but with a low ITS, therefore, the condition for maximizing $I(C; \hat{C})$ is sufficient but not necessary.

4.3 Experiments With Real Classifiers

For evaluating the relevance of the ITS defined above on a real classification task, we consider a 2 class toy problem using the Banana dataset available in the Matlab Pattern Recognition Toolbox [29]. We generate 1000 training examples for both classes and we split this training set into 15 smaller subsets by random sampling. We then train one classifier with each subset. A first experiment (Figure 7(a)) consists in training 15 Support Vector Machines (SVMs) with radial basis kernels (the parameters being evaluated by cross-validation). The 455 possible combinations of three classifiers (triplets) are exhaustively tested. For each triplet, we measure the ITS, the ensemble accuracy on a large test set and the we also compute the average individual accuracy of the three classifiers. This mean is represented by the grey level of the circles in Figure7(a). In the second experiment, three different learning algorithm are used. We trained 5 SVMs, 5 linear classifiers and 5 K-nearest neighbors (KNN) and again ITS is measured for each triplet. Results are reported in Figure 7(b).

As expected, the triplets of classifiers with low ITA (dark circles) lead to low classification accuracy. When the three individual classifiers are accurate (light circles in Figures 7(a) and 7(b)), the final classification is generally accurate. However, in both configuration, the lightest points (which means the 3 best classifiers combined together) do not give necessarily the best combination. This phenomenon is more visible in the case of 15-SVMs as they only have slight differences in their individual accuracies. In any case, the ensembles with high ITS are very accurate. These experiments show that, at least in toy problems, the ITS can overcome the limitations of diversity as presented in section 3.2.

5 Designing Ensembles of Support Vector Machines using ITS

The previous section showed the interest of using the ITS for selecting classifiers in a predefined team of classifiers. However, in many applications the classifiers are not given, that is why in this section we propose techniques for training ensembles such that ITS will be maximized. One of the drawbacks of the information theoretic approach is that the objective function to be optimized is not differentiable. To overcome this problem, we will use various iterative optimization algorithms for maximizing the ITS.

In most cases, these algorithms depend on the learning algorithm used for training the classifiers. In the remaining of the paper we will thus focus on the particular case of multiple Support Vector Machines which has proved to be efficient in many applications, particularly large scale problems.

5.1 An Overview of SVMs and Ensembles of SVMs

Support Vector Machines (SVMs) have been extensively used in many pattern recognition problems, mainly because of their impressive generalization performances compared to other algorithms.

Let us begin with a brief overview of the classical SVM algorithm. More information about SVM can be found in [30],[31]. Let $\{(\mathbf{x}_i, y_i) | i = 1, \ldots, l\} \subset \mathbb{R}^n \times \{-1, +1\}$ be a set of examples. From a practical point of view, the problem to be solved is to find that hyperplane that correctly separates the data while maximizing the sum of distances to the closest positive and negative points (i.e. *the margin*). The hyperplane is given by[1]:

$$h_{\mathbf{w},b}(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b = 0,$$

and the decision function is:

$$f(\mathbf{x}) = \operatorname{sgn}(h_{\mathbf{w},b}(\mathbf{x})) = \operatorname{sgn}\left(\langle \mathbf{w}, \mathbf{x} \rangle + b\right).$$

In the case of linearly separable data, maximizing the margins means to maximize $\frac{2}{\|\mathbf{w}\|}$ or, equivalently, to minimize $\|\mathbf{w}\|^2$, subject to $y_i(\langle \mathbf{w}, \mathbf{x} \rangle + b) \geq 1$. Suppose now that the two classes overlap in feature space. One way to find the optimal surface is to relax the above constraints by introducing the *slack variables* ξ_i and solving the following problem (using 2-norm for the slack variables):

$$\begin{array}{ll} \min_{\xi,\mathbf{w},b} \|\mathbf{w}\|^2 + C \sum_{i=1}^{l} \xi_i^2 & \\ \text{subject to} \quad y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i & \forall i = 1, \ldots, l, \end{array} \tag{14}$$

where C controls the weight of the classification errors ($C = \infty$ in the separable case).

This problem is solved by means of Lagrange multipliers method. Let $\alpha_i \geq 0$ be the Lagrange multipliers solving the problem above, then the separating hyperplane, as a function of α_i, is given by

$$h_{\alpha_i,b}(\mathbf{x}) = \sum_{i:\alpha_i > 0} y_i \alpha_i \langle \mathbf{x}_i, \mathbf{x} \rangle + b.$$

[1] We use $\langle \cdot, \cdot \rangle$ to denote the inner product operator

Note that usually only a small proportion of α_i are non-zero. The training vectors $\mathbf{x}_i$ corresponding to $\alpha_i > 0$ are called *support vectors* and are the only training vectors influencing the separating boundary.

In practice however, a linear separating plane is seldom sufficient. To generalize the linear case one can project the input space into a higher–dimensional space in the hope of a better training–class separation. In the case of SVM this is achieved by using the so–called "kernel trick". Basically, it replaces the inner product $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$ with a kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$which needs to satisfy Mercer's conditions. As the data vectors are involved only in this inner products, the optimization process can be carried out in the feature space directly. Some of the most used kernel functions are:

$$\begin{array}{ll} \text{the polynomial kernel} & K(\mathbf{x}, \mathbf{z}) = (\langle \mathbf{x}, \mathbf{z} \rangle + 1)^d \\ \text{the RBF kernel} & K(\mathbf{x}, \mathbf{z}) = \exp(-\gamma \|\mathbf{x} - \mathbf{z}\|^2) \end{array} \tag{15}$$

The main drawback of SVMs is that solving the problem requires an optimization with a complexity that varies at least quadratically with the number of training examples, which becomes intractable in large scale problems. To overcome this difficulty, several studies used mixtures of SVMs which lead to simpler optimization tasks.

Here is a brief overview of the most significant ones. A first technique referring to Mixtures of SVMs (MSVMs) was proposed by Kwok in [32]. He used several SVMs in a mixture of expert scheme and showed the efficiency of this technique with different hierarchical structures. Then Collobert et al. [33] trained several SVMs on subsets of the initial dataset to decrease the training complexity. The subdivision can be performed in many different ways. A simple but efficient technique is to sample the training set randomly [34, 33]. It is particularly efficient for large scale problems as it allows to reduce the noise present in the training set and decrease the influence of potential outliers. In [33], Collobert et al proposed to use Neural Networks for combining the decisions of the individual classifiers. The neural network gater was trained in order to minimize a squared error cost function. In [35], the outputs are combined using a second layer SVM trained on the margins ([35]) whereas in ([34]) they used simple probability rules. In case of simple majority voting, this last technique is similar to Bagging as introduced in [36] except that the sampling is done without replacement. An asymmetric version has been proposed for image retrieval using relevance feedback [37].

These studies also lie on interesting theoretical foundations showing the interest of using MSVMs rather than one single SVM trained on the complete training set. An interesting work focusing on generalization bounds of such kernel machines ensembles is presented in [38]. In particular, they show that the leave-one-out error $\mathcal{L}((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n))$ of M parallel SVMs is upper bounded by:

$$\mathcal{L}((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)) \leq EE_1 + \frac{1}{M} \sum_{m=1}^{M} \frac{D_m^2}{\rho_m^2},$$

where EE_1 is the margin empirical error with ensemble margin 1, D_m is the radius of the smallest sphere centered at the origin, in the feature space induced by m-th kernel, containing the support vectors of the m-th SVM, and ρ_m is the margin of m-th SVM . In some cases, this bound is smaller that the bound for a single SVM. For example suppose that the SVMs use most of their training points as support vectors, then clearly the D_m of each SVM in the ensemble is smaller than that of the single SVM. Moreover the margin of each individual SVM is expected to be larger than that of single SVM. We propose to see how the ITS defined

Population size	ITS-GA	Single SVM	Random sampling
6	87.1(%)		
10	89.5(%)	79.1	81.3
16	90.8(%)	(%)	(%)
20	91.5(%)		

Table 1: Comparison of 3 classification techniques on face detection datasets. A single SVM trained on the complete train set, mixtures of SVMs trained by random sampling and mixtures of SVMs obtained by GA using ITS as fitness function. Classification rates correspond to equal error rates.

in the previous sections can be used in this context of MSVMs. The proposed techniques will then be compared to the state-of-the-art.

5.2 Genetic Algorithms

In the MSVMs presented here above, the splitting of the training set is usually done by random sampling. However, this will lead to classifiers that are similar in the sens that they are trained from data that are just various estimates of the same distribution. We thus propose techniques for finding subdivisions of the initial training set that will finally maximize our ITS A first answer to this optimization problem is to use genetic algorithms (GA)[39].

GA have been found to be a robust and practical optimization method. A candidate solution is represented by a chromosome. A set of chromosomes (called population) is first generated randomly and it is then iteratively modified to converge towards the optimal. The relevance of each chromosome is measured using a fitness function. In our study, one chromosome represents one training set configuration for all the classifiers in the ensemble. It means that for each example, the chromosome encodes which classifiers will use the example in their training set. We use a coding based on a Venn diagram similar to the one proposed in [40] except that they work in the context of feature selection. The encoding is detailed in Figure 8. The evolution of the population if performed by crossover and mutations for adapting training sets of the 3 classifiers.

For evaluating this technique and give comparative results with MSVMs, we consider a face detection dataset. The classification task is to classify face images versus non face images. The training set contains 1000 face images extracted from BANCA Database [41] and 1000 non face training examples that were generated by bootstrapping on randomly selected images. These data as well as the large test set used in this paper are available upon request. The images were scaled at 15×20 pixels and then projected on a 18 dimensional space by Principal Component Analysis (PCA). (see [34] for details).

Results are compared to the technique used in [34] where they use random sampling to generate the training subsets and then combine the decisions using classical probability rules. The results are reported in table 1 . The classification rates reported in Table 1 correspond to equal error rates between face and non face classes.

With only few generations (e.g. 6 generations) the recognition rates are largely improved compared to simple random sampling introduced in [34].

This GA approach gives an ensembles with large ITS. However the training process is very

computationally expensive. In fact running the GA requires training $3 \times p \times g$ classifiers, where p is the size of the population and g the number of generations. While it remains efficient with non-trainable classifiers (e.g. k-nearest neighbors), it becomes practically infeasible with SVMs. The drawback of these non-trainable classifiers is that they are generally stable in the sense that small changes in the training set will not induce large changes in the output. Combining several classifiers becomes efficient when the classifiers are unstable (see [3]). Another drawback of this GA strategy is that the training complexity varies dramatically with the number of classifiers in the ensemble. In this work, only a setup with 3 classifiers has been tested and extending it to larger ensembles can be a very complex task. However this GA approach remains interesting for comparison purposes. It shows the classification rates that can be reached as well as the ITS measures obtained for good ensembles. In the following we propose a technique for achieving similar classification rates but with a much lower training complexity.

5.3 Kernel Adatron for maximizing ITS

In this work we use an online algorithm for training SVMs which is called Kernel Adatron (KA)[42]. An overview as well as implementation considerations can be found in [43]. KA simply uses a gradient ascent to solve the convex quadratic optimization described in Eq. 14. We extend this algorithm to train jointly M SVMs such that the ITS of the ensemble will be increased. The algorithm is described in Algorithm 5.1, where the index $x^{(c)}$ indicates that the variable x refers to the classifiers c. The standard formulation of KA is found by setting $M = 1$, $\mu = 0$. The adaptation to multiple classifiers appears in the function f_{ITS} weighted by the factor μ. It depends on the output of all the current classifiers. Basically, the Lagrange coefficients of the support vectors that are misclassified by the ensemble will be modified such that a majority of classifier classify correctly the support vectors. More precisely, let $f^{(m)} = \text{sign}\left(\sum_{j \in \text{SV}^{(m)}} y_j \alpha_j^{(m)} K^{(m)}(\mathbf{x}_i, \mathbf{x}_j)\right)$ be the decision of the m-th SVM at the current iteration. Then for all the support vectors, we compute the number of correct classification: if $\max\{\alpha^{(m)}\}_{m=1,\ldots,M} > 0$,

$$L = \sum_{m=1,\ldots,M} \mathbb{I}\left(f^{(m)}(\mathbf{x}_i) y_i > 0\right). \tag{16}$$

Then define $f_{ITS}^{(m)}$ as:

$$\begin{aligned} L^* &= \max\left(0, \lceil \tfrac{M}{2} \rceil + 1 - L\right) \\ f_{ITS}^{(m)} &= \begin{cases} +1 & \text{for the } L^* \text{ largest } \alpha_i^{(m)}, \text{with } m \in \{1, \ldots, M | f^{(m)}(\mathbf{x}_i) y_i < 0\} \\ 0 & \text{otherwise} \end{cases} \end{aligned} . \tag{17}$$

The parameter μ is a weighting coefficient that affects the convergence speed. Empirical studies using crossvalidation showed that μ should be chosen in the range $0.5\eta \leq \mu \leq 1.5\eta$. Clearly setting a too large μ will tend to overfit the training data. Note that it has been proved that if a large number of examples are considered as support vectors, the optimal choice is $\eta = \frac{1}{K(\mathbf{x}_i, \mathbf{x}_i)}$ which is 1 in case of RBF kernels. Note that that some iterations of the standard KA are ran before taking into account other classifiers in order to keep a fast convergence of the algorithm. This algorithms is used in the context of random sampling of the train set as described in Section 5.2. This technique performs an implicit clustering of the data such that each member of the ensemble behaves like an expert on its own subset.

This algorithm has been tested using the face dataset presented earlier. We first ran 5 iterations of standard KA (by setting $\mu = 0$) for training 3 SVMs on independent subsets , then μ is set to 1. At each iteration, we measure the ITS of the ensemble on a separate test set and compare it to the best ITS obtained by GA. Results are depicted in Figure 9. It clearly appears that the ITS increases significantly after the introduction of the joint term (5 iterations). It also shows that we quickely reach the ITS level of the computationally expensive GA technique. (The ITS level of the GA in Figure 9 is a mean of 10 trials,with a standard deviation of 0.04). Apart from its good classification skills, this algorithm presents also the advantage of being computationally friendly. Moreover, the complexity of the training process only increases linearly with the number of classifiers.

Algorithm 5.1: ITS - Kernel Adatron.

1 **Initialize** $\forall m \in \{1, \ldots, M\}$,
$\alpha_i^{(m)} = 0$

2 **Calculate** $\forall m \in \{1, \ldots, M\}$,

- $z_i^{(m)} = \sum_{j=1}^{n} \alpha_j^{(m)} y_j K(\mathbf{x}_i, \mathbf{x}_j)$
- $\delta\alpha_i^{(m)} = \eta(1 - y_i z_i^{(m)}) + \mu f_{ITS}^{(m)}$
- $\alpha_i^{(m)} = max(min(\alpha_i^{(m)} + \delta_i^{(m)}, C), 0)$

3 **Calculate new margins** $\forall m \in \{1, \ldots, M\}$,
$\gamma^{(m)} = \frac{1}{2}(\min_{\{i|y_i=+1,\alpha_i<C\}}(z_i^{(m)}) + \max_{\{i|y_i=-1,\alpha_i<C\}}(z_i^{(m)}))$

4 **Break if** $\forall m \in \{1, \ldots, M\}$, maximum number of iteration reached or the margin $\gamma^{(m)}$ has approached 1

5.4 Comparison of the Methods

The main goal of the experiments reported here is to investigate the behavior of the kernel Adatron adaptation of the MSVMs and to compare it with the other standard techniques. MSVMs are particularly effective in large scales applications that is why in our experiments we used the largest dataset available in the UCI repository [44]- the Forest dataset. We transformed the original multi-class problem into a binary classification task where the goal was to discriminate class 2 from all the other six classes, this kind of partitioning making the two new classes of roughly the same size. We took 10000 examples of each class for training and 30000 for testing. SVMs are trained using LIBSVM [45] by 5-fold cross-validation. We compare the following techniques: Single SVM trained on the complete dataset, Multiple SVMs (MSVMs) [34], Gated SVMs (GSVMs) [33], Genetic Algorithms using ITS as fitness function (GA-ITS), 1 monolithic SVM trained using Kernel Adatron (K-A 1-SVM), and finally 3 SVMs trained jointly using KA (K-A 3-SVMs). The results are reported in table 2.

The first comparison concerns a single SVM trained either by quadratic optimization (using the implementation in [45]) or the kernel Adatron implementation. They perform quite identically in terms of detection rates but the training time is much lower in the KA case. Then we see that the techniques [34, 33] improves the single classifier implementation as expected. GA-ITS and KA-3SVMs show how using a better subset selection than only random sampling significantly improves the results. Finnally the KA-3SVMs version converges much

Experiment	#SV	Detection rate(%)	Training time (mins)
1 SVM	966	72.4	195
3- MSVMs[34]	633+628+644	73.27 ± 0.12	70
Gated SVMs[33]	143+127+176	73.33 ± 0.14	62
GA- ITS	963+878+1035	75.02 ± 0.23	307
K-A 1SVM [42]	1331	72.77	19
K-A 3SVMs	1326+1345+1324	74.86 ± 0.12	23

Table 2: Comparison of various multiple SVMs techniques on UCI Forest database [44]. For each technique we report the number of support vectors (#SV), the test error on a large test set and the training time in minutes.

faster than GA-ITS. We notice that the number of support vectors is very high for the KA-3SVMs. It can be explained by the fact that contrarily to [34, 33], one input sample can be support vector for several of the 3 SVMs at the mean time.

6 Conclusions

This paper presents a new ensemble learning technique in an information theoretic framework. It provides a tool for measuring the goodness of an ensemble by taking into account a trade-off between individual accuracy and diversity. This information theoretic criterion has been used in learning of multiple SVMs. We propose an online algorithm for training multiple SVMs in this information theoretic framework. These techniques have been tested in the face class modeling application as well as a large scale problem, and they perform significantly better than state-of-the-art techniques.

Acknowledgments

This work is supported by the Swiss National Science Foundation through the National Center of Competence in Research on "Interactive Multimodal Information Management (IM2)".

References

[1] Thomas G. Dietterich, "Ensemble methods in machine learning," *Lecture Notes in Computer Science*, vol. 1857, pp. 1–15, 2000.

[2] Y. Freund, Y. Mansour, and R. Schapire, "Why averaging classifiers can protect against overfitting," in *In Proceedings of the Eighth International Workshop on Artificial Intelligence and Statistics*, 2001.

[3] Ludmila I. Kuncheva, *Combining Pattern Classifiers Methods and Algorithms*, John Wiley, New York,, New York, NY, USA, 2004.

[4] Josef Kittler, Mohamad Hatef, Robert P. W. Duin, and Jiri Matas, "On combining classifiers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 3, pp. 226–239, 1998.

[5] L. Kuncheva and C. Whitaker, "Measures of diversity in classifier ensembles," *Machine Learning*, vol. 51, no. 2, pp. 181 – 207, 2003.

[6] J.C. Principe, D. Xu, and J.W. Fisher, "Learning from examples with information theoretic criteria," *J. VLSI Signal Process. Syst.*, vol. 26, pp. 61 – 77, 2000.

[7] R.M. Fano, *Transmission of Information: A Statistical Theory of Communication*, MIT Press, Wiley, Cambridge, 1961.

[8] T. Cover and J. Thomas, *Elements of Information Theory*, John Wiley and Sons, Inc., New York, 1991.

[9] Deniz Erdogmus and Jose C. Principe, "Lower and upper bounds for misclassification probability based on renyi's information," *Journal of VLSI Signal Processing*, vol. 37, pp. 305–317, 2004.

[10] J. Fisher, III and J. Principe, "A methodology for information theoretic feature extraction," in *IEEE International Conference on Neural Networks (IJCNN'98)*, Anchorage, AK, 1998, vol. 3, pp. 1712–1716.

[11] Kenneth E. Hild II, Deniz Erdogmus, Kari Torkkola, and Jose C. Principe, "Feature extraction using information-theoretic learning," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 28, no. 9, pp. 1385–1392, 2006.

[12] Torsten Butz and Jean-Philippe Thiran, "From error probability to information theoretic (multi-modal) signal processing," *Signal Processing*, vol. 85, no. 5, pp. 875–902, 2005.

[13] Vikas Sindhwani, Subrata Rakshit, Dipti Deodhare, Deniz Erdogmus, Jose C. Principe, and Partha Niyogim, "Feature selection in mlps and svms based on maximum output information," *IEEE Transactions On Neural Networks,*, vol. 15, pp. 937–949, 2004.

[14] Yoav Freund and Robert E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, 1997.

[15] L.I. Kuncheva, C.J. Whitaker, C.A. Shipp, and R.P.W. Duin, "Limits on the majority vote accuracy in classifier fusion," *Pattern Analysis and Applications*, vol. 6, pp. 22–31, 2003.

[16] L. Lam and S.Y. Suen, "Application of majority voting to pattern recognition: An analysis of its behavior and performance," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 27, pp. 553–568, 1997.

[17] Narasimhamurthy, "Theoretical bounds of majority voting performance for a binary classification problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, pp. 1988– 1995, 2005.

[18] Dymitr Ruta and Bogdan Gabrys, "A theoretical analysis of the limits of majority voting errors for multiple classifier systems," *Pattern Analysis and Applications*, vol. 5, no. 4, pp. 333–350, 2002.

[19] L. Shapley and B. Grofman, "Optimizing group judgemental accuracy in the presence of interdependencies," *Public Choice*, vol. 43, pp. 329–343, 1984.

[20] C. A. Shipp and L. I. Kuncheva, "Relationships between combination methods and measures of diversity in combining classifiers," *Information Fusion*, , no. 3, pp. 135–148, 2002.

[21] Stefan Todorov Hadjitodorov, Ludmila I. Kuncheva, and Ludmila P. Todorova, "Moderate diversity for better cluster ensembles," *Information Fusion*, vol. 7, no. 3, pp. 264–275, 2006.

[22] G. Brown, J. Wyatt, R. Harris, and X. Yao, "Diversity creation methods: A survey and categorisation," *Journal of Information Fusion*, vol. 6, no. 1, pp. 5–20, March 2005.

[23] T. Windeatt, "Diversity measures for multiple classifier system analysis and design," *Information Fusion*, vol. 6, pp. 2136, 2005.

[24] Prem Melville and Raymond J. Mooney, "Creating diversity in ensembles using artificial data.," *Information Fusion*, vol. 6, no. 1, pp. 99–111, 2005.

[25] G. Yule, "On the association of attributes in statistics.," *Biometrika*, vol. 2, pp. 121–134, 1903.

[26] Giorgio Giacinto and Fabio Roli, "Design of effective neural network ensembles for image classification purposes," *Image Vision Comput*, vol. 19, no. 9-10, 2001.

[27] D. Skalak, "The sources of increased accuracy for two proposed boosting algorithms," in *AAAI '96 Workshop on Integrating Multiple Learned Models for Improving and Scaling Machine Learning Algorithms*, 1996.

[28] G. Brown, J. Wyatt, R. Harris, and X. Yao, "Diversity creation methods: A survey and categorisation," *Journal of Information Fusion*, vol. 6, no. 1, pp. 5–20, March 2005.

[29] R.P.W. Duin, P. Juszczak, P. Paclik, E. Pekalska, D. de Ridder, and D.M.J. Tax, "Prtools4, a matlab toolbox for pattern recognition," *Delft University of Technology*, 2004.

[30] Vladimir Vapnik, *The Nature of Statistical Learning Theory*, Springer Verlag, 1995.

[31] N.Cristianini and J.Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*, Cambridge University Press, 2000.

[32] J. T. Kwok, "Support vector mixture for classification and regression problems," in *International Conference on Pattern Recognition*, 1998, pp. Vol I: 255–258.

[33] Y. Bengio R. Collobert, S. Bengio, "A parallel mixture of svms for large scale problems," *Advances in Neural Information Processing Systems*, 2002.

[34] J. Meynet, V. Popovici, M. Sorci, and J. Thiran, "Combining svms for face class modeling," in *13th European Signal Processing Conference - EUSIPCO*, 2005.

[35] J. Meynet, V. Popovici, and JP. Thiran, "Face class modeling using mixture of svms," in *In Proceedings of International Conference on Image Analysis and recognition, ICIAR 2004, Porto, Portugal*, J. Bigun, Ed., Berlin, September 2004.

[36] Leo Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[37] D. Tao, X. Tang, X. Li, and X. Wu, "Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, no. 7, pp. 1088–1099, July 2006.

[38] Theodoros Evgeniou, Massimiliano Pontil, and André Elisseeff, "Leave one out error, stability, and generalization of voting combinations of classifiers," *Machine Learning*, vol. 55, no. 1, pp. 71–97, 2004.

[39] Darrell Whitley, "A genetic algorithm tutorial," *Statistics and Computing*, vol. 4, pp. 65–85, 1994.

[40] L. I. Kuncheva and L. C. Jain, "Designing classifier fusion systems by genetic algorithms," *IEEE-EC*, vol. 4, no. 4, pp. 327–348, November 2000.

[41] E. Bailly-Bailliere and al., "The banca database and evaluation protocol," in *4th International Conference on Audio- and Video-Based Biometric Person Authentication, Guildford, UK*, Berlin, June 2003, vol. 2688 of *Lecture Notes in Computer Science*, pp. 625–638, Springer-Verlag.

[42] Thilo-Thomas Frieß, Nello Cristianini, and Colin Campbell, "The Kernel-Adatron algorithm: a fast and simple learning procedure for Support Vector machines," in *Proc. 15th International Conf. on Machine Learning*. 1998, pp. 188–196, Morgan Kaufmann, San Francisco, CA.

[43] John Shawe-Taylor and Nello Cristianini, *Kernel Methods for Pattern Analysis*, CUP, June 2004.

[44] C.L. Blake D.J. Newman, S. Hettich and C.J. Merz, "UCI repository of machine learning databases," 1998.

[45] Chih-Chung Chang and Chih-Jen Lin, *LIBSVM: a library for support vector machines*, 2001, Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

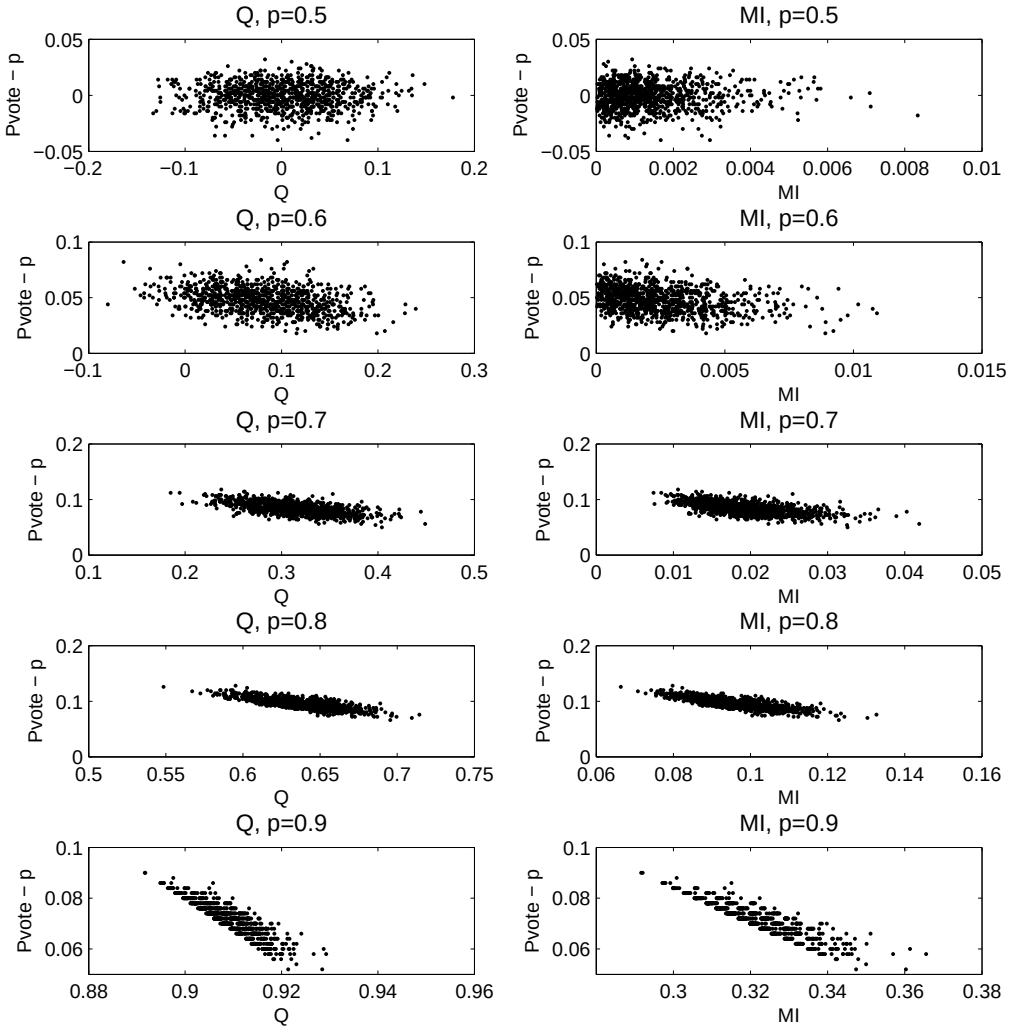


Figure 1: improvement of the ensemble w.r.t. the average individual accuracy $p = \{0.5, 0.6, 0.7, 0.8, 0.9\}$, function of 2 diversity measures.

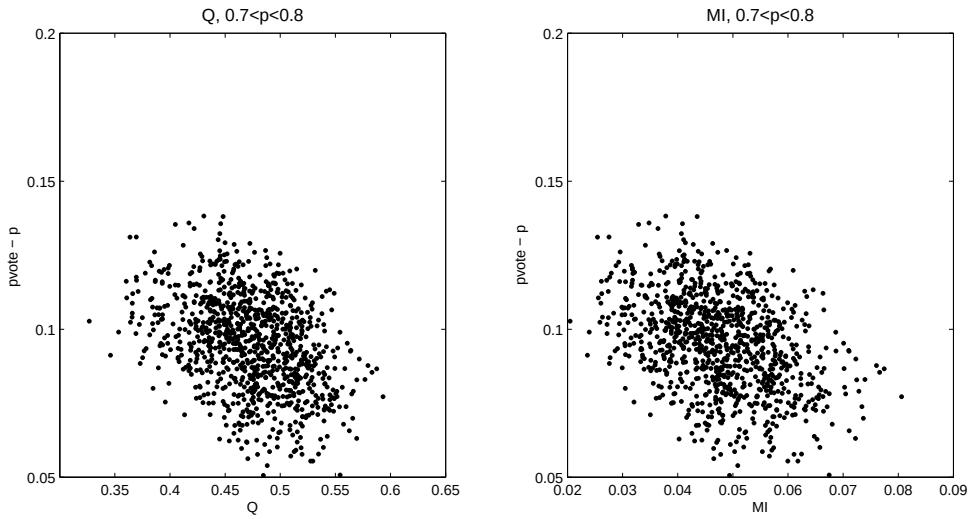


Figure 2: improvement of the ensemble w.r.t. the average individual accuracy $p \in [0.7\ 0.8]$, function of 2 diversity measures.

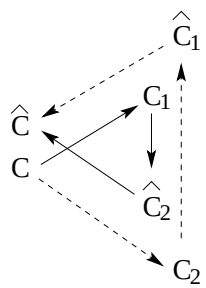


Figure 3: Coupled Markov chains for 2 classifiers trained differently from the same input data.

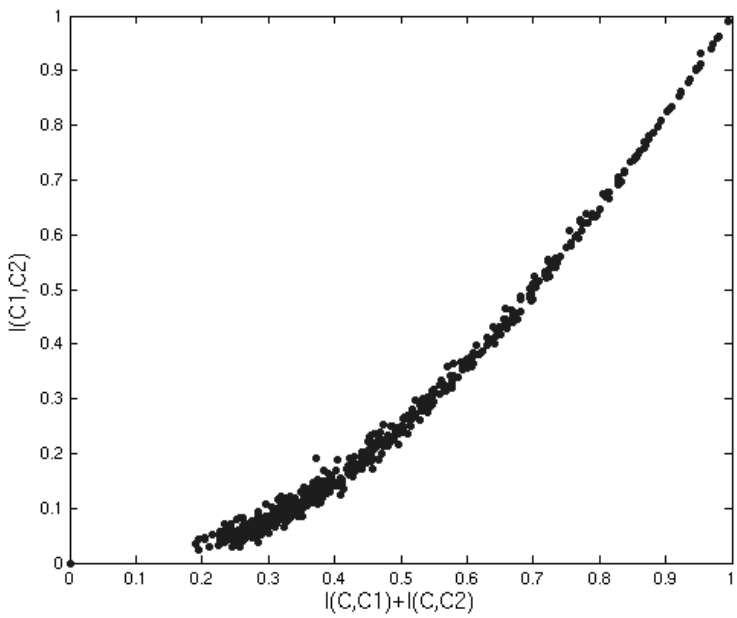


Figure 4: The similarity of 2 classifiers $I(C_1; C_2)$ function of the average individual accuracy $\frac{I(C_2;C)+I(C_1;C)}{2}$. The 2 classifiers have the same individual accuracy.

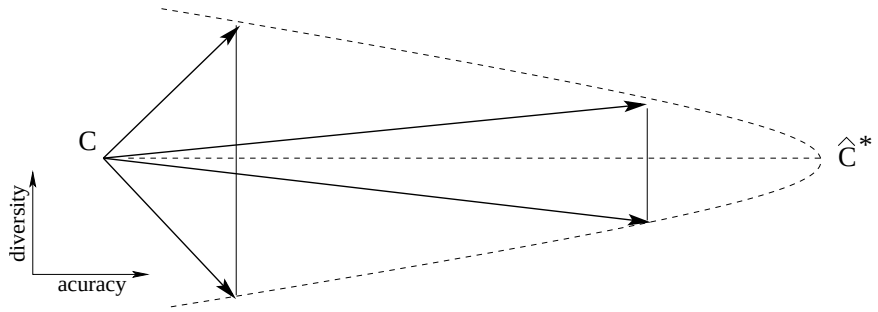


Figure 5: Graphical representation of Accuracy/Diversity dilemma.

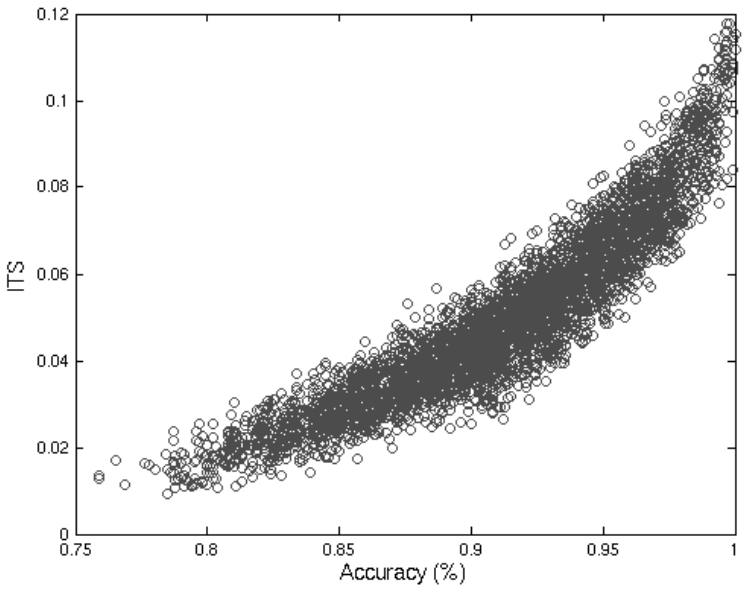
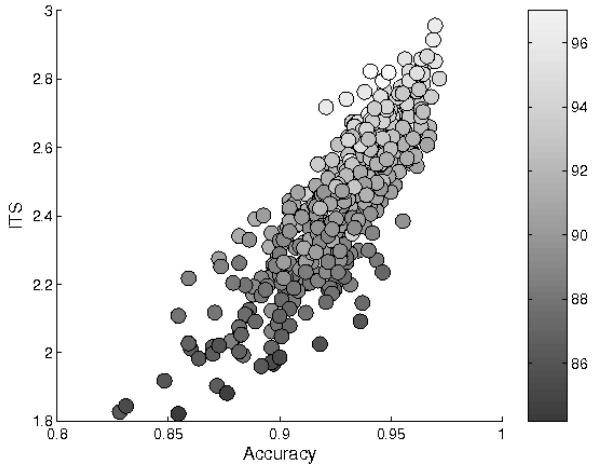
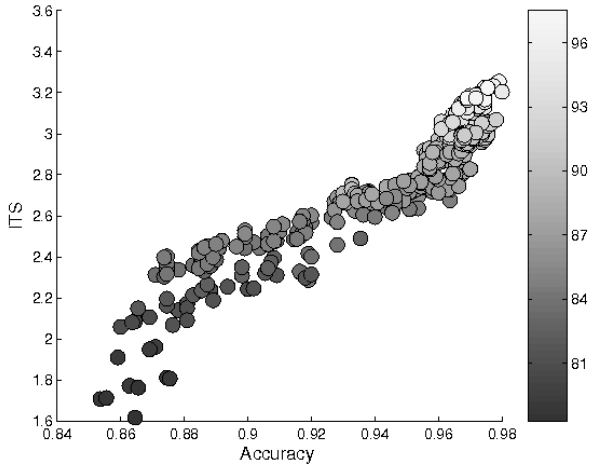


Figure 6: Score behavior with synthetic class labels



(a) 15 SVMs



(b) 5 SVMs, 5 KNN, 5 linear classifiers

Figure 7: Combination accuracy and ITS for each triplet of classifiers. (a)15 SVMs with RBF kernels and (b)5 SVMs with RBF kernels, 5 KNN classifiers and 5 linear classifiers. The color of the circle is proportional the the average accuracy of the ensembles.

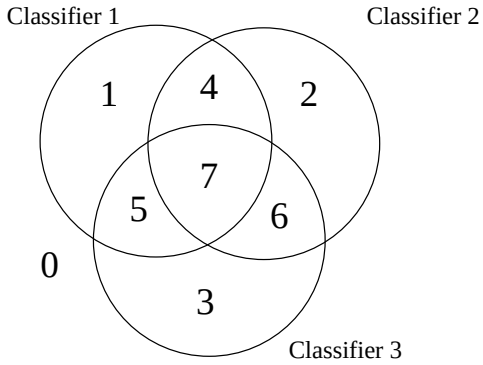


Figure 8: Chromosome encoding for GA optimization. Circles represents training samples that are used for learning each classifier. The code is 0 if none of the classifiers uses the example, 1 if only classifier 1 uses it, 4 if only classifiers 1 and 2 use it and 7 if the three classifiers have this training sample in their training set.

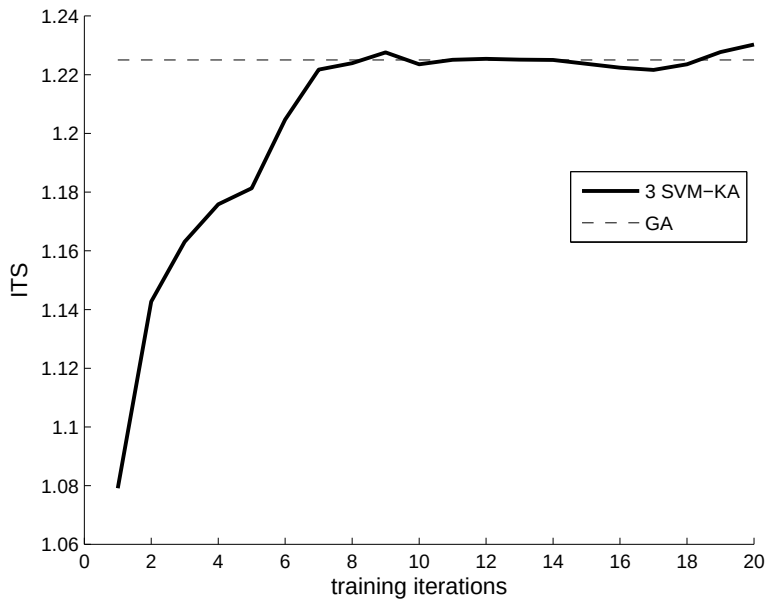


Figure 9: Comparison between ITS-Kernel Adatron (3 SVMs-KA) and Genetic Algorithms (GA)