Ensembles of Support Vector Machines for Classification Tasks with Reduced Training Sets

CLODOALDO AP. M. LIMA, ANDRÉ L. V. COELHO, FERNANDO J. VON ZUBEN
Department of Computer Engineering and Industrial Automation (DCA)
School of Eletrical and Computer Engineering (FEEC)
State University of Campinas - Unicamp
BRAZIL
{moraes,coelho,vonzuben}@dca.fee.unicamp.br http://www.dca.fee.unicamp.br/~vonzuben

Abstract: - Support vector machines (SVMs) tackle classification and regression problems by non-linearly mapping input data into high-dimensional feature spaces, wherein a linear decision surface is designed. In a previous work, we have conceived *ensembles of SVMs* (E-SVMs) in order to alleviate the performance bottlenecks incurred with the “kernel function choice” problem, that is, the necessity of choosing *a priori* the type of kernel function to realize the non-linear mapping. By this new approach, different component networks (single SVMs) with distinct kernel functions, such as polynomials or radial basis functions, may be created and properly combined into the same neural structure. E-SVMs have already been applied to a number of regression problems, yielding significantly improved generalization performance. In this paper, we extend the E-SVM methodology to also address classification tasks. Some experiments are conducted to assess E-SVM capabilities over a well-known classification problem with a reduced training set, namely, the two-intertwined spirals.

Key-Words: - Support vector machines, ensemble of support vector machines, classification problems.

1 Introduction

Support vector machines (SVMs) [1]-[4] are powerful tools for producing data classification and regression. In the classification problem, one attempts to classify points belonging to two (or more) given sets in $\Re^m$ by a linear or nonlinear separating surface. The learning process utilizes several training input data samples and the categories they belong to as the only information required for the creation of the decision surface. The produced surface is then tested on unseen (test) data. The main aim is to generate the lowest generalization error. From a statistical learning theory viewpoint, such a support vector machine will minimize the empirical error of the training data as well as the error bound for the unseen data [1].

Two key elements in any SVM implementation are the type of kernel functions and the techniques of mathematical programming. The SVM parameters are found by solving a quadratic programming (QP) problem with linear equality and inequality constraints rather than by solving a non-convex, unconstrained optimization problem. The flexibility behind the kernel functions allows the SVM to search a wide variety of hypothesis spaces. Experimentally, SVMs have outperformed other neural network (NN) configurations, e.g. in pattern recognition [3] and time series prediction [4].

By other means, in the last years, ensemble methods have shown their effectiveness in many application domains and constitute one of the main current directions in Machine Learning research. Ensembles of neural networks (ENNs) [5][6] involve the generation and linear/non-linear combination of a pool of individual NNs designed to produce redundant solution models to the same task that are complementary in terms of generalization. This is typically done through the variation of some configuration parameters and/or division of training data. The generalization capabilities of ensembles of learning machines have been interpreted in the framework of Statistical Learning Theory and in the related theory of Large Margin Classifiers.

There are several ways to use more than one classifier in a classification problem. A first "averaging" approach consists of generating multiple hypotheses from a single or multiple learning algorithms, and combining them through majority voting or alternate linear and nonlinear combinations. A "feature-oriented" approach is based on methods to build ensembles of learning machines by subdividing the input space (e.g., random subspace methods, multiple sensors fusion, feature transformation fusion). "Divide-and-conquer" methodologies isolate the regions in input space on which each classifier presents better performance, and direct new input accordingly, or subdivide a complex learning problem into a set of simpler subtasks, recombining them using suitable decoding methods. A "sequential-resampling" approach builds multiple classifier systems using bootstrap methods in order to reduce variance (bagging) or jointly bias and unbiased variance (boosting).

Previously [7], we introduced the concept of *ensembles of SVMs*, employing the combination criteria proposed in the work of Hashem [8][9] together with the selection criteria proposed by Perrone and Cooper [10]. In such work, we applied the E-SVM approach for regression problems, with simple averaging (equal weights for all SVM components) and weighted averaging (MSE-OLC) as combination criteria. There, we showed that the E-SVM approach promotes the automatic configuration and tuning of SVMs, and it yields better generalization capability when compared with the conventional best single SVM (S-SVM) approach.

In this work, we extend the concept of E-SVMs for classification tasks, employing as combination criterion the majority voting (i.e. each SVM classifier in the E-SVM contributes with the same strength for the final classification) and as selection criterion an extension of the one proposed by Perrone and Cooper [10] (to improve its generalization capabilities). The E-SVM performance is compared with the one produced by S-SVM in some experiments with the "intertwined spirals" problem [11]. In the testing samples, the desired pattern is made known in such a way as to measure the E-SVM accuracy regarding the number of misclassified points.

The paper is organized as follows. Section 2 presents the main E-SVM aspects. In Section 3, we formalize the combination problem and describe the selection criterion. Experimental results are discussed in Section 4, and Section 5 brings final remarks.

2 Ensembles of Support Vector Machines

A nice overview of SVMs may be found in [1]-[4][7], including formulations devoted to linearly and nonlinearly separable classification problems. As an extension to this conventional view, where a typical SVM archetype employs only one network configuration (as the classifier hyperplane or linear regression surface) in a high-dimensional space, we have investigated the combination of *M* networks as an ensemble of SVMs [7]. In this approach, the weighted output is given by:

$$y = \sum_{k=1}^{M} \pi_k f_k(x) = \sum_{k=1}^{M} \pi_k \left(w_k^T \phi_k(x) + b_k \right) \qquad (1)$$

where w_k, b_k are, respectively, the weights and bias of the *k*-th component network. This formulation is akin to the one proposed by Kwok [12] in the mixture of experts context, although there are two differences that are worth to be mentioned: i) in eq. (1), π_k is not a function of *x*; and ii) there exists a distinct ϕ_k for each support vector network (otherwise, eq. (1) may be replaced by a single SVM). As in [13][14], eq. (1) yields a smooth classification/regression surface in the mapped high-dimensional space governed by ϕ_k, $k = 1, \cdots, M$.

In what follows, the E-SVM training process is formalized as a QP problem, similarly to what is done in the conventional case. We address the linearly non-separable case, i.e. when training set cannot be divided without error in the $\Re^n$, which gives birth to minimizing $\frac{1}{2}\sum_k \left| w_k \right|^2 + C \sum_i \xi_i$ subject to

$$y_i \left(\sum_k \pi_{ki} \left(w_k^T \phi_k(x_i) + b_k \right) \right) \geq 1 - \xi_i$$
$$\xi_i \geq 0 \text{ for } i = 1, \cdots, N$$

where ξ_i measures the difference between y_i and the SVM output, for each of the *N* training samples, whereas *C* is a constant value indicating the contribution of each term in the optimization process. Then, the resulting QP problem may be written as:

$$\max W(\alpha) = \sum_{i}^{N} \alpha_i - \frac{1}{2} \sum_{i,j,k}^{N} y_i y_j \alpha_i \alpha_j \pi_{ki} \pi_{kj} K(x_i, x_j) \qquad (2)$$

subject to (i) $\sum_{i=1}^{N} \alpha_i y_i \pi_{ki} = 0$, for $k = 1, \cdots, M$; (ii) $0 \leq \alpha_i \leq C$, for $i = 1, \cdots, N$. For the training samples along the decision boundary, their corresponding coefficients α_i's are greater than zero, as ascertained by the Kuhn-Tucker Theorem. These samples are known as *support vectors* whose number tends to be small and proportional to the generalization error of the classifier.

3 Combination and selection criteria

In this section, we concentrate on the combination criterion of majority voting, and the selection criterion proposed by Perrone and Cooper [10].

4.1 Majority voting

Voting is the most common method used to combine classifiers. As pointed out by Ali and Pazzani [15], this strategy is motivated by the Bayesian learning theory which stipulates that, in order to maximize the predictive accuracy, instead of using just a single learning model, one ideally use all admissible models in the hypothesis space. In majority voting method the decision is made such that the label that receives more than half of the votes is taken as the final output.

4.2 Selection of the component networks

The idea of selecting nets for ensemble combination was raised by Perrone and Cooper [10], when they suggested discarding near identical nets. There are different ways in which such selection can be undertaken. Our approach has been based upon the method proposed by Perrone and Cooper, with adaptations to deal with classification problems.

For the ensemble of classifiers, as we increase the size of the component NN population, the assumption that $m_i(x) \equiv f(x) - f_i(x)$ (the deviations from the true solution) are mutually independent does not hold anymore. When this assumption fails, adding more NNs to the group incurs loss of computational resources, since this will not improve the ensemble performance. Moreover, this can be harmful in the sense that we include NNs with very bad performance, jeopardizing the execution of the whole resulting classifier.

Hence, the best choice would be to find out the optimal subset of the population over which we could calculate the majority voting. However, looking at all 2^{M-1} non-empty subsets might be unfeasible for large values of *M*. Instead, a more promising alternative is to order the population elements in consonance with the growth in the number of examples misclassified and then generate a set of classifiers by progressively combining the ordered elements. In this way, we can ascertain that the classifier is as good as the best component NN.

This process may be refined by considering the difference in the number of misclassified patterns when we pass from a ensemble of classifiers with a population of *K* elements to another with a population of *K+1* elements. From this comparison, a new component NN is only included to the group if the following inequality is satisfied:

$$(2K+1)NMP[\hat{f}_N] > 2\sum_{i \neq new} NMP[m_{new}m_i] + NMP[m_{new}] \tag{3}$$

where $NMP[\hat{f}_N]$ is the number of misclassified patterns produced by the ensemble of classifiers with *M* NNs, and $NMP[m_{new}m_i]$ is the number of misclassified patterns produced by the *i*-th (not yet tested) NN. If this criterion is not satisfied, we discard the current NN and apply the same comparative process to the next NN in the sequence.

4 Results

Here, we assess E-SVMs regarding specifically the classification problem of two intertwined spirals [11][16]. Three data sets were generated: one for training; another for the selection of the component NNs; and another to test the E-SVM performance. The "two-spirals" problem (Fig. 1), addressed by Lang and Witbrock [11], consists of two intertwined rings whose equations are given below:

Spiral #1	Spiral #2
x = 1 + (r +0.1) * cos(t)	x = 1 - (r +0.2) * cos(t)
y = 1+ (r+0.1) * sin(t)	y = 1- (r+0.2) * sin(t)

where *r* is defined in the range [0-1] and *t* is defined in the range [0-3π]. The idea is to categorize input patterns into one of two classes (50% of patterns should belong to each class).

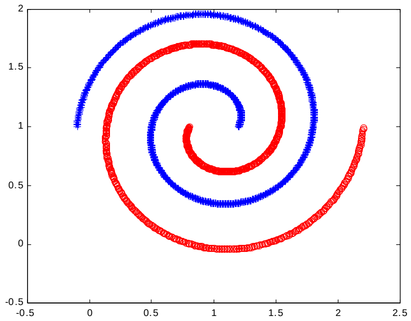

Figure 1. The "two intertwined spirals" problem

In the literature, there already exist some approaches employing SVMs for this problem, such as in the work of Suykens and Vandewalle [16]. In our case, instead of only achieving 100% of correct classification for the training data set (194 points), we have also looked for good performance of E-SVMs on data that had not been previously observed during the training phase; this is what we call here test data set. The test data points were located somewhere between the training data points, and both sets were uniformly sampled and noiseless. The selection data set was also noiseless but randomly sampled.

For assessing the E-SVM proposal, we adopted the following algorithm:

1. Generate and train (using the training data set) an E-SVM wherein each kernel type is assigned to a different NN component and the combination weights are all equal (in this process, the parameters, weights and bias, of each NN are discovered according to the problem solution).
2. Calculate the NN outputs for the selection data set.
3. Select the best NNs based on the selection criteria.
4. Calculate the weight factors (π's) using, for instance, the MSE-OLS method. In this paper, the weight factors (π's) are all equal to one.
5. Obtain the output of the E-SVM (and, consequently, the output of the NN components) using majority voting for the training and test data sets.

Table 1 - Test Results for the "two-spirals" experiment. *N* relates to the size of the training and test data sets, whereas *NSV* indicates the number of SVs found by each component NN. For each type of S-SVM (columns 3 to 10), we denote the number of training/test misclassified points, whereas bold values sign the best achieved S-SVM. The last column shows the indices of those component SVMs that integrate the final ensemble.

		SVM KERNEL TYPE								E-SVM
		LINEAR	POLY	RBF	ERBF	SIGMOID	FOURIER	SPLINE	BSPLINE	
N = 60	Train	24	23	0	0	0	3	0	**0**	0
	Test	374	367	15	0	89	111	32	**0**	0
	NSV	60	60	31	60	51	45	36	**38**	[viii]
N = 48	Train	18	18	0	0	2	4	0	**0**	0
	Test	374	366	131	4	239	272	95	**0**	0
	NSV	48	48	23	48	41	36	41	**36**	[viii]
N = 44	Train	18	17	0	**0**	0	3	0	0	0
	Test	379	369	133	**22**	201	386	99	30	0
	NSV	44	44	30	**44**	33	32	34	31	[viii]
N = 40	Train	16	17	0	0	0	4	0	**0**	0
	Test	374	366	172	11	138	360	60	**10**	0
	NSV	40	40	25	40	31	32	29	**30**	[iv,viii,i]
N = 36	Train	14	14	0	0	0	10	0	**0**	0
	Test	374	364	113	0	213	383	109	**0**	0
	NSV	36	36	23	36	30	28	27	**30**	[iv]
N = 32	Train	14	13	0	**0**	0	8	0	0	0
	Test	376	366	223	**23**	287	352	97	76	23
	NSV	32	32	20	**32**	27	29	26	28	[iv]

The classification quality was evaluated by comparing the output values of the E-SVM structures with the desired ones available in the test data set. Below, we list the various kernels adopted during the simulation experiments. A more detailed discussion on these kernels may be found elsewhere [2].

i. Linear

$$K(x,y) = x \cdot y$$

ii. Polynomial

$$K(x,y) = (x \cdot y + 1)^d$$

iii. Gaussian Radial Basis Function (for $\sigma=1$)

$$K(x,y) = \exp\left(-\frac{(x-y)^2}{2\sigma^2}\right)$$

iv. Exponential Radial Basis Function

$$K(x,y) = \exp\left(-\frac{|x-y|}{2\sigma^2}\right)$$

v. Sigmoid (for $b=1$, $c=0$)

$$K(x,y) = \tanh(b(x \cdot y) + c)$$

vi. Fourier Series

$$K(x,y) = \frac{sin(N+\frac{1}{2})(x-y)}{sin\left(\frac{1}{2}(x-y)\right)}$$

vii. Linear Splines

$$K(x,y) = 1 + xy + xy\min(x,y) - \frac{(x+y)}{2}(\min(x,y))^2$$
$$+\frac{1}{3}(\max(x,y))^3$$

viii. Bn-splines

$$K(x,y) = B_{2n+1}(x-y)$$

Several simulations were performed by varying the training data sets. Table 1 brings the most expressive results. The purpose here is to show that the E-SVM approach can outperform S-SVM in the generalization capability. In such effort, we decided to employ the uniform-majority voting scheme for combining NNs mainly for its simplicity, even though other methods (such as the weighted-majority) could produce better results. In some simulations, as we obtained classifiers with equal generalization capability, the decision was to favor those with less support vectors.

In Table 1, we emphasize the training sets with smaller sizes (given by *N*); this is due to the fact that, with large training sets, at least one of the kernels had already achieved fully-correct classification, bringing about an E-SVM with only one component. In this table, we present the number of misclassified patterns for each kernel type, both for the training (*Train*) and test (*Test*) data sets, as well as their associated number of SVs (*NSV*). As well, for each training size, the column with bold values indicates the single component SVM with best results. In these trials, the selection and test sets were composed of 384 and 944 samples, respectively.

From the results, we observe that *N*=36 can be regarded as a threshold for the E-SVM performance, since for lower values of this parameter the approach could not achieve good generalization ability. This

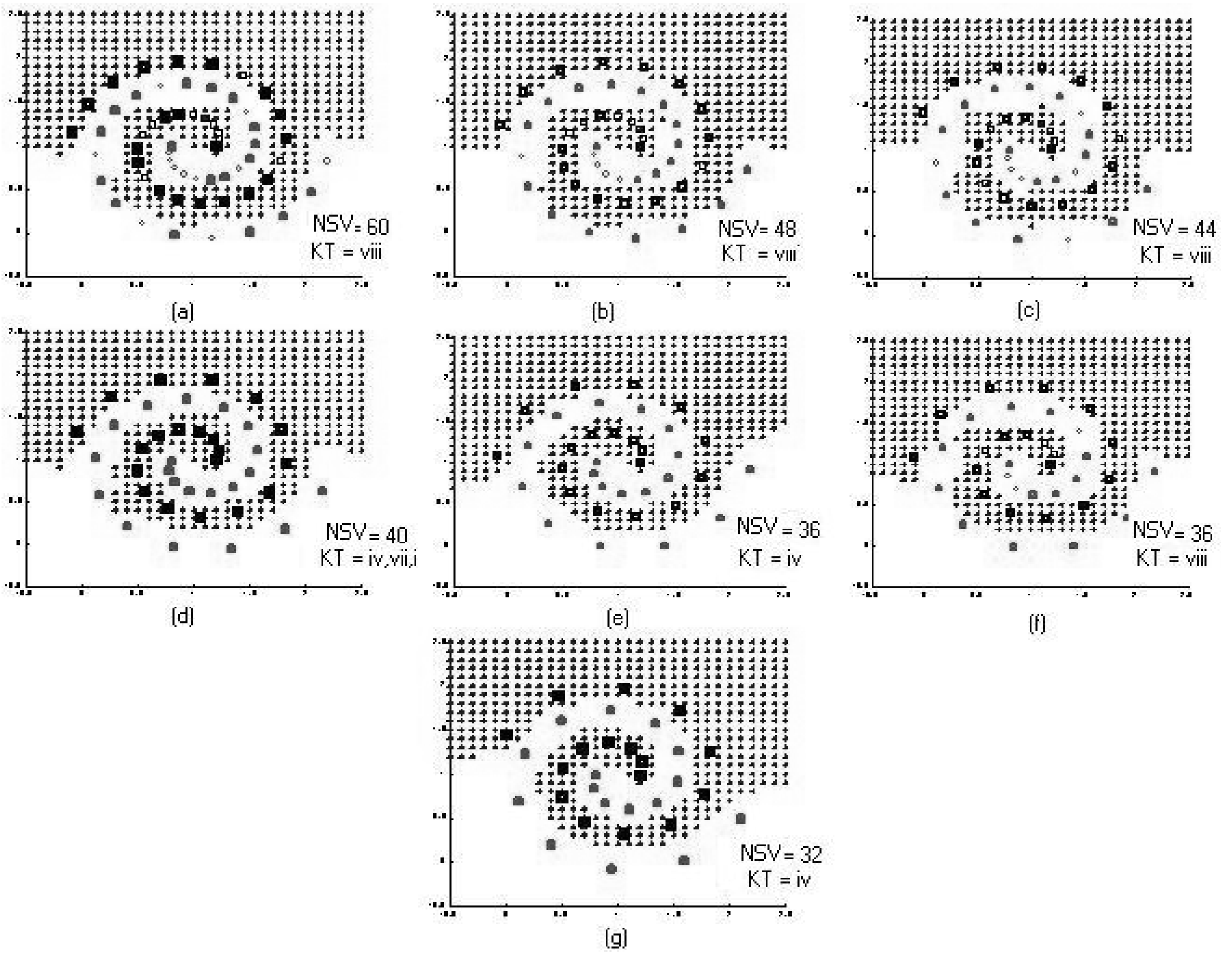

Figure 2. The "two intertwined spirals" problem as a 2D plane with x = [x_1 x_2] taken as input to the E-SVM classifier. All training data (small unfilled squares and circles) are correctly classified. Big filled squares and circles indicate the samples chosen as SVs. *NSV* and *KT* are the number of SVs and the kernel type, respectively.

owes mainly to the high-degree of correlation between the SVM components, as there were some patterns for which all NNs showed poor classification skills. Thus, the combination of such bad classifiers could not eliminate such errors. Another threshold would be for *N*=48, as for higher values than this number there is always at least one fully-successful single SVM classifier. Figures 2(a)-(g) bring the decision surfaces presented by the achieved E-SVMs, for all values of *N*.

In these figures, the "two-intertwined spirals" classification problem is represented as a 2D-plane with x = [x_1 x_2] taken as input for the E-SVM classifier. All training data (relative to two classes indicated by a black unfilled square and a red unfilled circle) are correctly classified. The big filled squares and circles indicate which training points were considered as SVs.

For *N*=36, the resulting E-SVM was formed by only one kernel type and all training samples were regarded as support vectors. Both kernel types *viii* and *iv* could achieve fully correct classification for the test data set (refer to Fig. 2d). For this training set size, a trial-and-error approach might produce the same final outcome. This would not be true for *N*=40, where all individual SVM classifiers had produced misclassifications in at least 10% of the patterns. Conversely, the resulting E-SVM was still very successful, and it was formed by kernel types *iv*, *viii*, and *i* (order by which they were inserted in). All the training samples were used as SVs, since the number of SVs for the kernel type *i* was 40. Even producing several misclassifications, the last kernel type's contribution was paramount for the classification points located at the outside boundaries (as we can see, most of the support vectors produced in all experiments reside inside the central region wherein the curves are more interlaced). We believe that with other combination criteria, such as the weighted-majority, this kernel type could be excluded from the ensemble, as the contribution of the others would be better calibrated by the weights.

Analyzing the decision surfaces at Figs. 2c and 2d, we can observe that there exist some few discrepancies between them, located, mainly, near the outside extremities (that is, out of the central region), where most of the classification errors appeared. These variations owed to the small sizes of the training data sets, as any change in the samples distribution could imply an alteration in the decision boundary. This is why we could achieve a good E-SVM for N=36, but not for N=40. Nevertheless, for N=36, many of the correctly-classified points are located very closely to the decision border, and so by changing the training data set for just a few samples, this would entail new misclassifications. That is why we consider this configuration as a threshold for E-SVM in the "two-spirals" problem.

4.1 Discussion

Some further discussion aspects come as follows:

- The E-SVM approach, in opposition to the conventional SVM method, is not prone to the size of the training data set.
- Amongst the employed kernel types, we can observe that kernels *iv*, *viii* were the most frequently chosen (at least one of them appears in the final E-SVM configurations of all experiments), which corroborates with the assumption that their generic format provides more flexibility to the classification process.
- The extra computational time required for the generation and combination of the ensemble is chiefly influenced by the number of points and the number of networks to be combined. Although more computationally expensive, the E-SVM approach, combined with the selection strategies, guide to better results, since they allow the automatic determination of those kernel types that are more appropriate to the classification problem at hand, being a more effective alternative to the common trial-and-error process.

5 Conclusion

In this paper, we have shown that the employment of ensembles of SVMs may significantly improve the classification accuracy when compared with the conventional, single SVM method. In classification problems, the training of E-SVMs leads to a QP formalization very similar to the one conceived for single SVMs.

Moreover, albeit there are some extra computational requirements underlying E-SVM simulations, they are justified in the light of the good performance achieved (see Table 1). As future work, we will continue to investigate other possibilities of automatically combining different kernel functions into the same neural structure, as well as to compare E-SVM with other ensemble approaches. Large data sets containing multiple classes will also be considered in further experiments.

6 Acknowledgements

CNPq has sponsored the authors via scholarships n° 141394/00-5 and n° 140719/99-7, and via grant n° 300910/96-7.

References:

[1] Vapnik, V. *The Nature of Stastical Learning Theory*. Springer, Verlag, 1995.

[2] Gunn, S. Support vector machine for classification and regression. Image Speech & Inteligent Systems Group, Technical Report ISIS-1-98, University of Southampton, Nov. 1998.

[3] Scholkopf, B; Sung, K; Burges, C; Girosi, F; Niyogi, P; Poggio, T; Vapnik, V. Comparing support vector machines with Gaussian kernels to radial basis function classifiers. A. I. Memo 1599, MIT, 1996.

[4] Muller, K; Smola, A; Ratsh, G; Scholkopf, B; Kohlmorgen, J.; Vapnik, V. Predicting time series with support vector machines. In *Procs. of the International Conference on Artificial Neural Networks*, 1997.

[5] Hansen, L.K.; Salamon, P. Neural Network Ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, n° 10, pp. 993-1001, 1990.

[6] Baxt, W.G. Improving the accuracy of an artificial neural network using multiple differently trained networks. *Neural Computation*, vol. 4, n° 5, pp. 135-144, 1992.

[7] Lima, C.A.M.; Coelho, A.L.V.; Zuben, F.J.V. Ensembles of support vector machines for regression problems In: 2002 INNS-IEEE International Joint Conference on Neural Networks, 2002, Hawaii. *Procs. of 2002 World Congress on Computational Intelligence (WCCI'2002)*. Piscataway, NJ: IEEE Press, vol.3. pp. 2381-2386, 2002.

[8] Hashem, S.; Schmeiser, B. Improving model accuracy using optimal linear combinations of trained neural networks. *IEEE Transactions on Neural Networks*, vol. 6, n° 3, pp. 792-794, 1995.

[9] Hashem, S. Optimal linear combinations of neural networks. *Neural Network*, vol. 10, n° 4, pp. 599-614, 1997.

[10] Perrone, M.P; Cooper, L.N. When network disagree: Ensemble method for neural networks. In Mammone, R. J., editor, *Neural Netwoks for Speech and Image processing*, pp. 126-142, Chapman-Hall, 1993.

[11] Lang, K.J; Witbrock, M.J. Learning to tell two spirals apart, In *Procs. of the 1988 Connectionist Models Summer School*, Morgan Kaufmann, 1988.

[12] Kwok, J. Tin-Yau. Support vector mixture for classification and regression problems, *Procs. of the International Conference on Pattern Recognition* (ICPR), pp. 255-258, Brisbane, August 1998.

[13] Jacobs, R; Jordan, M; Nowlan, S; Hinton, G. Adaptive mixtures of local experts. *Neural Computation*, vol. 3, n° 1, 79-87, 1991.

[14] Jordan, M; Jacobs, R. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, vol 6, n° 2, 181-214, 1994.

[15] Ali, K.M.; Pazzani, M.J. Error reduction through learning multiple descriptions. *Machine Learning*, vol. 24, pp. 173-202, 1995.

[16] Suykens, J.; Vandewalle, J. Training multilayer Perceptron classifiers based on a modified support vector method. *IEEE Transactions on Neural Networks*, vol 10, n° 4, 907-911, 1999.