Estimating Accurate Multi-class Probabilities with Support Vector Machines

Jonathan Milgram – Mohamed Cheriet – Robert Sabourin
Laboratoire d'Imagerie, de Vision et d'Intelligence Artificielle
École de Technologie Supérieure, Université du Québec
1100, rue Notre-Dame Ouest, Montréal, Canada, H3C-1K3
milgram@livia.etsmtl.ca - {mohamed.cheriet - robert.sabourin}@etsmtl.ca

Abstract – **In this paper, we propose a comparison of several post-processing methods for estimating multi-class probabilities with standard Support Vector Machines. The different approaches have been tested on a real pattern recognition problem with a large number of training samples. The best results have been obtained by using a "one against all" coupling strategy along with a softmax function optimized by minimizing the negative log-likelihood of the training data. Finally, the analysis of the error-reject tradeoff have shown that SVM allows to estimate probabilities more accurate than a classical MLP, which is indeed promising in the view of incorporated within pattern recognition system using probabilistic framework.**

I. INTRODUCTION

In 1995, Cortes and Vapnik proposed a new learning machine for two-group classification problems: the support-vector network [6], which has several interesting properties for application to pattern recognition [3]. Ten years later, this type of classifier, more known under the name of "Support Vector Machine" (SVM), has been successfully applied to several different areas [4]. Indeed, thanks to the improvement of the computing power and the development of fast learning algorithms [11][19], it is now possible to train SVM in real world applications.

Thus, although during the last years SVM has gained a lot of attention of machine learning community, multi-class SVM is still an open problem. Indeed, since standard SVM is a binary classifier, one solution to solve multi-class problems is to combine several SVMs. Then, different coupling strategies can be used. The two most classical strategies are the "one against all" and the "pairwise coupling". The first strategy constructs one SVM per class, which is trained to distinguish the examples in a single class from the examples in all remaining classes, while the second strategy construct one SVM for each pair of classes. Thus, with the first scheme, a c class problem is decomposed in c binary sub-problems and in $c(c-1)/2$ with the second scheme. In [10], the two strategies are compared on several benchmark datasets. The authors conclude that "pairwise coupling" is more suitable for practical use. Firstly, although it is necessary to train more classifiers, as each sub-problem is easier to solve, it is faster to train all the $c(c-1)/2$ SVMs of "pairwise coupling" than the c SVMs of the "one against all" strategy. Secondly, the authors have reported that "pairwise coupling" scheme offers better performance on several datasets. But, a more recent paper [20] disagrees with these results. The authors argue that the "one against all" strategy is as accurate as any other approach, assuming that SVMs are well-tuned.

On the other hand, in many pattern recognition applications the classifier only contributes a small part of the final decision. For example, in handwritten recognition it is common to use character recognizer to classify words [12] or numeral strings [16]. Then, it is essential that the output of the classifier should be a calibrated confidence measure, like posterior probability; well, standard SVMs do not provide such probabilities. However, different post-processing can be applied to map SVM outputs into posterior probabilities [9][13][18].

Thus, we propose to evaluate several methods to estimate multi-class probabilities with SVMs on a real pattern recognition problem with a large number of training samples. To compare the quality of the probabilities estimate by the different approaches, we use the Chow's rule to evaluate their error-reject tradeoff. Indeed, as it is shown in [7] this rule provides the optimal error-reject tradeoff only if the posterior probabilities of the data classes are exactly known. But, in real applications, such probabilities are affected by significant estimate errors. Thus, the better the probabilities estimate is, the better the error-reject tradeoff is.

This paper is organized as follows: Section 2 presents a simple method to map the outputs of a single SVM into posterior probabilities, while section 3 presents several approaches to estimating multi-class probabilities. Finally, section 4 summarizes our experimental results and the last section concludes with some perspectives.

II. Fitting a Sigmoid after the SVM

Although, standard SVM is a very discriminative classifier, its output values are uncalibrated. However, a simple solution is proposed in [18] to map the SVM outputs into posterior probabilities. So, given a training set of instance-label pairs $\{(x_k, y_k) : k = 1, \ldots, n\}$, where $x_k \in \mathbb{R}^d$ and $y_k \in \{1, -1\}$, the unthresholded output of an SVM is

$$f(x) = \sum_{k=1}^{n} y_k \alpha_k K(x_k, x) + b, \quad (1)$$

where the samples with non-zero Lagrange multiplier α_k are called support vectors.

Since the class-conditional between the margins are apparently exponential, the author suggests to fit an additional sigmoid to estimate probabilities:

$$\hat{P}(y = 1 \mid x) = \frac{1}{1 + \exp(A f(x) + B)}. \quad (2)$$

The parameters A and B are derived by minimizing the negative log-likelihood of the training data:

$$-\sum_{k=1}^{n} \left(t_k \log\left(\hat{P}(y_k = 1 \mid x_k)\right) + (1 - t_k) \log\left(1 - \hat{P}(y_k = 1 \mid x_k)\right) \right), \quad (3)$$

where $t_k = \frac{y_k + 1}{2}$ denotes the probability target.

Then, to solve this optimization problem, the author uses a model-trust minimization algorithm based on the Levenberg-Marquardt algorithm. But, in a recent note [13], it is shown that there are two problems in the pseudo-code provided in [18]. One is the computation of the objective value, and the other is the implementation of the optimization algorithm. Thus, the authors propose another minimization algorithm based on a simple Newton's method with backtracking line search. As we can see in sections 4.2, we tested the two algorithms on our data and noticed that the first approach induces a bias in the estimation of posterior probabilities.

III. Estimating Multi-class Probabilities

A. With the "One Against All" Strategy

In the "one against all" strategy each classifier is trained to distinguish the examples in a single class from the examples in all remaining classes. Therefore, to estimate posterior probabilities, it is possible to separately optimize each sigmoid. Then, the posterior probability of the j th class is obtain by

$$\hat{P}(\omega_j \mid x) = \frac{1}{1 + \exp(A_j f_j(x) + B_j)}, \quad (4)$$

where $f_j(x)$ denotes the output of SVM "j against all". But, nothing guarantees that the sum of all probabilities $\sum_{j=1}^{c} \hat{P}(\omega_j \mid x)$ is equal to one.

Thus, with the objective to exploit the outputs of all SVMs to estimate overall probabilities, we propose to use the softmax function, which can be regarded as a generalization of the sigmoid for multi-class case (see the section 6.9 of [2]):

$$\hat{P}(\omega_j \mid x) = \frac{\exp(\gamma f_j(x))}{\sum_{j'=1}^{c} \exp(\gamma f_{j'}(x))}. \quad (5)$$

Then, the parameter γ is derived by minimizing the negative log-likelihood of the training data, which takes form:

$$-\sum_{k=1}^{n} \sum_{j=1}^{c} \left(t_k^j \log\left(\hat{P}(\omega_j \mid x_k)\right) \right). \quad (6)$$

B. With the "Pairwise Coupling" Strategy

In the "pairwise coupling" strategy the difficulty is to combine the posterior probability of each pairwise classifier into posterior probability of multi-class classifier. With this intention, a "Resemblance Model" is proposed in [9]. Then, if prior probabilities are all the same, posterior probabilities can be estimated by

$$\hat{P}(\omega_j \mid x) = \frac{\prod_{j' \neq j} \hat{P}(\omega_j \mid x \in \omega_{j,j'})}{\sum_{j''=1}^{c} \prod_{j' \neq j''} \hat{P}(\omega_{j''} \mid x \in \omega_{j',j''})}, \quad (7)$$

where $\omega_{j,j'}$ denotes the union of classes ω_j and $\omega_{j'}$.

The main problem with this coupling scheme is related to the nonsense introduced by the value of $\hat{P}(\omega_j \mid x \in \omega_{j,j'})$ when the sample x belongs neither to class ω_j nor to class $\omega_{j'}$. A solution to overcome this problem is proposed in [15], which consists of training additional "two against all" classifiers separating classes ω_j and $\omega_{j'}$ from all remaining classes. Then, we can estimate the probabilities $\hat{P}(\omega_{j,j'} \mid x)$ which can be used to compensate the pairwise probabilities $\hat{P}(\omega_j \mid x \in \omega_{j,j'})$.

Thus, we obtain *c*-1 estimates of each probability $\hat{P}(\omega_j \mid x)$, which can be combined as follows:

$$\hat{P}(\omega_j \mid x) = \frac{\sum_{j' \neq j} \hat{P}(\omega_j \mid x \in \omega_{j,j'}) \cdot \hat{P}(\omega_{j,j'} \mid x)}{\sum_{j''=1}^{c} \sum_{j' \neq j''} \hat{P}(\omega_{j''} \mid x \in \omega_{j',j''}) \cdot \hat{P}(\omega_{j',j''} \mid x)} . \quad (8)$$

IV. Experimental Results

The training and testing of SVMs were performed with the LIBSVM software of which all the algorithms are described in [5]. We used the C-SVM with a Gaussian kernel $K(x_k, x) = \exp\left(-\frac{1}{2\sigma^2}\|x_k - x\|^2\right)$.

A. Baseline System

To evaluate the quality of the probabilities estimate with the SVMs, we chose to compare them with those estimated by a Multi Layer Perceptron (MLP). With this in mind, we used the same dataset, feature set, architecture and learning parameters, as those used in [16], where a MLP is used in a probabilistic framework to recognize handwritten numerical strings. Thus, we used the SD19 database [8], which is provided by the American National Institute of Standards and Technology (NIST). This database contains the full page binary images of 3,699 Handwriting Sample Forms (HSFs) and 814,255 segmented handprinted digit and alphabetic characters from those forms. In our experiments, we used only the images of isolated handwritten digit. The learning dataset contains the 195,000 first samples from the hsf_{0,1,2,3} subsets, while two different testing datasets are used, the hsf_7 subset, which contains 60,089 samples and the hsf_4 subset which contains 58,646 samples more difficult to classify. The feature set used, which contains 132 components, is based on a mixture of concavity and contour measures. The MLP has one hidden layer, which contains 80 neurons. The neurons of the input and the output layers are fully connected with neurons of the hidden layer. The transfer function employed is the sigmoid function and the network is trained with a sequential gradient descent with momentum applied to a sum-of-squares error function (see the sections 6.1 and 7.5 of [2]).

Furthermore, in [17] the authors have tested to replace the MLP by SVMs using "one against all" strategy and several sigmoids optimized by the original Platt algorithm. Although SVMs obtained similar accuracy on isolated digit recognition, a significant improvement is observed on numerical string recognition. Thus, we chose to use the same penalty parameter C and the kernel parameter σ, which were empirically optimized by minimizing the error rate on a validation dataset. Then, for all subsequent experiments we used C = 1000 and σ = 1.15.

B. Fitting a Sigmoid after the SVM

In a first time, we have compared the algorithm proposed by Platt in [18] and the improved version proposed by Lin *et al.* in [13]. Fig.1 shows histograms of the class-conditional densities (with bins 0.5 wide) for the SVM trained to distinguish the examples of the class ω_8 and ω_{10}. With the intention to construct the database that will be used to optimize the sigmoid, we used a cross-validation technique, in which the training set is split into four parts. Each of four SVMs are trained on permutations of three out of four parts, and the SVM outputs are evaluated on the remaining fourth. The union of all four sets of SVM outputs can form the training set of the sigmoid. Once sigmoid parameters are determined, the main SVM are re-trained on the entire training set.

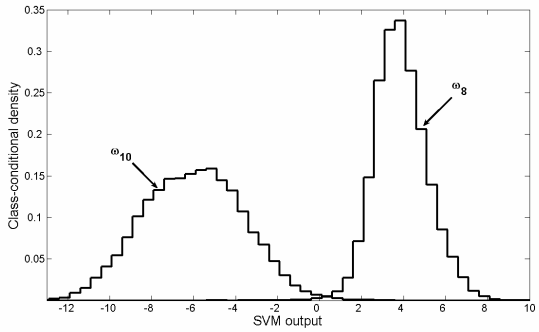

Fig.1: The histograms of the densities for the pair $\omega_{8,10}$

Thus, as we can see in Fig. 2, the sigmoid fit does not work well with the initial algorithm but is valuable with the improved version. The data points (+) in Fig. 2 are derived by using Bayes' rule on the histogram estimates of the densities in Fig.1.

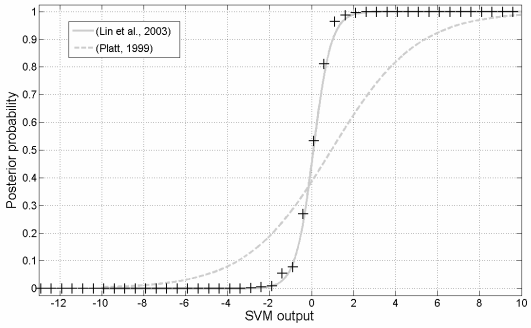

Fig. 2: The fit of the sigmoid to the data presented in Fig.1

Afterward, we have constructed the 10 SVMs of the "one against all" strategy and the 45 SVMs corresponding to the "pairwise coupling" strategy. With the intention to test the effect of the sigmoid on multi-class classification, we have used the two algorithms to optimize the sigmoid corresponding to each SVM and we have compared the results obtained on the hsf_7 dataset. Fig. 3 shows the error-reject tradeoff obtained with the "one against all" strategy and several sigmoid functions to estimate the posterior probabilities (equ. 4) and the results with the "pairwise coupling" strategy using a resemblance model (equ. 7) are shown Fig. 4. As mentioned in introduction, to evaluate the error-reject tradeoff, we use the Chow's rule, which consist to reject a sample x if

$$\max_{j=1,\ldots,c} \left(\hat{P}(\omega_j \mid x) \right) < T \,, \qquad (9)$$

where T is the reject threshold. Then, to obtain the error-reject curves, we vary the value of this threshold from 0 to 1, and for each value, we evaluate the rate of samples rejected and the error rate among the samples accepted.

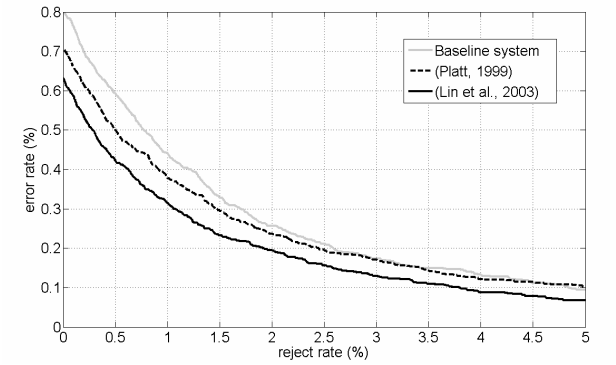

Fig. 3: Effect of the sigmoid on the "one against all" strategy

Thus, we can observe that the effect of the sigmoid with the "one against all" strategy is slightly significant, while it is catastrophic with the "pairwise coupling" strategy.

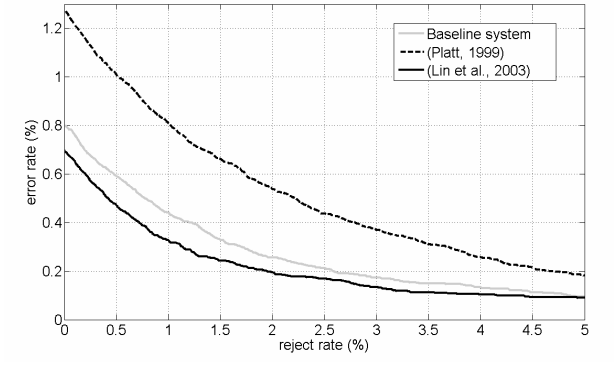

Fig. 4: Effect of the sigmoid on the "pairwise coupling" strategy

On the other hand, we have tested to optimize the sigmoid directly on the training set of the output of the SVM, without using a cross-validation technique. Although Platt has pointed out that using the same data twice can lead to biased fits, we obtained similar error-reject tradeoff. Hence, in all subsequent experiments we optimized directly the parameters of softmax and sigmoid functions on the training set.

C. Estimating Multi-class Probabilities

First remark, in opposition to the results reported in [17], we obtained a significant improvement compared to the MLP used as baseline system. Indeed, on hsf_7 we obtain an error rate without reject of 0.80 % with the MLP, while the "one against all" SVMs allows to fall this rate at 0.63 %. In the same way, on hsf_4 the error rate falls from 2.30 % to 1.89 %. Moreover, considering the error-reject tradeoff obtained on hsf_7 (see Fig. 5) and hsf_4 (see Fig. 6), a number of conclusions can be drawn.

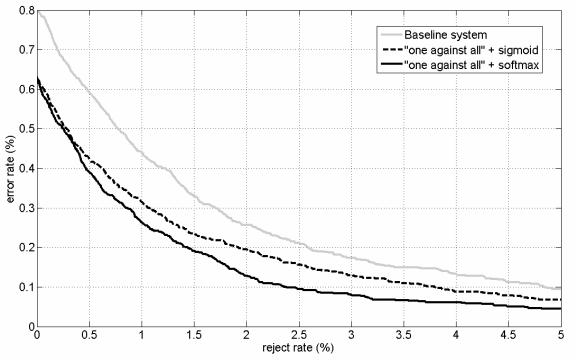

Fig. 5: Results with the "one against all" strategy on hsf_7

First, with appropriate post-processing, SVMs estimate better posterior probabilities than MLP. Second, although the error rates are similar, the error-reject tradeoff shows that it is better to use a global softmax function to estimate multi-class probabilities than several local sigmoids.

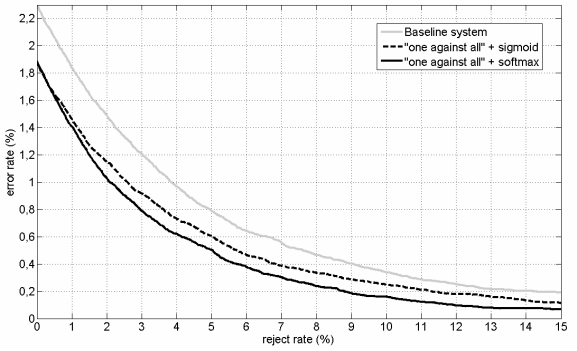

Fig. 6: Results with the "one against all" strategy on hsf_4

Furthermore, our experiments confirmed that "pairwise coupling" is faster to train and use less support vectors than "one against all", as it was observed in [10]. Indeed, the training of all the 45 SVMs of the first strategy is roughly 50 times faster than the learning of the 10 SVMs of the second strategy; furthermore, "pairwise coupling" use 5,753 SVs *vs.* 8,514 SVs for the "one against all".

The error-reject tradeoffs obtained with the "pairwise coupling" strategy are shown in Fig. 7 and Fig. 8.

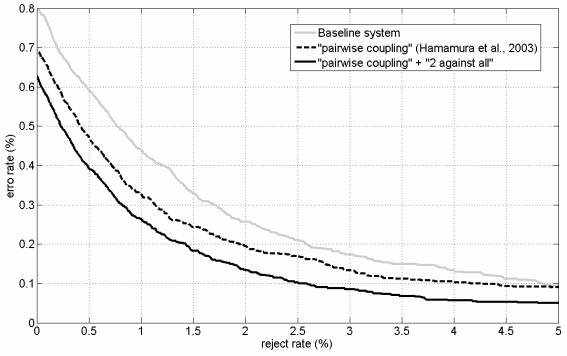

Fig. 7: Results with the "pairwise coupling" strategy on hsf_7

Thus, when we use only the pairwise SVMs with the resemblance model (equ. 7) the error-rate is higher than with the "one against all" strategy (0.70 % *vs.* 0.63 % on hsf_7 and 2.09 % *vs.* 1.89 % on hsf_4) and the error-reject tradeoff confirms that the probabilities estimate by this approach are worse.

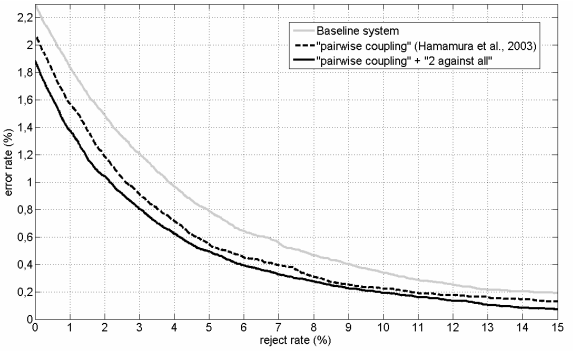

Fig. 8: Results with the "pairwise coupling" strategy on hsf_4

In addition, when we use the "two against all" SVMs to correct the probabilities of pairwise SVMs (equ. 8), we observe a significant improvement; but the accuracy obtained is not better than with the "one against all" SVMs and the softmax function, and unfortunately, the learning of the 45 "two against all" SVMs is very slow and the combination of the 90 SVMs uses a large number of support vectors (18,638 SVs).

V. Conclusions and Perspectives

In this paper, we have compared several post-processing methods for estimating multi-class probabilities with standard Support Vector Machines. For this sake, we used a real pattern recognition problem with a large number of training samples: isolated handwritten digit recognition. The main results are summarized in Table 1. Thus, as we can see, SVMs estimate better posterior probabilities than MLP. The best results have been obtained by using a "one against all" coupling strategy along with a softmax function optimized by minimizing the negative log-likelihood of the training data. Then, the reject rate necessary to fall the error rate to 0.1 % is roughly divided by two in comparison to the baseline system. Finally, these good results open the way to promising perspectives of incorporating such techniques in pattern recognition systems using probabilistic framework, as for handwritten numeral strings recognizer [16] or unconstrained handwritten words recognizer [12].

Furthermore, in future works, it will be interesting to use a more elaborated procedure to optimize the kernel parameter of each SVM [1]. In that case, it is possible that "pairwise coupling" strategy with correcting classifiers becomes more accurate than the "one against all" strategy. Indeed, this coupling strategy is more modular and should better be able to focus on local problems.

On the other hand, SVMs suffer from an important burden: the complexity necessary to make decision. But, as it is proposed in [14], it is possible to combine SVM with others classifiers in a probabilistic frameworks to speed up the decision making of the classification system.

Acknowledgment

This work was supported by research grants received from NSERC (The Natural Sciences and Engineering Research Council of Canada).

Table 1: Summary of the results obtained with the different strategies

	error rate (%) *at 0 % of reject*		**reject rate (%)** *at 0.1% of error*	
	hsf_7	**hsf_4**	**hsf_7**	**hsf_4**
baseline system (MLP)	0.80	2.30	4.83	20.48
"pairwise coupling" + resemblance model	0.70	2.09	4.16	17.29
"pairwise coupling" + "two against all"	0.63	1.89	2.37	12.51
"one against all" + sigmoid	0.63	1.88	3.79	16.01
"one against all" + softmax	0.63	1.89	2.36	11.92

REFERENCES

[1] Ayat, N.-E., Cheriet, M., Suen, C.Y. (2005) Automatic Model Selection for the Optimization of the SVM kernels. *Pattern Recognition*, accepted for publication

[2] Bishop, C.M. (1995) *Neural Networks for Pattern Recogniton*, Oxford University Press.

[3] Burges, C.J.C. (1998) A tutorial on support vector machines for pattern recognition, *Data Mining and Knowledge Discovery*, 2(2), 121-167.

[4] Byun, H., Lee, S.W. (2003) A survey on pattern recognition applications of support vector machines, *International Journal of Pattern Recognition and Artificial Intelligence*, 17(3), 459-486.

[5] Chang, C.-C., Lin, C.-J., (2001) *LIBSVM: a library for support vector machines*, Technical report, Department of computer science and information engineering, National Taiwan University.

[6] Cortes, C., Vapnik, V. (1995) Support-vector networks, *Machine Learning*, 20(3), 273-297.

[7] Fumera, G., Roli, F., Giacinto, G. (2000) Reject option with multiple thresholds, *Pattern Recognition*, 33(12), 2099-2101.

[8] Grother, P.J. (1995) *NIST Special Database 19-Handprinted Forms and Characters Database*, National Institute Standards and Technology.

[9] Hamamura, T., Mizutani, H., Irie, B. (2003) A multiclass classification method based on multiple pairwise classifiers. *International Conference on Document Analysis and Recognition*, 809-813.

[10] Hsu, C.-W., Lin, C.-J. (2002) A comparison of methods for multi-class support vector machines, *IEEE transactions on Neural Networks,* 13(2), 415-425.

[11] Joachims, T. (1999) Making large-scale SVM learning practical, *Advances in kernel methods: support vector learning*, MIT Press, 169-184.

[12] Koerich, A.L., Sabourin, R., Suen, C.Y. (2005) Recognition and verification of unconstrained handwritten words, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* in press.

[13] Lin, H.-T., Lin, C.-J., Weng, R.C. (2003) *A note on Platt's probabilistic outputs for support vector machines*. Technical report, Department of computer science and information engineering, National Taiwan University.

[14] Milgram, J., Cheriet, M., Sabourin, R. (2004) Speeding up the decision making of support vector classifiers, *International Workshop on Frontiers in Handwriting Recognition*, 57-62.

[15] Moreira, M., Mayoraz, E. (1998) Improved pairwise coupling classification with correcting classifiers, *European Conference on Machine Learning,* 160-171.

[16] Oliveira, L.S., Sabourin, R., Bortolozzi, F., Suen, C.Y. (2002) Automatic recognition of handwritten numerical strings: a recognition and verification strategy, *IEEE transactions on Pattern Analysis and Machine Intelligence*, 24(11), 1438-1454.

[17] Oliveira, L.S., Sabourin, R. (2004) Support vector machines for handwritten numerical string recognition*, International Workshop on Frontiers in Handwriting Recognition*, 39-44.

[18] Platt, J.C. (1999) Probabilities for SV Machines, *Advances in Large Margin Classifiers*, MIT Press, 61-74.

[19] Platt, J.C. (1999) Fast training of support vector machines using sequential minimal optimization, *Advances in kernel methods: support vector learning*, MIT Press, 185-208.

[20] Rifkin, R., Klautau, A. (2004) In defence of one-vs-all classification, *Journal of Machine Learning Research*, 5(Jan), 101-141.