Electronic Letters on Computer Vision and Image Analysis 5(2):1-15, 2005

Combining Model-based and Discriminative Approaches in a Modular Two-stage Classification System: Application to Isolated Handwritten Digit Recognition

Jonathan Milgram, Robert Sabourin and Mohamed Cheriet

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

Received 16 July 2004; accepted 10 March 2005

Abstract

The motivation of this work is based on two key observations. First, the classification algorithms can be separated into two main categories: discriminative and model-based approaches. Second, two types of patterns can generate problems: ambiguous patterns and outliers. While, the first approach tries to minimize the first type of error, but cannot deal effectively with outliers, the second approach, which is based on the development of a model for each class, make the outlier detection possible, but are not sufficiently discriminant. Thus, we propose to combine these two different approaches in a modular two-stage classification system embedded in a probabilistic framework. In the first stage we estimate the posterior probabilities with a model-based approach and we re-estimate only the highest probabilities with appropriate Support Vector Classifiers (SVC) in the second stage. Another advantage of this combination is to reduce the principal burden of SVC, the processing time necessary to make a decision and to open the way to use SVC in classification problem with a large number of classes. Finally, the first experiments on the benchmark database MNIST have shown that our dynamic classification process allows to maintain the accuracy of SVCs, while decreasing complexity by a factor 8.7 and making the outlier rejection available.

Key Words: Classifier Combination, Support Vector Classifier, Model-based Approach, Outlier Detection, Errror-Reject Tradeoff, Classifying Cost, Isolated Handwritten Digit Recognition.

1 Introduction

The principal objective of a pattern recognition system is to minimize classification errors. However, another important factor is the capability to estimate a confidence measure in the decision made by the system. Indeed, this type of measure is essential to be able to make no decision when the result of classification is uncertain. From this point of view, it is necessary to distinguish two categories of problematic patterns. The first one relates to ambiguous data which may cause confusion between several classes and the second category consists of data not belonging to any class: the outliers.

Furthermore, most classification algorithms can be divided into two main categories denoted as discriminative and model-based approaches. The former tries to split the feature space into several regions

Correspondence to: milgram@livia.etsmtl.ca

Recommended for acceptance by J.M. Ogier, T. Paquet, G. Sanchez
ELCVIA ISSN: 1577-5097
Published by Computer Vision Center / Universitat Autonoma de Barcelona, Barcelona, Spain

by decision surfaces, whereas the latter is based on the development of a model for each class along with a similarity measure between each of these models and the unknown pattern (see Figure 1). Different terms are used in literature to refer it, generative method [2], density model [18], approach by modeling [23] or model-based classifier [21].

(a) discriminative (b) model-based

Figure 1: Two types of classification approaches

Thus, as is shown in [18], the discriminative classifiers are more accurate in classifying ambiguous data, but not suitable for outlier detection, whereas model-based approaches are able to reject outliers but not effective in classifying ambiguous patterns. Considering this, the authors propose to hybridize the two types of approaches internally or to combine them externally. In a more recent paper [19], the same authors have tested an internal fusion of the two approaches. Their method improves the accuracy of the model-based approach by using discriminative learning. However, even though their classifier is more accurate, it is not as accurate as the best discriminative approaches such as support vector classifiers.

Hence, in this paper, we propose to combine a model-based approach with support vector classifier (SVC). This classification system should give high accuracy and strong outlier resistance. The idea is to develop a two-stage classification system. At the first stage, a model-based approach can directly classify patterns that are recognized with high confidence, reject outliers or insulate those classes in conflict. Then, if conflict is detected, the appropriate SVCs will make better decision at the second stage. Another advantage of this combination is to reduce the main burden of SVC: the processing time necessary to make a decision.

Thus, the proposed system is a multiple classifiers combination, which is a widely studied domain in classification [4][9][15][14][15]. Although a number of similar ideas related to two-stage classification to treat ambiguity were introduced in recent papers [1][5][8][21][22][24], our classification system remains different and original. Indeed, the idea of multiple classifiers combination to treat ambiguity is presented in [8], but the proposed system combine only different model-based classifiers and is only tested on 2D artificial data. On the other hand, the combination of model-based and discriminative approaches is proposed in [5][8][21][22] but their motivations are different. In [5], the model-based approach is used in a second stage to slightly improve the rejection capability of the MLP used at the first stage. In [21], the authors use only a few MLPs to improve the accuracy of the first classifier, which used a reduced number of prototypes. In [22], the authors use fuzzy decision trees to improve significantly a first system based on fuzzy clustering, but their combination is not as accurate as SVC. Concerning the use of SVCs in a second stage of classification to improve the accuracy two different approaches are presented in [1][24]. In [1], the authors take into account the problem of complexity of SVCs, but in the first-stage they use MLP which is another discriminative approach. Furthermore, their system does not make decisions at the first-stage and always uses one SVC, and never more than one, which limits the performance of the system. In [24], the authors propose several elaborate strategies for detecting conflicts. However, they do not take into account the problem of complexity. Indeed, the first-stage uses a complex ensemble of classifiers. Moreover, the results of their two-stage system are not compared to a full SVCs system. Thus, if the use of SVCs can improve the accuracy of the ensemble of classifier used in the first stage, would it then be better to use a full SVCs system?

Moreover, we embed our system within a probabilistic framework, because as mentioned in [20]: “The output of a classifier should be a calibrated posterior probability to enable post-processing”. Indeed, this type of confidence measure is essential in many application, when the classifier only contributes a small part of the final decision or if it is preferable to make no decision when the result of classification is uncertain. So, in the first stage, we estimate the probabilities with a model-based approach and re-estimate only the highest

probabilities with appropriate SVCs in the second stage. Thus, to compare the quality of the probabilities estimate by the different methods, we use the Chow's rule to evaluate their error-reject tradeoff. Indeed, as it is shown in [6], 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. In consequence, the better the probabilities estimate is, the better the error-reject tradeoff is.

This paper is organized as follows: Section 2 presents the model-based approach, while the section 3 presents its combination with discriminative approach. Section 4 summarizes our experimental results and the last section concludes with some perspectives.

2 Model-based approach

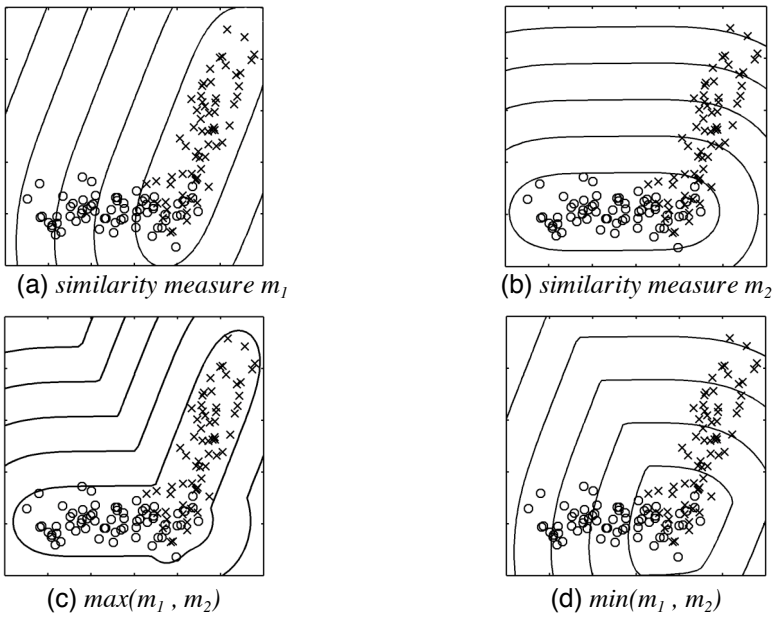
One of the main advantages of this type of approach is the modularity. Indeed the training process is computationally cheap because the model of each class is learned independently. Thus, it is well scalable to large category problems such as Chinese character recognition [12]. On the other hand, this also facilitates the increment/decrement of categories without re-training all categories.

2.1 Characterization of the pattern recognition problem

Although this type of approach is not very discriminant, it can be used to characterize the problem of pattern recognition. Thus, three cases can be considered during testing:

- A single similarity measure is significant. The pattern can be directly classified.
- Several similarity measures are comparable. It is an ambiguous pattern and it is better to use a discriminative approach to make decision.
- All similarity measures are unsignificant. The pattern can be considered as an outlier.

An artificial toy example with only 2 features is presented in Figure 2 to show how this type of classifier is able to detect outliers and ambiguous patterns. The ideal similarity measure of each class is represented by level line in (a) and (b). Thus, we can see that it is possible to use it to make new interesting measures. Indeed, in this simple example with two classes, the maximum of the two similarity measures shown in (c) can be used to detect outlier, whereas the minimum shown in (d) can be used to detect conflict.



(a) *similarity measure* m_1 (b) *similarity measure* m_2

(c) $max(m_1 , m_2)$ (d) $min(m_1 , m_2)$

Figure 2: Use of model-based approach to detect outliers (c) and ambiguous patterns (d)

2.2 Modeling data with hyperplanes

To start, we make the assumption that each class is composed of a single cluster in the feature space and that data distributions are Gaussian in nature. Then, a classical Bayesian approach consists to use parametric methods to model each class statistically based on data means and covariance, which can be used in quadratic discriminant functions to make decision. But, it is shown in [12] that quadratic discriminant functions are very sensitive to the estimation error of the covariance matrix. Thus, in many applications with a large number of features, it is preferable to regularize the covariance matrix. Another improvement proposed in [12] is to neglect the nondominant eigenvectors, because the estimation errors in the nondominant eigenvectors are much greater than those of the dominant eigenvectors.

With the same idea, it is possible to model each class ω_j with a hyperplane defined by the mean vector μ_j, and the matrix Ψ_j which contains the k first eigenvectors ϕ_j^i extracted from the covariance matrix Σ_j.

Then, the measure of the similarity (or dissimilarity) used is the projection distance on the hyperplane:

$$d_j(x) = \left\| x - f_j(x) \right\|^2 \quad (1)$$

Thus, given a data point x of the feature space, the membership to the class ω_j can be evaluated by the square of the Euclidean distance d_j from the point x to its projection on the hyperplane:

$$f_j(x) = \left(\left(x - \mu_j \right) \Psi_j \right) \Psi_j^T + \mu_j \quad (2)$$

Finally, it is possible to reformulate the projection distance to reduce the complexity of calculation:

$$d_j(x) = \left\| x - \mu_j \right\|^2 - \sum_{i=1}^{k} \left\{ (x - \mu_j) \phi_j^{\,i} \right\}^2 \quad (3)$$

The Figure 3 shows a simple example of projection distance, where each class is modeled by its principal axis ($k = 1$) and the data point x is projected on $f_1(x)$ and $f_2(x)$.

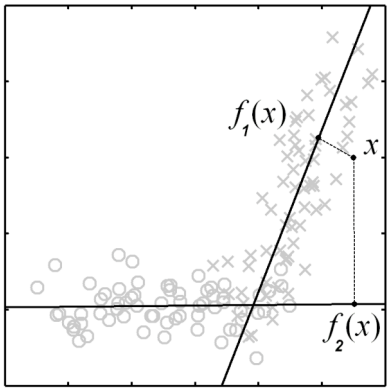


Figure 3: 2D example of projection distance

Although it would be preferable to bound the hyperplanes with the intention to close the decision surface, when the feature space is very large and bounded, it seems that the probability that a pattern is far away from the training data and close to the hyperplane is very low. Thus, it is shown in [11] that the accuracy obtained by the projection distance method is very close to the accuracy of a three layer autoassociative neural networks with sigmoid function on the hidden layer, which guaranties to close the decision surface [7].

Furthermore, this method requires the optimization of only one parameter: the number k of eigenvectors used. But, as we can see in section 4.1, this parameter is crucial for classification. Thus, if k is too small, the models are not precise so we loose too much information. In fact, while $k = 0$, each class is model by a simple prototype that is the mean vector μ_j of training data. On the other hand, if the value of k is too large, the models are not discriminative. At worst, if $k = d$, where d is the dimension of the input pattern, the hyperplane embeds all the points of the feature space. Hence, for all point x, the projection distance will be null.

2.3 Estimate posterior probability

Thus, if the processed pattern is not an outlier, we can estimate posterior probability in the first stage of our system. Then, if we suppose that the distribution of the projection distances between the margins is exponential, we can use the softmax function to map projection distance to posterior probability:

$$\hat{P}_f(\omega_j \mid x) = \frac{\exp(-\alpha d_j(x))}{\sum_{j'=1}^{c} \exp(-\alpha d_{j'}(x))} \quad (4)$$

3 Combination with discriminative approach

Thereafter, if a pattern is considered as ambiguous in the first stage of our system, we use appropriate discriminative experts to re-estimate only the most significant posterior probabilities in the second stage.

3.1 Conflict detection

The first step is to detect the patterns that may cause confusion. In [1] and [21], the authors consider that conflict involves only two classes and they use appropriate experts, to reprocess all samples in [1], or just the samples rejected by the first classifier in [21]. However, we consider that conflict may involve more than two classes. Hence, it is preferable to use a dynamic number of classes in conflict. With this intention, we determine the list of p classes $\{\omega_{\ell(1)},\ldots,\omega_{\ell(p)}\}$ of which the posterior probabilities estimated in the first stage are higher than a threshold ε. Thus, $\ell(j)$ is the index of the j th class that verifies:

$$\hat{P}_f(\omega_{\ell(j)} \mid x) > \varepsilon \quad (5)$$

Then, if p is superior to one, we use in the second stage the appropriate discriminative expert to re-estimate the posterior probabilities of the p classes. Finally, this parameter controls the tolerance level of the first stage of classification and consequently the classifying cost. Indeed, the smaller the threshold ε is, the larger the number p will tend to be. If ε is too large, then we never use the second stage of classification. But, if ε is too small, then the system uses unnecessary discriminative classifiers.

3.2 Use of Support Vector Classifiers

A recent benchmarking of state-of-the-art techniques for handwritten digit recognition [19] has shown that Support Vector Classifier (SVC) gives higher accuracy than classical neural classifiers like Multi Layer Perceptron (MLP) or Radial Basis Function (RBF) networks. However, thanks to the improvement of the computing power and the development of new learning algorithms, it is now possible to train SVC in real world applications. Thus, we choose to use SVC in the second stage of our system.

Also, if an SVC can possibly make good decisions, these output values are uncalibrated. But, a simple solution is proposed in [20] to map the SVC outputs into posterior probabilities. 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 SVC is

$$f(x) = \sum_{k=1}^{n} y_k \alpha_k K(x_k, x) + \beta \ , \quad (6)$$

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

Since the class-conditional between the margins are apparently exponential the authors suggest to fit an additional sigmoid function (equation 7) to estimate probabilities.

$$\hat{P}(y = 1 \mid x) = \frac{1}{1 + \exp(a f(x) + b)} \quad (7)$$

The parameter *a* and *b* are derived by minimizing the negative log likelihood of the training data, which is a cross-entropy function:

$$-\sum_{k=1}^{n}\left(t_k \log\left(\hat{P}(y_k = 1 \mid x_k)\right) + (1-t_k)\log\left(1-\hat{P}(y_k = 1 \mid x_k)\right)\right), \quad (8)$$

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 [17] it is shown that there are two problems in the pseudo-code provided in [20]. One is the calculation of the objective value, and the other is the implementation of the optimization algorithm. Therefore, the authors propose another minimization algorithm more reliable, based on a simple Newton's method with backtracking line search. Thus, we use this second algorithm to fit additional sigmoid function and estimate posterior probabilities.

Furthermore, SVC is a binary classifier, so it is necessary to combine several SVCs to solve a multi-class problem. A most classical method is the "one against all" strategy in which one SVC per class is constructed. Each classifier is trained to distinguish the examples in a single class from the examples in all remaining classes. Although this strategy is very accurate, it seems better to use in the second stage of our system a "pairwise coupling" approach, which consists to construct a classifier for each pair of classes. Indeed, this strategy is more modular and as reported in [3], although we have to train as many as c(c-1)/2 classifiers, as each problem is easier, the total training time of "pairwise coupling" may not be more than that of the "one against all" method. Furthermore, if we use "one against all" SVCs in the second stage, we are obliged to calculate the distances of a large number of SVs belonging to the implausible classes, which increases the classifying cost. Thus, we choose to use a "pairwise coupling" approach and we apply the "Resemblance Model" proposed in [10] to combine posterior probability of each pairwise classifier into posterior probability of multi-class classifier. Then, since prior probabilities are all the same, posterior probabilities can be estimated by

$$\hat{P}(\omega_j \mid x) = \frac{\prod_{j' \neq j} \hat{P}(\omega_j \mid x \in \omega_{j,j'})}{\sum_{j''=1}^{c} \prod_{j' \neq j''} \hat{P}(\omega_{j''} \mid x \in \omega_{j'',j'})}, \quad (9)$$

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

3.3 Re-estimate posterior probabilities

Finally, as we can see in Figure 4, we use only $p(p\text{-}1)/2$ SVCs to re-estimate only the most significant posterior probabilities. In consequence, the final probabilities are not homogeneous, since they can be estimated by different approaches. However, it is not an important drawback. Indeed, when *p* is superior to one, the first stage estimates only the smallest probabilities, which are negligible, and in this case the second stage estimates all the remaining probabilities. These *p* significant probabilities are obtained by

$$\hat{P}_s(\omega_{\ell(j)} \mid x) = \frac{\prod_{j''=1, j'' \neq j}^{p} \hat{P}_s(\omega_{\ell(j)} \mid x \in \omega_{\ell(j),\ell(j'')})}{\sum_{j'=1}^{p} \prod_{j''=1, j'' \neq j'}^{p} \hat{P}_s(\omega_{\ell(j')} \mid x \in \omega_{\ell(j'),\ell(j'')})} \times \left(1 - \sum_{j'=p+1}^{c} \hat{P}_f(\omega_{\ell(j')} \mid x)\right), \quad (10)$$

where the first term is related to the second stage, while the second term is related to the first stage. The objective of this second term is to maintain the sum of all the probabilities equal to one.

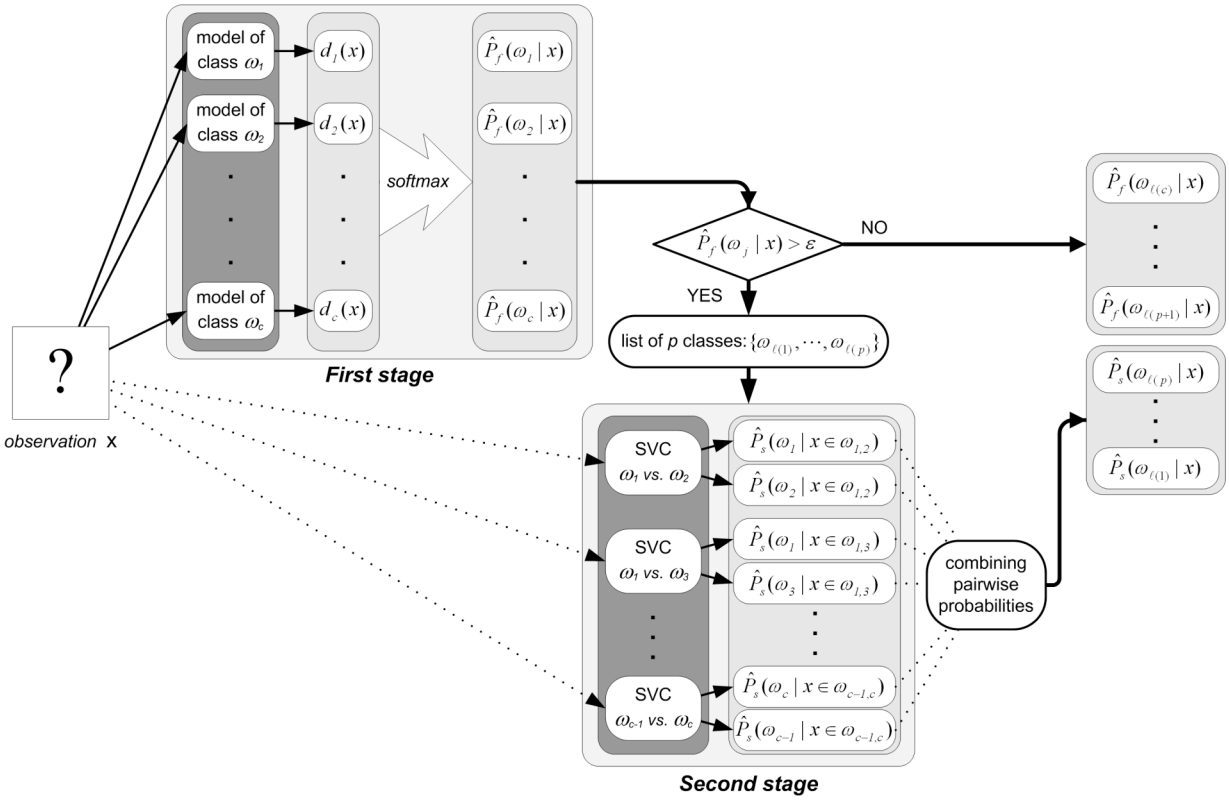


Figure 4: Overview of our two-stage classification system

4 Experimental results

To evaluate our method, we chose a classical pattern recognition problem: isolated handwritten digit recognition. Thus, in our experiments, we used a well-known benchmark database. The MNIST (Modified NIST) dataset[1] was extracted from the NIST special database SD3 and SD7. The original binary images were normalized into 20×20 grey-scale images with aspect ratio preserved and the normalized images were centered by center of mass in 28×28 images. Some sample images of this database are shown in Figure 5.

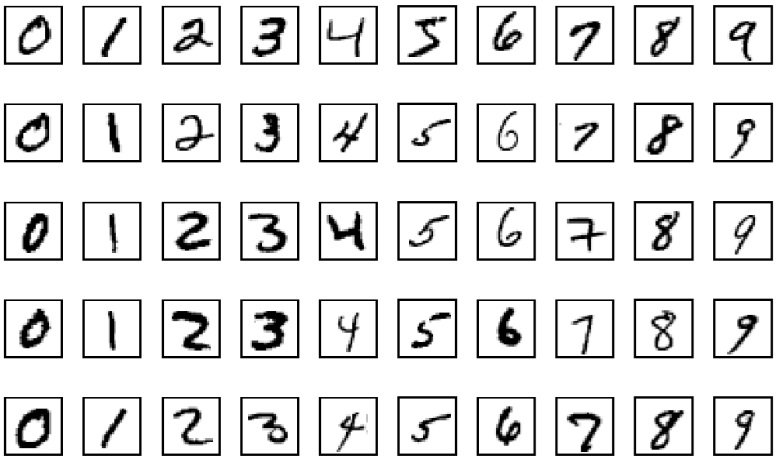


Figure 5: Sample images of MNIST dataset

[1] available at http://yann.lecun.com/exdb/mnist/

The learning dataset contains 60,000 samples and 10,000 others are used for testing. Moreover, we have divided the learning database into two subsets. The first 50,000 samples have been used for training and the next 10,000 for validation. Finally, the number of samples per class for each subset is reported in the Table 1.

	ω_1	ω_2	ω_3	ω_4	ω_5	ω_6	ω_7	ω_8	ω_9	ω_{10}
training	4932	5678	4968	5101	4859	4506	4951	5175	4842	4988
validation	991	1064	990	1030	983	915	967	1090	1009	961
test	980	1135	1032	1010	982	892	958	1028	974	1009

Table 1: Number of samples per class in the three subset of the MNIST database

Several papers dealt with the MNIST database. The best result mentioned in the original paper [16] is obtained by the convolutional neural network LeNet-5 (0.95% of error rate on the test dataset). More recently, a benchmarking of state-of-the-art techniques [19] has shown that SVC with 8-direction gradient features gives the highest accuracy reported at this day (0.42% of error rate on the test dataset). A short summary of results obtained in [19] is reported in Table 2.

	k-NN	LVQ	RBF	MLP	SVC
without feature extraction	3.66 %	2.79 %	2.53 %	1.91 %	1.41 %
with feature extraction	0.97 %	1.05 %	0.69 %	0.60 %	0.42 %

Table 2: Error rate on the MNIST test dataset reported in [19] with state-of-the-art techniques

Although, feature extraction allows a better accuracy, we chose to use the original database to make the proof of concept of our modular two-stage combination.

4.1 Model-based approach

Initially, we must fix the dimensionality of the hyperplane models. For this purpose, we chose to use the same value of k for all hyperplanes, because it is not trivial to find the optimal values of each hyperplane. Furthermore, we think that it is not a problem to use a suboptimal solution because the second stage is here to refine classification. Finally, we use the validation dataset to find the better value of k and we can see in Figure 6 that this parameter strongly influences the accuracy of the classification. Consequently, we use $k = 25$ and we obtain an error rate of 4.09 % on the test dataset. For comparison, we obtain an error rate of 7.06 % with the quadratic discriminant function. Indeed, because the data have many singular directions, we are forced to add an important constant ($\lambda = 0.4$).

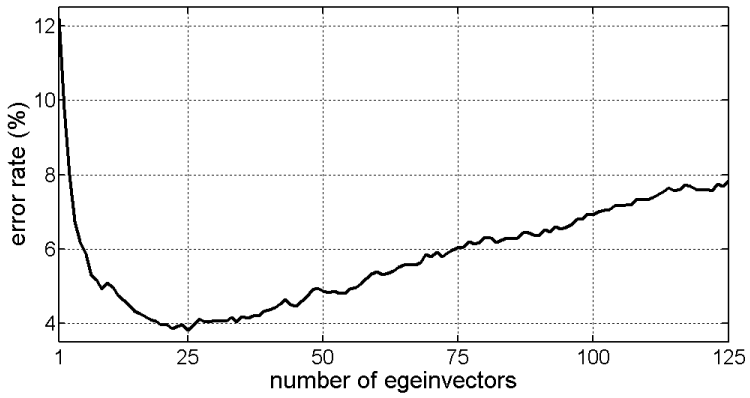


Figure 6: Effect of the dimensionality of the hyperplane models

Thereafter, the α parameter of the softmax function (equation 4) is chosen to minimize the cross entropy error on the validation dataset. We obtain the best result with $\alpha = 5.6$. We can notice in Figure 7 that the use of the softmax function improves significantly the error-reject tradeoff of the model-based and that half of the examples with the highest confidence levels are correctly classified.

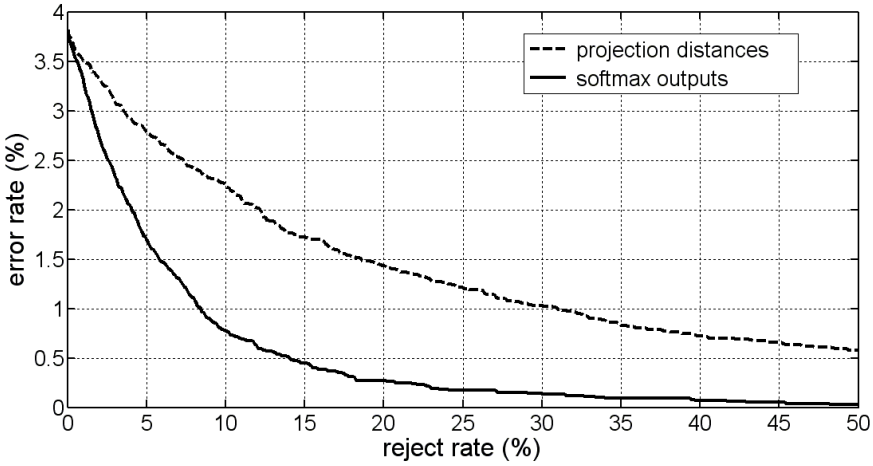


Figure 7: Error-reject tradeoff of the model-based approach on the validation dataset

Finally, even though the reliability of the proposed model-based approach is not very high, it should be able to characterize the pattern recognition problem. Indeed, as we can see below, the three cases considered in section 2.1 can be observed in real application like isolated digit recognition:

- A single projection distance is very small. The pattern can be considered as **unambiguous** and the posterior probabilities can be directly estimated (see Figure 8).
- Several projection distances are small. The pattern can be considered as **ambiguous** and it is preferable to re-estimate the posterior probabilities with the discriminative approach (see Figure 13).
- All projection distances are high. The pattern can be considered as **outlier** and can be rejected (see Figure 9).

class:	ω_1	ω_2	ω_3	ω_4	ω_5	ω_6	ω_7	ω_8	ω_9	ω_{10}
$d_j(x)$:	5.0701	6.4723	5.5140	4.9112	5.2510	5.2979	**1.8797**	6.2682	6.0157	6.0526
$\hat{P}_f(\omega_j \mid x)$:	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**1.0000**	0.0000	0.0000	0.0000

Figure 8: Example of unambiguous pattern (8,400th sample of the test dataset)

$f_j(x)$:										
class:	ω_1	ω_2	ω_3	ω_4	ω_5	ω_6	ω_7	ω_8	ω_9	ω_{10}
$d_j(x)$:	5.0830	6.7728	5.5134	5.9115	5.2804	5.6850	5.6278	5.4458	6.0277	5.5054

Figure 9: Example of outlier (generated with the 12th and 13th sample of the test dataset)

4.2 Support Vector Classifiers

The training and testing of all SVCs are performed with the LIBSVM software of which all the algorithms are described in [3]. We use the C-SVC with a Gaussian kernel $K(x_k,x)=\exp(-\gamma\|x_k-x\|^2)$. The penalty parameter C and the kernel parameter γ are empirically optimized by trial and error. Then, we have chosen parameters that minimize the error rate on the validation dataset. Finally, we used C = 10 and γ = 0.0185 and we obtain an error rate of 1.48 % on the test dataset, which is comparable with those reported in [19] when no discriminative features are extracted. Moreover, as we can see in Figure 10 the SVCs estimate better probabilities than model-based approach.

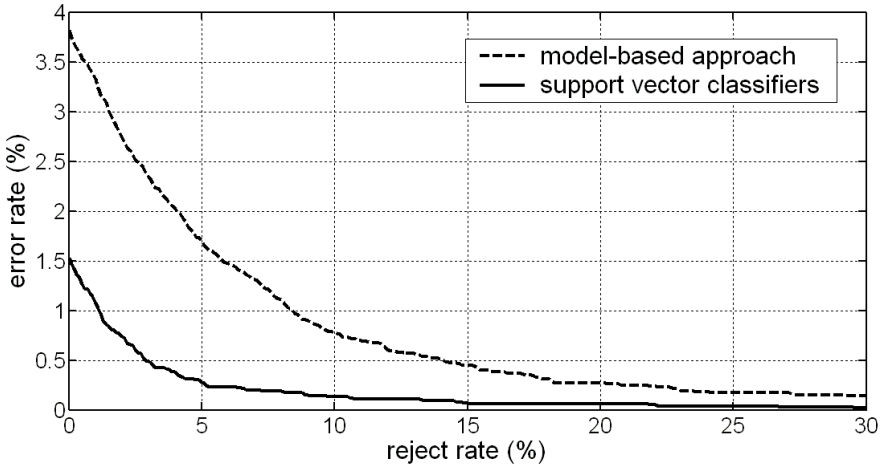


Figure 10: Error-reject tradeoff of Support Vector Classifiers on validation dataset

On the other hand, we adopt the number of kernel evaluation per pattern (KEPP) as a measure for the classifying cost, since it is the main cause of the computation effort during the test phase. Thus, our ensemble of 45 SVCs requires 11,118 KEPPs to make decision.

4.3 Two-stage classification system

As we can see on Table 3, after the first stage of classification the label of the data is not always in the first two classes, which justifies the choice of a dynamic number of classes in conflict.

ranking of the label	1	2	3	> 3
% of the dataset	96.18	2.50	0.76	0.56

Table 3: Ranking distibution of the label obtained with the model-based approach on the validation dataset

According to the application constraints, it is necessary to make a compromise between accuracy and complexity. The threshold ε of equation (5) controls this tradeoff. Then, the validation dataset can be used to fix this parameter according to the constraints fixed by the application.

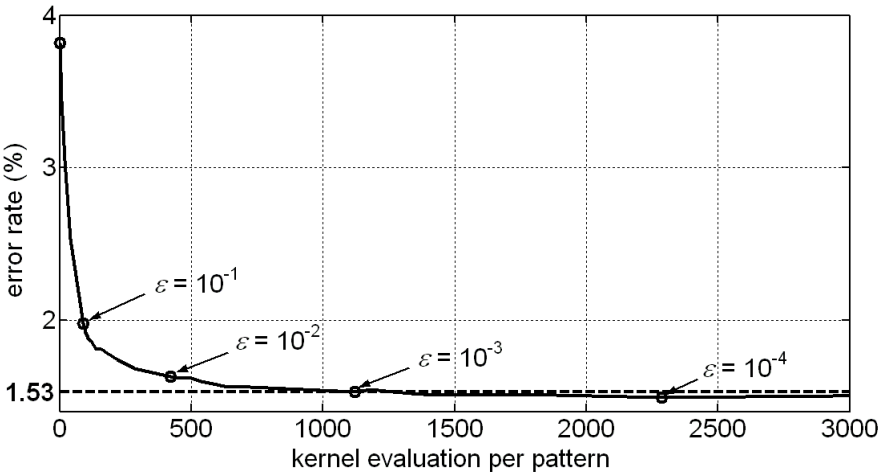


Figure 11: Accuracy-complexity tradeoff on the validation dataset

As we can see in Figure 11, while using a threshold of 10^{-3}, it is possible to obtain exactly the same error rate of 1.53% than with the full "pairwise coupling" ensemble. Moreover, the use of a smaller threshold ($\varepsilon = 10^{-4}$) allows a slightly better error-reject tradeoff (see Figure 12), but the number of KEPP is multiplied by two.

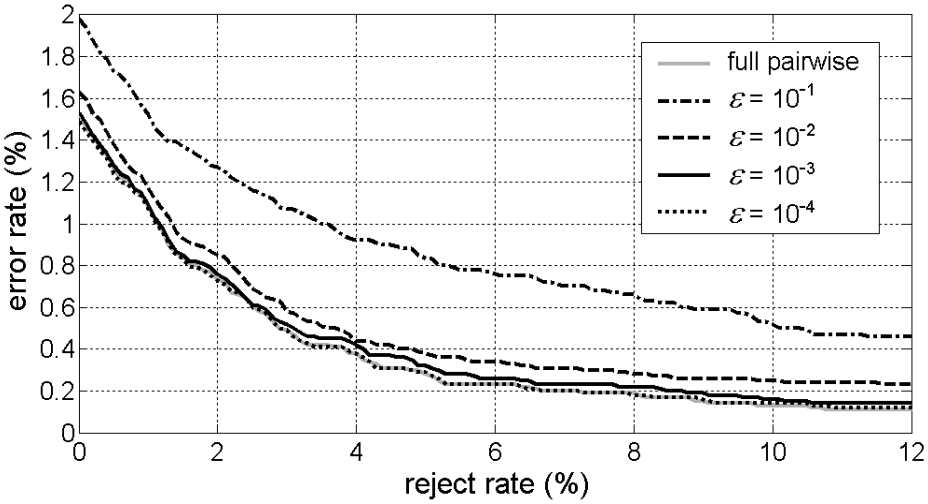
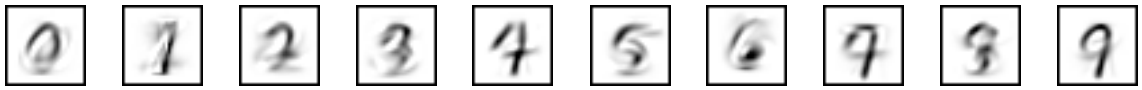


Figure 12: Error-reject tradoff of our two-stage classification system on the validation dataset

For this reason, we fix the tolerance threshold ε at 10^{-3}, which seems a good tradeoff between accuracy and complexity. The Figure 13 shows an example of ambiguous pattern. We can see in dark the posterior probability efficiently re-estimated by the second stage. Thus, if we had used $\varepsilon = 10^{-4}$, we would have obtained for this example a number $p = 7$ of classes in conflict and we would have used 21 SVCs to re-estimate posterior probabilities.



$f_j(x)$:										
class:	ω_1	ω_2	ω_3	ω_4	ω_5	ω_6	ω_7	ω_8	ω_9	ω_{10}
$d_j(x)$:	4.9319	5.3658	4.8692	4.8020	**3.2414**	5.1457	5.4457	**3.5028**	4.7798	**3.4728**
$\hat{P}_f(\omega_j \mid x)$:	0.0001	0.0000	0.0001	0.0001	**0.6641**	0.0000	0.0000	**0.1537**	0.0001	**0.1818**
$\hat{P}_s(\omega_j \mid x)$:	0.0001	0.0000	0.0001	0.0001	**0.0000**	0.0000	0.0000	**0.9967**	0.0001	**0.0029**

Figure 13: Example of ambiguous pattern (5,907th sample of the test dataset)

Also, while the number p of SVCs used is dynamic, it is interesting to observe the distribution of p (Figure 15). Hence, we can see that with our threshold of 10^{-3}, the half of the examples are processed without SVC, which confirms the previous remark related to Figure 7.

Finally, our two-stage system uses a mean of 1,120.1 KEPP and obtained on the test dataset an error rate of 1.50 %, which is comparable to the result of the full "pairwise coupling" ensemble (1.48 %). The analysis of these 150 errors reported in Figure 14, shows that only one error is due to the first stage, which classify directly 4,890 test samples.

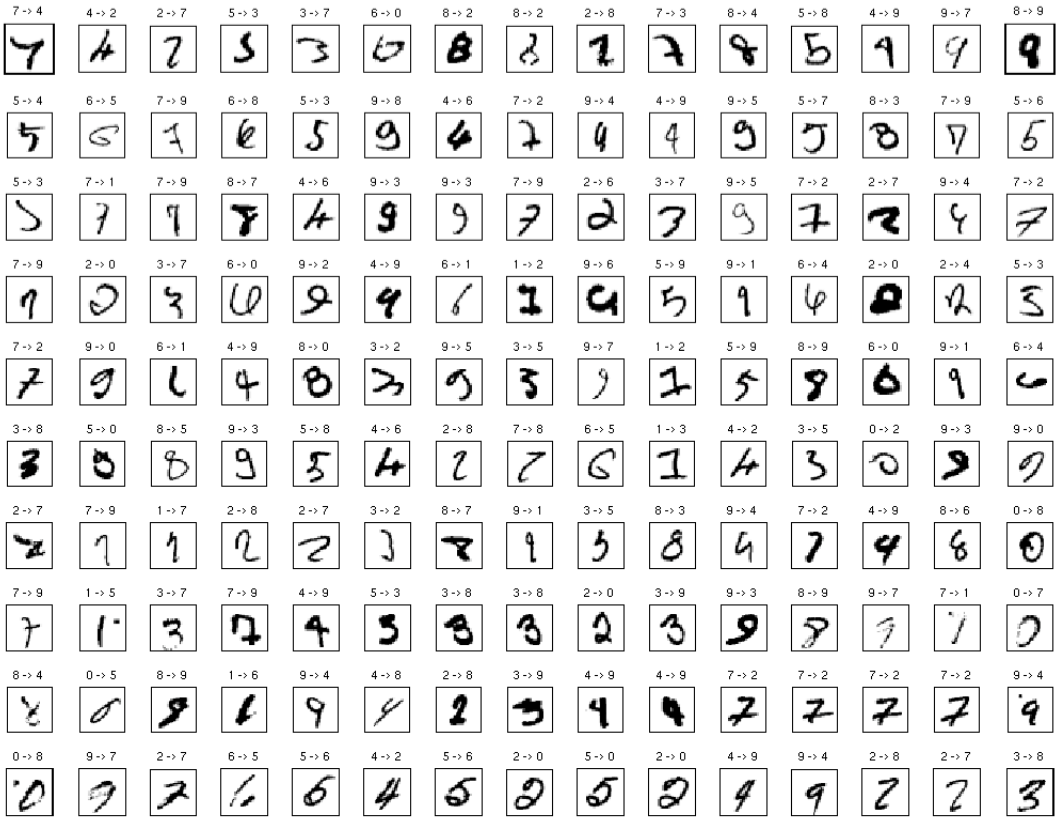


Figure 14: The 150 errors obtained on the test dataset (label -> decision)

Moreover, as we can see in Figure 15, it is necessary to use more than one SVC to resolve conflict. This fact shows that the first level is not effective enough.

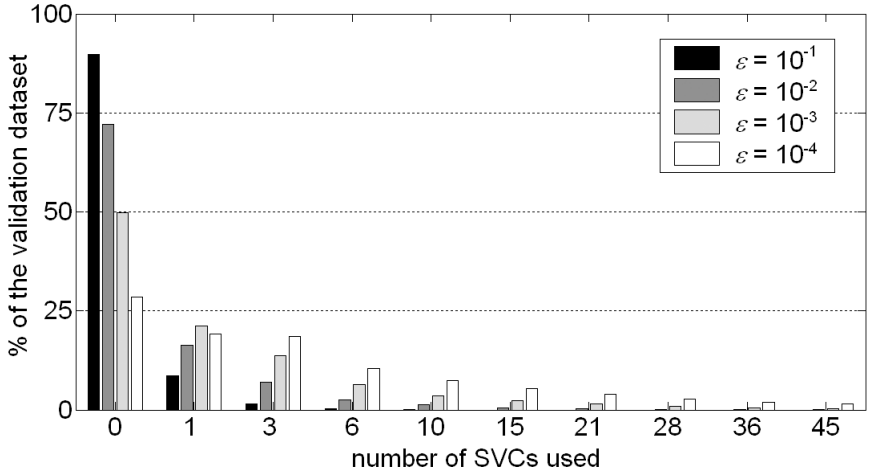


Figure 15: Distribution of the number *p* of SVCs used to classify the validation dataset

5 Conclusions and perspectives

We have presented a new classification architecture that has several interesting properties for application to pattern recognition. It combines the advantages of a model-based classifier, in particular modularity and efficient rejection of outliers, with the high accuracy of SVC. Moreover, it greatly reduces the decision time related to the SVC, which is very important in the majority of real pattern recognition systems.

The results on the MNIST database show that the use of the first stage to estimate probabilities allows to reduce the classifying cost by a factor 8.7, while preserving the accuracy of the full "pairwise coupling" ensemble (see Table 4). Indeed, if we express the computational complexity in number of floating point operations (FLOPs), a kernel evaluation requires 2,355 FLOPs and a projection distance evaluation requires 81,510 FLOPs. Thus, the computational cost necessary to classify a pattern is approximately 26.2 MFLOPs with the full "pairwise coupling" ensemble, only 0.4 MFLOPs with the model-based approach and an average of 3.0 MFLOPs with our dynamic two-stage process.

error rate (%)		**0.5**	**0.4**	**0.3**	**0.2**	**0.1**
reject rate (%)	**model-based approach**	12.68	13.74	16.97	20.01	28.59
	our two-stage system	3.31	3.99	4.94	6.57	9.85
	full "pairwise coupling"	3.29	4.00	5.13	6.34	9.55

Table 4: Error-reject tradeoff of the three approaches on the test dataset

Furthermore, while this implementation is only a proof of concept, several aspects can be improved in future works. Indeed, the model-based approach used in the first stage is not accurate. Thus, the use of a mixture of hyperplanes to model each class instead of one single hyperplane per class should improve significantly the accuracy of the first stage. Then, it will be interesting to test the capability of model-based approach to reject outliers. With this intention, we propose to generate a database of artificial outliers like "touching digit" shown in Figure 9.

In addition, to improve the generalization performance, it is preferable to extract discriminative features, as in [19] where 8-direction gradient features are extracted and allows to reduce the error-rate to only 0.4 %. On the other hand, it will be interesting to train local SVC only with training data rejected by the first stage.

To conclude, the modularity of the proposed architecture open the way to use SVC to resolve classification problems with a large number of classes. Indeed, we can use the first stage, which are suited for this type of problems, to evaluate the possible conflict and we construct only the appropriate SVCs.

References

[1] A. Bellili, M. Gilloux and P. Gallinari (2003) An MLP-SVM combination architecture for offline handwritten digit recognition, *International Journal on Document Analysis and Recognition*, 5(4), 244-252.

[2] C.M. Bishop (2004) Generative versus Discriminative Methods, in Computer Vision, invited keynote talk at *International Conference on Pattern Recognition. Powerpoint slides are available at* http://research.microsoft.com/~cmbishop/

[3] C.-C. Chang and C.-J. Lin (2001) LIBSVM : a library for support vector machines. Software available at: http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

[4] V. Di Lecce, G. Dimauro, A. Guerriero, S. Impedovo, G. Pirlo and A. Salzo (2000) Classifier combination: the role of a-priori knowledge, *International Workshop on Frontiers in Handwriting Recognition*, 143-152.

[5] E. Francesconi, M. Gori, S. Marinai and G. Soda (2001) A serial combination of connectionist-based classifiers for OCR, *International Journal on Document Analysis and Recognition,* 3(3), 160-168.

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

[7] M. Gori and F. Scarselli (1998) Are Multilayer Perceptrons Adequate for Pattern Recognition and Verification ?, *IEEE transaction on Pattern Analysis and Machine Intelligence*, 20(11), 1121-1132.

[8] V. Gunes, M. Ménard and P. Loonis (1999) Fuzzy clustering with ambiguity for multi-classifiers fusion: Clustering-Classification Cooperation. *EUSFLAT-ESTYLF Joint Conference*, 505-508.

[9] V. Gunes, M. Ménard, P. Loonis and S. Petit-Renaud (2003) Combination, cooperation, and selection of classifiers, *International Journal of Pattern Recognition and Artificial Intelligence*, 17(8), 1303-1324.

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

[11] F. Kimura, S. Inoue, T. Wakabayashi, S. Tsuruoka and Y. Miyake (1998) Handwritten numeral recognition using autoassociative neural networks, *International Conference on Pattern Recognition,* 166-171.

[12] F. Kimura, K. Takashina, S. Tsuruoka and Y. Miyake (1987) Modified Quadratic Discriminant functions and the Application to Chinese Character Recognition, *IEEE transaction on Pattern Analysis and Machine Intelligence*, 9(1), 149-153.

[13] J. Kittler, M. Hatef, R.P.W. Duin, and J. Matas (1998) On combining classifiers, *IEEE transaction on Pattern Analysis and Machine Intelligence*, 20(3), 226-239.

[14] L.I. Kuncheva, J.C. Bezdek and R.P.W. Duin (2001) Decision templates for multiple classifier fusion: an experimental comparison. *Pattern Recognition*, 34(2), 299-314.

[15] L. Lam (2000) Classifier Combinations: Implementations and Theoretical Issues, *Multiple Classifier Systems, volume 1857 of Lecture Notes in Computer Science*, 77-86.

[16] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner (1998) Gradient-based learning applied to document recognition. *Proceedings of IEEE,* 86(11), 2278-2324.

[17] H.-T Lin, C.-J. Lin and R.C. Weng (2003) *A note on Platt's probabilistic outputs for support vector machines*. Technical report, Department of computer science and information engineering, National Taiwan University. Available at http://www.csie.ntu.edu.tw/~cjlin/papers.html

[18] C.-L. Liu, H. Sako and H. Fujisawa (2002) Performance evaluation of pattern classifiers for handwritten character recognition, *International Journal on Document Analysis and Recognition*, 191-204.

[19] C.-L. Liu, K. Nakashima, H. Sako and H. Fujisawa (2003) Handwritten digit recognition: benchmarking of state-of-the-art techniques, *Pattern Recognition*, 36(10), 2271-2285.

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

[21] L. Prevost, C. Michel-Sendis, A. Moises, L. Oudot and M. Milgram (2003) Combining model-based and discriminative classifiers: application to handwritten character recognition, *International Conference on Document Analysis and Recognition,* 31-35.

[22] N. Ragot and E. Anquetil (2003) A generic hybrid classifier based on hierarchical fuzzy modeling: Experiments on on-line handwritten character recognition, *International Conference on Document Analysis and Recognition,* 963-967.

[23] H. Schwenk (1998) The diabolo classifier, *Neural Computation*, 10(8), 2175-2200.

[24] L. Vuurpijl, L. Schomaker and M. Van Erp (2003) Architectures for detecting and solving conflicts: two-stage classification and support vector classifiers, *International Journal on Document Analysis and Recognition,* 5(4), 213-223.