Improving classification using a Confidence Matrix based on weak classifiers applied to OCR

Juan Ramon Rico-Juan[a], Jorge Calvo-Zaragoza[a,*]

[a]*Software and Computing Systems, University of Alicante, Carretera San Vicente del Raspeig s/n, E-03071 Alicante, Spain*

Abstract

This paper proposes a new feature representation method based on the construction of a Confidence Matrix (CM). This representation consists of posterior probability values provided by several weak classifiers, each one trained and used in different sets of features from the original sample. The CM allows the final classifier to abstract itself from discovering underlying groups of features. In this work the CM is applied to isolated character image recognition, for which several set of features can be extracted from each sample. Experimentation has shown that the use of the CM permits a significant improvement in accuracy in most cases, while the others remain the same. The results were obtained after experimenting with four well-known corpora, using evolved meta-classifiers with the k-Nearest Neighbor rule as weak classifier and by applying statistical significance tests.

Keywords: confidence matrix, posterior probability, weak classifiers, feature spaces

1. Introduction

Classification systems have been widely studied in pattern recognition tasks. The classical classification scheme is based on a sequential model that consists in extracting features from a sample, using a classification technique and obtaining a final hypothesis [7]. This scheme has been exploited in

*Corresponding author
Email addresses: JuanRamonRico@dlsi.ua.es (Juan Ramon Rico-Juan), jcalvo@dlsi.ua.es (Jorge Calvo-Zaragoza)

order to attain fairly complex techniques with which to improve classification accuracy, such as Artificial Neural Networks [11] or Support Vector Machines [4]. The evolution in this field has led to the development of new schemes in supervised learning. For example, a new classification scheme has emerged based on the assumption that it is more robust to combine a set of simple hypotheses than to use just one complex hypothesis [13].

This scheme can be viewed from different perspectives according to the means used to combine decisions. On the one hand, there are algorithms that combine the scores of individual classifiers (usually weak classifiers) to produce a final score, and which are commonly referred to as ensemble classifiers. A wide analysis of this kind of algorithms can be found in Kuncheva [14]. On the other hand, another approach has recently been proposed which uses several dissimilarity measures from the samples to obtain different scores that are subsequently combined. This approach has several approximations, which are described in Pekalska and Duin [19]. A more recent article [12] proposes refinements of error correction for fusion strategies in classification.

In this paper we propose a kind of combination of the two aforementioned schemes: first, each weak classifier –in our case classifiers based on the nearest neighbor rule (NN) [6]– provides the probability of belonging to each class. All these probabilities are in turn combined to form a Confidence Matrix (from here on referred to as CM) which is used as input to a final classifier.

The construction of this matrix can be viewed as the same basic idea as that of the *Stacking* [32] family algorithms. These algorithms are based on the generation of meta-features. Each feature represents the *a posteriori* probability of the actual prototype belonging to each class depending on each weak classifier. In its initial version, Stacking is used to obtain the probability of each possible class using all the weak classifiers, and then it classifies the samples in the space of meta-features, principally through the use of a multilinear regression approach. An evolved version known as Stacking-C [26] generates these meta-features class by class, adding an additional feature to indicate whether the sample belongs to the class being treated.

The construction of the CM just requires different groups of features to be extracted from the original signal. Each one of these groups has to be used with a different weak classifier, so that the final meta-classifier does not have to discover these underlying points of view by itself. Hence, our work establishes a case of study in which the CM representation is applied to the OCR task since this kind of data is known to allow several ways of extracting features [22].

In this paper, some meta-classifiers are tested by using original features and meta-features (CM). The experiments show how the power of this technique lies in the mapping of features onto meta-features. When the meta-feature space is used, any advanced classifier can be applied to recognize the samples without being limited to a set of algorithms based on linear regression. That is, the intention of this paper is to address the construction of a CM that can be used at the meta-feature level and combined with any meta-classifier. As discussed in the experimental section, the accuracy of the results obtained using CM are, in most cases, significantly better or, at worst, the same as when using the original features. These empirical results are obtained by means of several experiments using different corpora, various evolved meta-classifiers and statistical analysis techniques.

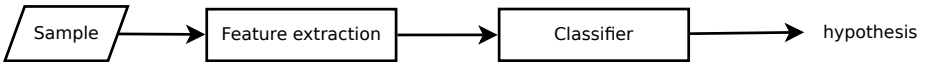
The remainder of the paper is structured as follows: Section 2 describes the construction of the Confidence Matrix. Section 3 details the experimental setup. Section 4 shows the results obtained. The paper ends in Section 5 with our conclusions and future work.

2. A new classification scheme based on confidence measures

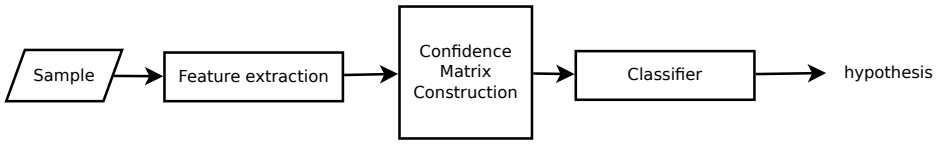
This section presents a new classification scheme based on the generation of a Confidence Matrix. This section will present a generic construction of this representation regardless the specific task or set of features. The application of this construction for its use in the OCR task will be addressed in the next section.

If Ω is a set of class labels and D is the set of weak classifiers, then a $|D| \times |\Omega|$ matrix is obtained. This matrix (CM) contains the confidence (represented as probabilities) that the weak classifiers give to each prototype belonging to each class. That is, CM_{ij} represents the probability that the sample belongs to the class Ω_i based on the weak classifier D_j. The CM can thus be viewed as a new feature representation (meta-features) that can be used to feed the final classifier rather than using the original features (see Figure 1).

When this matrix is used, the final classifier does not need to distinguish the different points of views of the signal. In classical approaches, the final classifier has the responsibility of discovering them by itself. Furthermore, unlike that which occurs with the ensemble classifiers, this new scheme avoids the need to define distinct dissimilarity measures or types of weak classifiers.



(a) Classification scheme without CM



(b) Classification scheme with CM

Figure 1: Classification schemes with and without Confidence Matrix (CM).

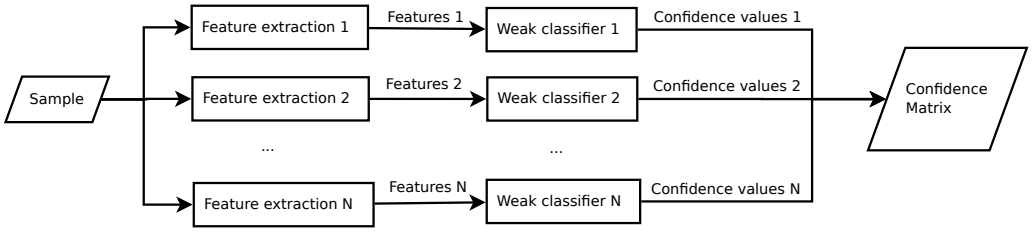


Figure 2: Construction of the Confidence Matrix.

It is only necessary to group the input characteristics according to their nature, which is often relatively simple for a user with domain expertise.

In order to build the CM, it is necessary to train a set of weak classifiers, each of which is responsible for exploiting one group of features separately. One weak classifier is therefore trained for each set of features, thus producing confidence values that work on the different points of view of the input data. Each weak classifier must generate a vector of confidence values that are grouped to form the final CM (see Figure 2). Each individual weak classifier can use different methods or measures to estimate the probability. In our case, the same methods are used based on different groups of input features, as will be shown in the experimental section.

These confidence values should indicate the possibility of the sample belonging to a certain class. Although these confidence values do not have to exactly represent a probability, in this paper the values will be formalized as posterior probabilities. It is thus possible to state that the CM is composed of the likelihood of the sample belonging to each class according to each weak

classifier considered.

From the algorithmic point of view, the CM representation entails some interesting advantages: (1) the implementation is very straightforward and it only requires weak classifiers; (2) the pipeline of the algorithm can be easily parallelized so that each weak classifier runs at the same time. Additionally, there may be some scenarios in which the CM is not only helpful but necessary. For example, when several input signals from the same sample come from different, incompatible structures (eg. trees, strings or feature vectors). In these cases, scores from weak classifiers trained separately with each kind of structure can be easily combined (early fusion) within the CM representation.

Note, however, that this new scheme does not produce a final decision. It merely maps the input features into another space (meta-features). This signifies that it is necessary to use an algorithm (meta-classifier) that employs the CM to make a decision. From this point of view, CM representation is therefore related to both early and late fusion. Several inputs are combined with the CM structure which can be seen as an early fusion for the final meta-classifiers at the same time it is a late fusion from the weak classifiers point of view.

Our assumption is that it is usually simple to provide some good weak classifiers (at least, better than random decisions). If some good weak classifiers are provided, it is expected that the final meta-classifiers can detect which meta-features are most promising to use in the decision. Thus, if some weak classifier is giving bad meta-features, it is also expected that the final classifier can detect that it is better to avoid their scores.

Since we only provide a new representation of the input, it should be emphasized that the main goal is to prove that the use of the CM either improves on or maintains the accuracy obtained without it. To this end, a series of meta-classifiers for its use as a final algorithm will be considered in the experimentation.

3. Experimental setup

In this paper, various experiments are carried out to compare the benefits attained by the use of the CM. The experiments are focused on classifying binary OCR images, using four different datasets:

- The NIST SPECIAL DATABASE 3 of the National Institute of Standards and Technology, from which a subset of the upper case characters

(26 classes, [A-Z]) was randomly selected: 6500 images (250 examples per class) from 500 writers.

- The MNIST dataset of handwritten digits (10 classes, [0-9]) [16]: 60000 training images and 10000 test images. In our experiments, both sets will be mixed.

- The United States Postal Office handwritten digit (10 classes) dataset [9]: 9298 images.

- The MPEG-7 shape silhouette dataset (Core Experiment CE-Shape-1 part B, 70 classes) [15]: 1400 images. Although this dataset is not composed by characters, the images contained are very similar to that of previous datasets. Thus, it is included in the experiments to test the technique proposed.

3.1. Feature extraction from binary images

As it was mentioned previously, the inclusion of the CM is projected for tasks in which it is possible to extract different groups of features. The same feature extraction as that detailed in [22] was therefore used. Nevertheless, a brief explanation is provided in the following paragraphs.

As depicted in Section 2, the main idea is to obtain different kinds of features, each of which will be used on a specific weak classifier. A preprocessing stage is performed first, during which the image is binarized using a global histogram threshold algorithm [18]. A morphological closing filter [27] is then applied in order to correct any gaps and spurious points that may have appeared. In the next step, the character is located in the image and the region of interest (ROI) is selected. The ROI is divided into a substructure of smaller regions in order to extract local features. The number of sub-regions must be fixed according to each dataset (see Table 1)

Once the image has been preprocessed, the feature extraction takes place. Four types of features are extracted:

- Foreground features. A vector with the number of foregrounds pixels in each of the image sub-regions is produced.

- Background features. The background feature extraction is based on that of [30]. It computes four projections (up, down, left, and right) which are considered for each pixel in the image. When any of these

subregion	NIST	MNIST	USPS	MPEG-7
02 x 02	39.7 (1.6)	35.5 (1.5)	38.5 (1.7)	41.2 (1.8)
03 x 03	19.1 (1.4)	21.2 (1.3)	23.1 (1.5)	26.0 (1.6)
04 x 04	10.1 (1.2)	16.4 (1.1)	16.4 (1.2)	12.4 (1.1)
05 x 05	8.5 (0.9)	13.1 (1.0)	10.2 (1.0)	9.3 (1.0)
06 x 06	**8.3 (1.0)**	**11.2 (0.9)**	6.1 (0.9)	**8.1 (0.9)**
07 x 07	8.5 (1.0)	12.0 (1.0)	**4.2 (0.8)**	9.2 (1.0)
08 x 08	8.7 (0.9)	13.3 (1.0)	5.3 (0.9)	10.0 (1.0)
09 x 09	10.0 (1.0)	15.4 (1.1)	6.8 (1.1)	12.5 (1.1)

Table 1: Mean error rates (standard deviation) of 4-cross-validation preliminary experiment with different sub-region sizes. Results for each database using the MACP (Maximum Average Class Probability) algorithm are shown.

projections touches the foreground object, the number associated with that pixel increases by one unit. It is thus possible to distinguish four different categories of background pixels, according to their projection values $(1, 2, 3, 4)$. A fifth category is also added in order to provide more information: there are two situations that are similar in geometry but are totally different from a topological point of view. Our algorithm therefore assigns a value of 5 to the category if the pixel lies in an isolated background area, signifying that five vectors are extracted as features, one for each pixel projection category. Each vector represents the number of pixels with the particular category in each image sub-region.

- Contour features. The object contour is encoded by the links between each pair of 8-neighbor pixels using 4-chain codes in the manner proposed by [17]. These codes are used to extract four vectors (one for each direction), and the number of each code is counted in each image sub-region.

3.2. Weak classifiers

In order to construct the CM, a set of weak classifiers with which to map each group of features onto confidence values is needed. In this case, a formula based in the Nearest Neighbor (NN) [6] rule was used since it is a common, robust and simple technique with which to produce a weak classifier. As discussed in Section 2, one weak classifier per group of features

should be generated. Each weak classifier is trained using a leaving-one-out scheme: each single sample is isolated from the training set T and the rest are used in combination with the NN to produce the confidence values. The formula detailed below is used in Rico-Juan and Iñesta [23] and is inspired by Pérez-Cortés et al. [20]. If x is a training sample, then the confidence value for each class $w \in \Omega$ is based on the following equation:

$$p(w|x) = \frac{1}{\min_{x' \in T_w, x \neq x'} d(x, x') + \epsilon} \tag{1}$$

where T_w is the training set for w label and ϵ is a non-zero value provided to avoid infinity values. In our experiments, the dissimilarity measure $d(\cdot, \cdot)$ is the Euclidean distance. After calculating the probability for each class, the values are normalized such that $\sum_{w \in \Omega} p(w|x) = 1$. Once each training sample has been mapped onto the CM, the samples can be used in the test phase.

3.3. Evolved meta-classifiers

Once the CM has been calculated as explained in the previous section, it must be used in combination with a classifier to output a hypothesis. In this work we shall use well-known evolved meta-classifiers. The intention of this measure is to avoid the possibility that improvements in the results may be caused by the use of over simple classifiers. The underlying idea of meta-classification is to solve the problem of combining classifiers. A meta-classifier is in charge of gathering individual classification decisions in order to combine them into a unique final decision. We shall use the following three meta-classifiers: Maximum Average Class Probability, Stacking-C, Rotation Forest.

Note that we are not trying to outperform the existing late fusion techniques. Since these three classifiers perform some kind of late fusion by their own, our intention is just to find out whether our CM representation can improve the performance achieved with classical feature vectors.

In addition to these three meta-classifiers, Support Vector Machines and Multi-Layer Perceptron algorithms are also included in the experimentation. All of these techniques will be experimentally compared with and without the use of the CM.

3.3.1. Maximum Average Class Probability

This meta-classifier is based on combining decisions by using the voting methods from the weak classifier hypothesis with the average rule [14]. In this case, each weak classifier classifies a new sample by computing the *a posteriori* probability for each class. The class that obtains the maximum average from among these values is selected. This method was chosen as baseline because of the good results obtained previously for this type of tasks [22]. In this previous work, the classification error rate was lower than 1-NN technique applied to each group of individual features (image, background, contour) and than a 1-NN technique gathering as input the three groups of features.

3.3.2. Stacking-C

Given that our classification scheme is based on the main idea of Stacking algorithms, we have included this algorithm to prove the improvement that can be obtained by means of the CM. We have selected one of the most successful algorithms from this family: Stacking-C [26], an extension to Stacking with which to accurately address multi-label classification problems.

3.3.3. Rotation Forest

Rotation Forest (RoF) [24] is an ensemble method that is focused on building accurate and diverse classifiers. It trains a set of decision trees (forest), each of which uses an independent feature extraction. RoF makes use of a base classifier to generate the decision trees. In our work, two alternatives will be considered: C4.5 [21] (J48 implementation [8]) and Random Forest (RaF) [3]. The first alternative is proposed by the original RoF article and by WEKA Data Mining tool [8] as a default parameter, whilst the latter is considered in this paper due to its best performance in our OCR preliminary experiments despite not being a common ensemble in RoF experimentation.

3.3.4. Support Vector Machines

Support Vector Machines (SVM) is a common pattern recognition algorithm developed by Vapnik [29]. It seeks for a hyperplane which maximizes the separation (margin) between the hyperplane and the nearest samples of each class (support vectors). The libSVM implementation [8] with Polynomial kernel is used in our experimentation.

3.3.5. Muti-Layer Percepton

Artificial Neural Networks is a family of structures developed in an attempt to mimic the operation of the nervous system to solve machine learning problems. The topology of a neural network can be quite varied. For this work, the common neural network called Multi-Layer Perceptron (MLP) [25] is used, as implemented in [8].

4. Results

The results of each dataset are detailed in the following subsections. Table 6 shows a summary of the average final results. A short discussion about the statistical significance of the results is also developed at the end of this section. The WEKA version 3.6.9 tool [8] has been used for testing RoF and Staking-C algorithms with their default parameters.

Note that our main goal is not to measure the goodness of each considered ensemble but to compare their results with and without using the CM representation proposed.

4.1. NIST SPECIAL DATABASE 3

For this dataset, the best number of image sub-regions is 6, signifying that $(6 \times 6) + (6 \times 6) \times 5 + (6 \times 6) \times 4 = 360$ features are extracted from each of the samples in this set. The results of the experimentation is shown in Table 2. Upon viewing the results it will be noted that the inclusion of the CM has, on average, improved the results of all the algorithms.

Note that the improvement achieved by using the CM in both SVM and MLP is remarkably high. The latter case is specially interesting since this algorithm has the best error rate when using CM representation, which did not occur without it.

4.2. MNIST

The number of optimum image sub-regions in this dataset is also 6, signifying that 360 features are again used. Table 3 details the results of the experiment for this dataset. The results follow a similar trend to those of the NIST dataset. In this case, the improvement achieved by the inclusion of the CM would appear to be even more noticeable. MLP (with CM) was again reported as being the best classifier.

dataset	RoF RaF		RoF J48		MACP		Stacking		SVM		MLP	
	with	without	with	without	with	without	with	without	with	without	with	without
A	3.5	4.0	5.0	6.5	7.0	6.0	9.5	8.5	9.0	18.5	1.0	10.5
B	6.5	6.0	6.5	10.0	6.0	5.5	7.5	14.5	11.5	14.5	2.5	58.0
C	4.0	3.0	4.5	6.0	5.5	4.5	8	8.5	11.5	12.5	4.0	8.5
D	10.5	14.0	11.0	18.5	14.5	11.5	16.5	22	33.5	43.5	8.0	61.5
E	2.5	6.0	4.0	11.0	5.0	6.0	6.5	9	1.5	20.5	6.0	35.5
F	2.5	3.0	4.5	4.0	2.0	3.5	3	5.5	6.0	14.0	3.5	30.0
G	7.5	9.0	8.0	10.0	9.5	8.5	10	17	8.5	25.0	7.0	32.5
H	3.5	7.5	6.5	6.5	5.0	7.5	5.5	13.5	2.0	22.0	4.5	33.5
I	6.5	10.5	4.5	8.0	6.0	7.5	9	11	42.5	45.5	5.5	72.5
J	5.0	6.0	8.0	5.0	7.5	5.5	8	9.5	13.5	45.0	5.0	58.0
K	4.0	7.0	5.0	9.0	9.5	9.0	7.5	18	5.0	26.0	4.0	46.5
L	3.5	4.0	5.5	6.0	4.5	6.0	4	3.5	25.5	28.5	4.5	7.5
M	4.0	2.5	5.5	7.0	5.0	3.5	4.5	6.5	7.0	18.0	5.0	34.5
N	7.0	8.5	10.5	10.0	10.5	8.5	12	20	8.5	38.5	6.0	55.5
O	12.0	13.0	12.0	14.5	18.5	11.0	19.5	15.5	16.5	18.0	10.0	38.0
P	4.0	5.0	5.0	5.0	3.0	3.5	4	5	16.0	13.0	5.0	34.5
Q	13.5	16.0	12.0	15.0	7.5	15.0	13	24	25.0	39.0	5.5	16.0
R	5.0	7.0	7.5	6.5	9.5	9.0	9	13	20.0	26.5	4.0	55.5
S	5.0	6.5	7.0	8.0	4.5	9.5	8	10.5	9.0	14.0	4.5	7.0
T	1.0	2.5	2.0	2.5	1.0	2.0	3.5	2.5	14.5	20.0	2.0	52.5
U	6.5	10.0	6.5	12.0	10.5	12.0	15	11	18.5	29.5	8.0	59.0
V	8.5	8.5	11.0	10.0	10.5	8.0	10.5	10	22.5	34.5	9.0	38.5
W	7.0	6.5	6.5	8.0	4.5	6.5	5	6.5	7.5	23.5	4.0	12.0
X	7.5	8.5	9.5	11.5	4.5	11.0	8.5	16.5	17.0	25.0	4.5	33.0
Y	8.0	12.5	9.0	10.5	6.5	8.5	7	14	12.0	27.5	7.0	57.0
Z	3.0	2.0	3.5	3.5	3.0	3.0	5	3.5	11.0	13.5	4.0	30.0
avg. error	**5.8**	7.3	**6.9**	8.6	**7.0**	7.4	**8.4**	11.5	**14.4**	25.2	**5.2**	37.6

Table 2: Comparison of NIST classification average error rate per class with 4-cross-validation, comparing the methods with and without CM.

dataset	RoF RaF		RoF J48		MACP		Stacking		SVM		MLP	
	with	without	with	without	with	without	with	without	with	without	with	without
0	1.0	5.0	1.0	4.5	3.0	1.0	2.5	1.0	8.3	5.9	0.9	50.9
1	3.5	2.0	3.5	3.0	3.0	4.5	3.0	2.0	11.9	20.6	3.3	58.6
2	7.0	8.5	7.5	8.0	8.0	15.5	11.5	8.5	10.4	27.3	3.4	10.1
3	9.5	13.0	11.5	12.0	14.0	14.5	12.0	15.0	20.3	15.0	6.8	53.9
4	4.5	7.5	6.5	8.5	4.0	9.0	5.0	8.0	12.1	22.3	3.3	76.3
5	7.0	13.0	9.5	13.0	9.5	17.0	11.5	17.0	6.9	26.5	4.1	78.5
6	2.5	3.0	3.0	3.5	4.5	3.0	6.5	4.0	10.4	7.5	1.6	1.9
7	5.0	8.5	4.5	8.0	10.0	9.5	9.0	12.0	43.0	30.9	3.6	3.8
8	7.0	13.5	9.5	16.0	13.5	17.0	12.5	15.5	9.6	22.0	5.5	56.9
9	7.5	12.5	8.0	10.0	11.0	10.0	10.5	8.5	13.8	21.0	8.3	80.0
avg. error	**5.5**	8.7	**6.5**	8.7	**8.1**	10.1	**8.4**	9.2	**14.7**	19.9	**4.1**	47.1

Table 3: Comparison of MNIST classification average error rate per class with 4-cross-validation, comparing the methods with and without CM.

dataset	RoF RaF		RoF J48		MACP		Stacking		SVM		MLP	
	with	without	with	without	with	without	with	without	with	without	with	without
0	3.0	4.5	3.0	3.9	3.0	2.0	5.0	2.0	0.6	0.0	1.6	37.2
1	4.5	7.0	5.5	5.5	3.9	3.9	3.9	5.5	1.5	65.8	0.5	27.7
2	2.5	3.9	3.0	3.5	3.5	3.5	3.9	3.5	0.2	98.7	1.3	40.8
3	4.0	6.0	6.9	6.5	4.5	4.5	4.0	5.0	82.7	90.7	1.8	34.9
4	4.0	6.5	4.5	7.0	4.5	4.5	5.0	5.5	34.2	99.7	1.8	99.7
5	3.0	5.0	3.5	6.0	5.0	4.0	4.5	4.0	79.7	79.7	2.1	79.7
6	1.5	3.0	2.5	2.0	1.5	1.0	1.5	0.0	5.7	84.7	1.0	51.1
7	3.5	4.0	4.5	4.5	3.5	3.5	2.5	4.5	25.1	73.2	1.1	31.1
8	2.0	4.5	2.0	4.0	4.0	3.5	2.5	3.5	82.7	82.7	1.2	19.9
9	1.5	3.0	2.0	3.0	4.0	1.5	2.5	2.0	42.1	88.2	1.5	18.9
avg. error	**2.9**	3.7	**3.7**	4.6	3.7	**3.2**	3.5	3.5	**35.4**	76.3	**1.4**	44.1

Table 4: Comparison of USPS classification average error rate per class with 4-cross-validation, comparing the methods with and without CM.

4.3. USPS

The preliminary results as regards obtaining the best number of image sub-regions is 7 in the case of the USPS dataset, and $(7 \times 7) + (7 \times 7) \times 5 + (7 \times 7) \times 4 = 490$ features are therefore considered. The results of the final experiment are shown in Table 4. This is the first case in which the inclusion of the CM does not improve all the classifiers considered, since $MACP$ increases its error rate from 3.2 to 3.7 when using the CM. The other classifiers decrease or maintain their error rates with the CM. Once again, the MLP (with CM) classification achieved the best classification result.

4.4. MPEG-7

As occurred for the NIST and MNIST, the best image sub-region size for the MPEG-7 database is 6, and 360 features are therefore used to classify the samples. Table 5 shows the results of the classification experiment with this database. The results of the datasets in which all the classifiers obtain a perfect classification have been removed owing to the size of the table. Note that some classifiers are enhanced with the inclusion of the CM while others are not. This also occurred in previous databases, but in this case the best classification was obtained for the MACP without CM.

4.5. Statistical significance

The intention of this experimental section is to assess whether the inclusion of the CM can achieve significantly better classification results. We shall therefore use the KEEL [1] software, which contains statistical tools. These tools will allow us to quantify the difference between the results with

dataset	RoF RaF		RoF J48		MACP		Stacking		SVM		MLP	
	with	without	with	without	with	without	with	without	with	without	with	without
bat	6.3	6.3	18.8	18.8	6.3	6.3	18.8	18.8	25.0	31.3	6.3	43.8
beetle	12.5	0.0	0.0	12.5	12.5	0.0	12.5	12.5	25.0	37.5	0.0	50.0
bird	43.8	50.0	43.8	56.3	25.0	50.0	50.0	37.5	68.8	56.3	43.8	81.3
butterfly	12.5	18.8	25.0	56.3	31.3	12.5	12.5	18.8	31.3	6.3	18.8	56.3
camel	0.0	12.5	6.3	12.5	6.3	6.3	6.3	0.0	25.0	0.0	12.5	87.5
cattle	0.0	0.0	0.0	6.3	0.0	0.0	0.0	0.0	25.0	12.5	6.3	68.8
chicken	43.8	37.5	37.5	43.8	18.8	25.0	25.0	18.8	81.3	68.8	43.8	100.0
classic	0.0	12.5	6.3	25.0	0.0	0.0	0.0	0.0	50.0	18.8	12.5	43.8
comma	0.0	0.0	0.0	0.0	6.3	0.0	0.0	0.0	0.0	25.0	0.0	6.3
crown	6.3	6.3	6.3	0.0	6.3	6.3	6.3	6.3	6.3	6.3	6.3	37.5
cup	6.3	6.3	6.3	12.5	6.3	0.0	6.3	6.3	12.5	6.3	12.5	6.3
deer	31.3	43.8	31.3	56.3	43.8	37.5	31.3	37.5	62.5	68.8	50.0	87.5
device0	0.0	0.0	0.0	0.0	6.3	0.0	0.0	0.0	31.3	37.5	81.3	25.0
device2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	6.3	43.8	31.3	0.0	100.0
device3	0.0	0.0	0.0	0.0	0.0	0.0	6.3	0.0	43.8	62.5	0.0	31.3
device4	0.0	6.3	0.0	0.0	0.0	0.0	0.0	0.0	18.8	37.5	0.0	43.8
device6	0.0	0.0	0.0	0.0	0.0	0.0	12.5	0.0	0.0	18.8	0.0	0.0
device9	6.3	0.0	6.3	6.3	0.0	0.0	6.3	12.5	31.3	6.3	0.0	68.8
dog	6.3	0.0	12.5	0.0	6.3	6.3	12.5	18.8	18.8	50.0	18.8	43.8
elephant	31.3	25.0	25.0	37.5	31.3	12.5	25.0	6.3	56.3	43.8	37.5	56.3
fish	18.8	18.8	18.8	18.8	12.5	12.5	12.5	12.5	25.0	18.8	18.8	81.3
flatfish	6.3	12.5	6.3	6.3	6.3	6.3	6.3	6.3	6.3	6.3	6.3	25.0
fly	25.0	12.5	25.0	31.3	6.3	25.0	25.0	37.5	37.5	12.5	37.5	100.0
fork	12.5	12.5	31.3	18.8	18.8	12.5	18.8	12.5	50.0	31.3	43.8	62.5
frog	12.5	18.8	18.8	25.0	18.8	12.5	37.5	6.3	31.3	50.0	31.3	37.5
guitar	12.5	12.5	18.8	18.8	0.0	6.3	18.8	6.3	93.8	43.8	43.8	62.5
hammer	6.3	12.5	6.3	12.5	6.3	6.3	6.3	6.3	93.8	12.5	12.5	62.5
horse	37.5	25.0	31.3	25.0	25.0	12.5	56.3	18.8	43.8	75.0	25.0	93.8
horseshoe	0.0	0.0	0.0	0.0	0.0	0.0	12.5	0.0	0.0	18.8	0.0	68.8
jar	0.0	12.5	0.0	25.0	0.0	12.5	12.5	25.0	50.0	75.0	6.3	75.0
key	0.0	6.3	6.3	0.0	6.3	6.3	6.3	6.3	6.3	43.8	6.3	43.8
lizzard	12.5	18.8	18.8	37.5	37.5	31.3	43.8	25.0	56.3	62.5	37.5	43.8
lmfish	12.5	31.3	12.5	31.3	18.8	12.5	25.0	12.5	25.0	56.3	31.3	62.5
misk	0.0	0.0	6.3	0.0	0.0	0.0	0.0	0.0	0.0	37.5	0.0	43.8
octopus	18.8	25.0	12.5	18.8	12.5	18.8	12.5	12.5	37.5	50.0	18.8	62.5
pencil	0.0	12.5	0.0	12.5	6.3	0.0	18.8	18.8	100.0	56.3	37.5	100.0
personal_car	0.0	6.3	0.0	6.3	12.5	0.0	6.3	12.5	31.3	6.3	6.3	87.5
pocket	6.3	6.3	6.3	6.3	6.3	6.3	6.3	6.3	12.5	12.5	0.0	6.3
ray	18.8	25.0	0.0	25.0	6.3	0.0	31.3	12.5	25.0	37.5	0.0	37.5
sea_snake	12.5	31.3	12.5	31.3	25.0	18.8	25.0	43.8	37.5	18.8	25.0	81.3
shoe	0.0	0.0	0.0	6.3	0.0	0.0	0.0	0.0	12.5	12.5	0.0	62.5
spoon	31.3	68.8	43.8	50.0	37.5	62.5	43.8	87.5	100.0	75.0	68.8	87.5
spring	6.3	18.8	18.8	12.5	12.5	18.8	6.3	18.8	87.5	87.5	6.3	68.8
stef	6.3	6.3	6.3	6.3	6.3	6.3	6.3	6.3	12.5	18.8	6.3	81.3
tree	12.5	6.3	12.5	6.3	6.3	6.3	12.5	6.3	31.3	100.0	6.3	62.5
turtle	31.3	31.3	31.3	37.5	6.3	18.8	31.3	12.5	37.5	12.5	18.8	93.8
watch	6.3	12.5	6.3	18.8	12.5	6.3	6.3	25.0	12.5	6.3	6.3	68.8
avg. error	**7.3**	9.6	**8.2**	11.9	7.4	**6.9**	10.3	**9.1**	**24.9**	29.8	**12.3**	52.2

Table 5: Comparison of MPEG-7 classification average error rate per class with 4-cross-validation, comparing the methods with and without CM.

dataset	RoF RaF		RoF J48		MACP		Stacking		SVM		MLP	
	with	without	with	without	with	without	with	without	with	without	with	without
NIST	5.8	7.3	6.9	8.6	7.0	7.4	8.4	11.5	14.4	25.2	**5.2**	37.6
MNIST	5.5	8.7	6.5	8.7	8.1	10.1	8.4	9.2	14.7	19.9	**4.1**	47.1
USPS	2.9	3.7	3.7	4.6	3.7	3.2	3.5	3.5	35.4	76.3	**1.4**	44.1
MPEG-7	7.3	9.6	8.2	11.9	7.4	**6.9**	10.3	9.1	24.9	29.8	12.3	52.2

Table 6: Summary of the average error rates obtained by the ensembles in the corpus considered.

dataset	RoF RaF		RoF J48		MACP		Stacking		SVM		MLP	
	with	without	with	without	with	without	with	without	with	without	with	without
NIST	128.5 (0.5)	116.8 (0.4)	388 (4)	504 (6)	23.0 (0.1)	7.3 (0.4)	3517 (10)	2888 (12)	90 (3)	120 (4)	1650 (10)	3000 (90)
MNIST	23.5 (0.5)	38.0 (0.1)	35 (7)	150 (2)	8.3 (0.4)	4.0 (0.1)	75.3 (0.4)	265.3 (0.4)	85.0 (0.2)	297 (3)	313.7 (0.8)	7700 (90)
MPEG-7	49.3 (0.3)	26.2 (0.4)	224 (4)	97.0 (0.6)	4.3 (0.1)	2.7 (0.1)	923.7 (0.3)	436.2 (0.6)	7.2 (0.1)	15.2 (1.3)	750 (20)	2500 (60)
USPS	122 (2)	167.4 (0.6)	181 (3)	754 (4)	50 (2)	4.1 (0.1)	994 (4)	3845 (5)	73.0 (0.3)	248.1 (1.2)	291.2 (1.3)	794 (5)

Table 7: Summary of the average (standard deviation) execution time in seconds obtained by the ensembles in the experiments.

and without the CM. Specifically, a Wilcoxon 1×1 test was performed. The significance p-values considering all the experiments are shown in Table 8. These values represent the overlap between the two distributions, assuming that the classifiers are better with the CM. We can consider the p-values as a confidence measure for comparison. The significance of a low value is a high probability that the distributions compared are different.

As is shown in this table, most of the values are lower than 0.05, signifying that the use of CM significantly decreases the error rate at a confidence level of 95%. Special cases are reported for the USPS and MPEG-7 datasets using MACP and Stacking-C meta-classifiers for which the significance test yielded that the CM does not improve the accuracy although, in the opposite

dataset	RoF RaF	RoF J48	MACP	Stacking-C	SVM	MLP
NIST	**0.000004**	**0.000288**	0.404899	**0.000643**	**0.000007**	**0.000004**
MNIST	**0.007649**	**0.001312**	0.168282	0.132416	0.126279	**0.004317**
USPS	**0.0059**	**0.0043**	1 (0.796)	0.9582 (0.9582)	**0.016605**	**0.004317**
MPEG-7	**0.017337**	**0.002064**	1 (0.15168)	1 (0.399268)	**0.001516**	**0.000007**

Table 8: Wilcoxon statistical significances (p-values) reported in datasets considered with 4-cross-validation, assuming that the classifiers are better with CM than without it. The numbers in parentheses represent the opposite relationship (without CM is better than with it) when the main p-value is close to 1. The values in bold type represent a level of significance that is higher than $\alpha = 0.95$

case (values in parentheses), CM does not worsen it either. This signifies that throughout the experiments the inclusion of the CM has significantly improved, or in the worst cases maintained, the results of the meta-classifiers.

Note the good performance of the MLP classifier when CM is included, given that it is significantly better and obtains some of the lowest error rates in the entire corpora.

Table 7 also shows a summary of the average execution time in order to assess how the inclusion of the CM affects the cost of the ensembles. In general, it is clear that the MACP obtains the lowest time because it computes few calculations, while Stacking-C obtains the highest times. With regard to the results obtained when using or not using CM, there is a considerable amount of variability depending on both the corpus and the algorithm, particularly in the case of decision tree based algorithms.

5. Conclusions

A new approach with which to transform original features into a Confidence Matrix (CM) is presented. This CM consists of a set of posterior probability values which were obtained by using several weak classifiers. This approach enables the features to be transformed into a new space (meta-features), thus allowing the dimensionality of the data (in our case) to be reduced and a more meaningful value to be provided in each dimension. This is expected to help to reduce the error rate of the final classifier.

In our experimentation, 1-NN was used as a weak classifier, and several algorithms (MACP, Stacking-C, RoF RaF, RoF J48) were considered as final meta-classifiers. A 4-fold cross-validation was conducted for each of the 4 different datasets, and a statistical significance test was also applied in order to measure the effect of including the CM in a comprehensive manner. These tests reported that the inclusion of the CM can significantly improve the results of evolved meta-classifiers. In most of the cases considered, the results were either improved or remained the same. With regard to the execution time, there is no clear trend in the results (see Table 7). The inclusion of the CM decreases the execution time of the most complex ensembles considered (RoF and Stacking-C) in some corpora and increases it in others.

Although our case of study is focused on OCR classification, our intention is that the CM representation can be used in any task as long as several different feature extraction techniques can be applied to the data.. The main drawback is that it requires the user to be an active part of the process by

extracting these different sets of features. Therefore, an interesting option for future work would be to find a way to extract different groups of features automatically. A bagging system could be explored in order to obtain different weak classifiers trained with a different subset of features. The way in which RoF builds its forests also represents an idea to explore.

One of the main lines of future research is to test our CM against other meta-classification schemes. On one hand, meta-feature extraction methods and early fusion methods could be applied in order to compare its performance with CM. Special interest arises in the comparison with embedding methods [19], since its requirements are quite similar. Nevertheless, embedding methods need to tune correctly some other parameters such as dimensionality or pivot selections so a comprehensive review and experimentation would be necessary. On the other hand, there are several late fusion algorithms (such as those reported in [28, 5, 10]). It would be of great interest to compare their performance both against our proposal and in combination with it.

The goodness of including the CM could be also tested against other benchmark corpora with lower features per prototype, such as UCI Machine Learning Repository [2]. In our experiments, the original prototypes have between 360 and 490 features, while the datasets of the UCI have between 17 and 35. What is more, the features of the datasets belonging to UCI have some unknown data, and a measure of similarity other than the Euclidean distance such as the HVDM (Heterogeneous Value Difference Metric) [31] would therefore have to be used to deal with these unknown data.

Acknowledgements

This work was partially supported by the Spanish CICyT through the project TIN2013-48152-C2-1-R, the Consejería de Educación de la Comunidad Valenciana through project PROMETEO/2012/017 and a FPU fellowship (AP2012-0939) from the Spanish Ministerio de Educación Cultura y Deporte.

References

[1] Alcalá-Fdez, J., Sánchez, L., García, S., Jesus, M. J. D., Ventura, S., Garrell, J. M., Otero, J., Bacardit, J., Rivas, V. M., Fernández, J. C., and Herrera, F. (2009). KEEL: A Software Tool to Assess Evolutionary Algorithms to Data Mining Problems. *Soft Computing*, 13(3):307–318.

[2] Asuncion, A. and Newman, D. (2007). UCI Machine Learning Repository.

[3] Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

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

[5] Chang, S.-F. (2012). Robust late fusion with rank minimization. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, CVPR '12, pages 3021–3028, Washington, DC, USA. IEEE Computer Society.

[6] Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27.

[7] Duda, R. O. and Hart, P. E. (1973). *Pattern Recognition and Scene Analysis*. John Wiley and Sons, New York.

[8] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18.

[9] Hull, J. (1994). A database for handwritten text recognition research. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 16(5):550–554.

[10] Jain, A., Nandakumar, K., and Ross, A. (2005). Score normalization in multimodal biometric systems. *Pattern Recogn.*, 38(12):2270–2285.

[11] Jain, A. K., Mao, J., and Mohiuddin, K. M. (1996). Artificial neural networks: A tutorial. *Computer*, 29(3):31–44.

[12] Kim, S. and Duin, R. (2009). A Combine-Correct-Combine Scheme for Optimizing Dissimilarity-Based Classifiers. *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pages 425–432.

[13] Kittler, J., Hatef, M., Duin, R. P. W., and Matas, J. (1998). On combining classifiers. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 20:226–239.

[14] Kuncheva, L. I. (2004). *Combining pattern classifiers : methods and algorithms*. John Wiley & Sons.

[15] Latecki, L. J., Lakämper, R., and Eckhardt, U. (2000). Shape descriptors for non-rigid shapes with a single closed contour. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 424–429.

[16] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (2001). Gradient-Based Learning Applied to Document Recognition. In *Intelligent Signal Processing*, pages 306–351. IEEE Press.

[17] Oda, H., Zhu, B., Tokuno, J., Onuma, M., Kitadai, A., and Nakagawa, M. (2006). A Compact On-line and Off-line Combined Recognizer. In *Tenth International Workshop on Frontiers in Handwriting Recogntion*, volume 1, pages 133–138.

[18] Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):62–66.

[19] Pekalska, E. and Duin, R. (2005). *The dissimilarity representation for pattern recognition: foundations and applications*. World Scientific Pub Co Inc.

[20] Pérez-Cortés, J. C., Llobet, R., and Arlandis, J. (2000). Fast and Accurate Handwritten Character Recognition Using Approximate Nearest Neighbours Search on Large Databases. In Ferri, F. J., Iñesta, J. M., Amin, A., and Pudil, P., editors, *Advances in Pattern Recognition*, volume 1876 of *Lecture Notes in Computer Science*, pages 767–776, Berlin. Springer-Verlag.

[21] Quinlan, J. R. (1993). C4.5: Programs for machine learning. *Machine Learning*.

[22] Rico-Juan, J. R. and Iñesta, J. M. (2007). Normalisation of Confidence Voting Methods Applied to a Fast Handwritten OCR Classification. In Kurzynski, M., Puchala, E., Wozniak, M., and Zolnierek, A., editors, *Computer Recognition Systems 2*, number 45 in Advances in Soft Computing, pages 405–412, Wroclaw, Poland. Springer.

[23] Rico-Juan, J. R. and Iñesta, J. M. (2012). Confidence voting method ensemble applied to off-line signature verification. *Pattern Analysis and Applications*, 15(2):113–120.

[24] Rodriguez, J., Kuncheva, L., and Alonso, C. (2006). Rotation forest: A new classifier ensemble method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1619–1630.

[25] Rosenblatt, F. (1962). *Principles of neurodynamics: perceptrons and the theory of brain mechanisms*. Report (Cornell Aeronautical Laboratory). Spartan Books.

[26] Seewald, A. K. (2002). How to make stacking better and faster while also taking care of an unknown weakness. In *Proceedings of the Nineteenth International Conference on Machine Learning*, ICML '02, pages 554–561, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[27] Serra, J. (1982). *Image Analysis and mathematical morphology*. Academic Press.

[28] Terrades, O., Valveny, E., and Tabbone, S. (2009). Optimal classifier fusion in a non-bayesian probabilistic framework. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(9):1630–1644.

[29] Vapnik, V. N. (1998). *Statistical learning theory*. Wiley, 1 edition.

[30] Vellasques, E., Oliveira, L., Jr., A. B., Koerich, A., and Sabourin, R. (2006). Modeling Segmentation Cuts Using Support Vector Machines. In *Tenth International Workshop on Frontiers in Handwriting Recogntion*, volume 1, pages 41–46.

[31] Wilson, D. R. and Martinez, T. R. (1997). Improved Heterogeneous Distance Functions. *Journal of Artificial Intelligence Research*, pages 1–34.

[32] Wolpert, D. H. (1992). Stacked generalization. *Neural Networks*, 5:241–259.