International Journal of Pattern Recognition
and Artificial Intelligence
Vol. 25, No. 8 (2011) 1219–1241
© World Scientific Publishing Company
DOI: 10.1142/S021800141100910X

World Scientific
www.worldscientific.com

MULTICLASS CLASSIFICATION BASED ON META PROBABILITY CODES

NACER FARAJZADEH*, GANG PAN†,
ZHAOHUI WU‡ and MIN YAO§

College of Computer Science and Technology
Zhejiang University, Hangzhou 310027, P. R. China
*nafa@zju.edu.cn
†gpan@zju.edu.cn
‡wzh@zju.edu.cn
§myao@zju.edu.cn*

This paper proposes a new approach to improve multiclass classification performance by employing Stacked Generalization structure and One-Against-One decomposition strategy. The proposed approach encodes the outputs of all pairwise classifiers by implicitly embedding two-class discriminative information in a probabilistic manner. The encoded outputs, called *Meta Probability Codes* (MPCs), are interpreted as the projections of the original features. It is observed that MPC, compared to the original features, has more appropriate features for clustering. Based on MPC, we introduce a cluster-based multiclass classification algorithm, called MPC-Clustering. The MPC-Clustering algorithm uses the proposed approach to project an original feature space to MPC, and then it employs a clustering scheme to cluster MPCs. Subsequently, it trains individual multiclass classifiers on the produced clusters to complete the procedure of multiclass classifier induction. The performance of the proposed algorithm is extensively evaluated on 20 datasets from the UCI machine learning database repository. The results imply that MPC-Clustering is quite efficient with an improvement of 2.4% overall classification rate compared to the state-of-the-art multiclass classifiers.

Keywords: Multiclass classification; classifier; stacked generalization; decomposition; one-against-one; support vector machine; multilayer perceptron; clustering; self organizing map.

1. Introduction

Classification is the act of deciding the category of a given object based on a number of attributes related to that object. Despite the long history of classification, the research on this topic was limited in theory before 1960.[31] Alongside the progress of computers and due to new interest, automatic pattern classification has gained more attention. Automatic pattern classification employs a machine learning algorithm to

† Author for correspondence

induce a classifier given a training data set. The induced classifier then should be able to assign a predefined class label for new data from the same domain.[21]

So far, a wide variety of machine learning algorithms have been proposed for pattern classification.[28] Most of these techniques essentially involve the discrimination of only two classes such as SVM,[6] Perceptron algorithm,[15] and RIPPER.[9] However, real world applications are often not that simple and they demand the construction of classifiers capable of distinguishing multiple patterns. To generalize binary classifiers to multiclass classification problems, there are generally two techniques: solving a single optimization problem by adapting internal operations of binary classifiers,[37,39] and decomposition. Due to the fact that inducing multiclass classifiers by adapting internal operations of binary classifiers is not easy to accomplish, and in some cases it is impractical,[17,25] the decomposition technique has become more popular within the community. In the following, we give a brief review of two commonly used decomposition strategies, followed by short abstracts of the well-known alternative decomposition approaches proposed in the literature.

1.1. *Two major decomposition strategies*

1.1.1. *One-Against-All*

Perhaps the most standard method for decomposition of a multiclass classification problem into binary subproblems is the One-Against-All (OAA) strategy. In this strategy, k different binary classifiers are trained to classify k different classes, each of which separate a single class from the remaining. That is, the samples in one class are considered positive examples and the remaining samples belonging to the other classes are considered negative examples. Using the highest output value for an unknown sample, OAA reveals the corresponding class of the sample. The main disadvantage of OAA is that it may induce an inaccurate binary classifier for given classes when the data is unbalanced,[30] i.e. the number of positive examples is too low compared to the number of negative examples and vice versa.

1.1.2. *One-Against-One*

Another common and popular decomposition strategy is the One-Against-One (OAO). In this strategy, all possible pairs of different classes are taken into account and therefore $k(k-1)/2$ binary classifiers are induced, each of which separate a pair of classes. Then the final classifier is built by combining individual binary classifiers. The main drawback of OAO is as follows: if the number of training samples is not enough and the binary classifiers are not regularized carefully, the final classifier will tend to overfit.[30] The training process of the binary classifiers in this approach is simplified and needs less time compared to OAA. This is due to the fact that in OAO for each binary classifier, only the samples of two classes are considered, while in OAA all the samples are used for training binary classifiers. In this strategy, however, the number of binary classifiers grows super linearly with the number of classes.

1.2. *Alternatives to decomposition strategies*

The main issue in decomposition technique is the method of combining each binary classifier's result to produce the final result. One simple and basic solution is to use majority voting. In this solution, each of the classifiers has the same influence on the final result. To weight classifiers and define their degree of importance, one can use artificial neural networks.[15]

Stacked Generalization is a well-known technique, proposed by Wolpert,[41] to weight the outputs of the individual classifiers through a combination method rather than using a voting scheme.[32] In this technique, different (non-identical) base learners are trained using a part of the training set. Subsequently, their outputs for the remaining set of the training examples are generated. This stage is known as 1-level and the generated outputs at this stage are called meta-features. In the next stage, 2-level, a multiclass classifier called meta-learner, is trained based on the meta-features obtained from 1-level. The objective of this classifier is to learn the correct output given a certain combination of the base learners' output.[2] It has been shown that Stacked Generalization has a good generalization performance compared with individual classifiers. However, its performance decreases when the number of classes and the dimension of the feature space increase.[24]

Another way of combining the results of individual binary classifiers is to use Decision Directed Acyclic Graph (DDAG) architecture.[26] This algorithm reduces a multiclass problem to a set of two-class classifiers at each node by organizing them in a tree structure. Thus, an unknown sample is evaluated at each node, and depending on the result at each node, the sample traverses the tree until a solution is obtained. This approach has some disadvantages that were pointed out by Kijsirikul and Ussivakul.[19] The result of the final classifier in DDAG depends on the sequence of the binary classifiers in the nodes of the graph. Therefore, different permutations of the nodes may produce different results affecting the reliability of the final classifier. Additionally, the number of evaluations depends on the position of the true class in the graph, which in turn increases the cumulative error.[19]

Dietterich and Bakiri[11] introduced the Error Correcting Output Coding (ECOC) approach to combine the output of binary classifiers. They proposed employing k binary classifiers to produce a binary pattern of length k, so called code-word, and applying an exhaustive method to find optimal code-words to assign to each class. For a given unknown sample, its code-word is generated first and then is compared with the preassigned code-words. The closest preassigned code-word to the sample's code-word, in terms of Hamming distance, reveals the sample's class. In this approach, to possess a good error-correction capability, preassigned code-words should be well separated from each other. Additionally, there should be no correlation between any two bits in a column.[14] While this method has demonstrated a good performance in pattern recognition problems, it has been pointed out in Ref. 18 that this approach is an NP-complete problem.

Although decomposition techniques and their alternatives have been very prominent in the literature, there are some heuristic and interesting methods proposed

to improve the classification accuracy and overcome the aforementioned disadvantages (see Refs. 29 and 38 and references therein). Recently, Mehrotra *et al.*[22] have introduced the idea of classification based on clustering. In their method, which is typically well suited for problems with a large feature set, the training samples are clustered first based on the selected features among the available features.[36] Then, individual multiclass classifiers are trained on each cluster. For a given sample, its cluster is determined first and then the corresponding classifier is used to classify it.

Cluster-based classification approach can improve the classification performance by squeezing out the last drop.[22] However, we believe that there are yet more drops that can be squeezed by improving the clustering step of the cluster-based classification. In this paper, we propose a new approach to improve multiclass classification performance by employing Stacked Generalization structure and OAO decomposition strategy. The proposed approach encodes the outputs of all the pairwise classifiers by implicitly embedding two-class discriminative information in a probabilistic manner. The encoded outputs are interpreted as the projections of the original features. The motivation behind our approach is to search an optimal transformed feature space that can outperform the original feature space in terms of clustering and improve the multiclass classification performance.

The performance of our proposed algorithm is evaluated by applying it on 20 different datasets from the UCI machine learning database repository.[35] It is shown that our algorithm improves the classification rate by almost 2.4% on average. Moreover, the performance of the projected features is also evaluated without applying a clustering step. That is, a known multiclass classifier is trained directly on the projected samples. It is shown that the classification accuracy of SVM and kNN trained on the projected features improved by 0.99% and 3.62%, respectively.

The rest of this paper is organized as follows. Section 2 introduces our approach for projecting the original feature's space to a new feature space; Sec. 3 presents an algorithm for multiclass classifier induction based on the proposed projection; experimental results are given in Sec. 4; and Sec. 5 is the conclusion.

2. Meta Probability Code

In this section, we aim to introduce a novel approach to project the original feature space to a new feature space. The basic idea of the proposed approach is established based on Stacked Generalization structure. Therefore, there are base learners from Stacked Generalization in our approach as well. To make the proposed projection approach compact, the base learners in our scheme are chosen to be identical; whereas in the original idea of Stacked Generalization, the base learners were not identical.

Given k classes $\{C_i, i = 1, \ldots, k\}$, and a training sample set $\mathbf{X} = \{(\mathbf{x}_i, y_i), i = 1, \ldots, l\}$, where $\mathbf{x}_i \in \mathbb{R}^N$ is the ith sample, $y_i \in \{1, \ldots, k\}$ is the class label of the ith sample, and l is the number of samples, $K = k(k-1)/2$ pairwise binary classifiers

(i.e. base learners) are trained according to the OAO strategy:

$$h^B_{r,s}(\mathbf{x}_i) = \begin{cases} 1 & \text{if } \mathbf{x}_i \in C_r \\ 0 & \text{if } \mathbf{x}_i \in C_s \end{cases} \quad \text{for } r = 1, \ldots, k-1, \quad \text{and} \quad s = r+1, \ldots, k, \tag{1}$$

where the superscript B indicates that h^B is a binary classifier.

All the binary base learners can be trained using the training set. Our goal is to train the base learners and build a projection function accordingly. The outputs of the base learners are concatenated in order to form a new feature vector $\mathbf{t}$:

$$\begin{aligned} \mathbf{t} &= f(\mathbf{x}), \\ f(\mathbf{x}) &= \oplus h^B_{r,s}(\mathbf{x}), \quad f : \mathbb{R}^N \to \mathbb{Q}^K, \end{aligned} \tag{2}$$

where N and K are the dimensions of the original data space and the projected data space, respectively.

The output of h^Bs can either be the class probability (real-valued output) or the class prediction (binary-valued output). Since the class probability produces a smoother confidence measure compared to the class prediction,[33] in the proposed approach we consider the class probabilities for h^Bs as our primary choice, which is indicated by h^{Bp}. Nevertheless, the class prediction (h^{Bb}), is also considered in our work and its performance is evaluated. We call $\mathbf{t}$ *Meta Probability Code* (MPC) if the base learners are considered to be h^{Bp}, and *Meta Binary Code* (MBC) if the base learners are considered to be h^{Bb}. That is:

$$MPC(\mathbf{x}) = \oplus h^{Bp}_{r,s}(\mathbf{x}) \quad \text{where } h^{Bp}_{r,s}(\mathbf{x}) = p(r|\mathbf{x}), \tag{3}$$

$$MBC(\mathbf{x}) = \oplus h^{Bb}_{r,s}(\mathbf{x}) \quad \text{where } h^{Bb}_{r,s}(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \in C_r \\ 0 & \text{if } \mathbf{x} \in C_s. \end{cases} \tag{4}$$

It should be noted that to generate MPCs, we only use the probability of being a member of class r, $p(r|\mathbf{x})$, rather than using both of the probabilities.

3. MPC-Based Algorithm

In this section a cluster-based multiclass classification algorithm based on MPC is introduced.

3.1. *MPC-Clustering overview*

In the MPC-Clustering (MPCC) algorithm, both the training and testing procedures consist of three steps.

3.1.1. *Training*

The first step deals with projecting original samples to MPCs [Eq. (3)]. The second step clusters the MPCs, and the final step trains individual multiclass classifiers on the produced clusters, i.e. the classification task is localized and limited on each cluster.

Let us assume that for a given projected samples' set $\{(\mathbf{t}_i, y_i), i = 1, \ldots, l\}$, where $\mathbf{t}_i \in \mathbb{Q}^K$ is the ith projected sample, S clusters (partitions) $\mathcal{L} = \bigcup_{s=1}^{S} \ell_s$ are produced (the elements of ℓs are pairwise disjoint). An optimal multiclass classifier h_s^M is trained on cluster $\ell_s = \{(\mathbf{t}_j, y_j), j \in \mathcal{N}_s\}$, where $\mathcal{N}_s$ is the set of samples' indexes in cluster ℓ_s, such that:

$$h_s^M(\mathbf{t}_j) = y_j. \tag{5}$$

The superscript M indicates that h^M is a multiclass classifier. In the next subsection, we will describe a criterion that should be taken into account for clustering.

3.1.2. *Testing*

Given an unknown sample, its MPC is generated first. Then, the cluster the MPC belongs to is determined. Finally, the class label of the sample is obtained using the multiclass classifier of the corresponding cluster.

Figures 1(a) and 1(b) show the training and the testing procedures of MPCC algorithm, respectively.

3.2. *Cluster post-processing*

It is obvious that in a given dataset, if the samples of some categories are very similar or the number of clusters is considered to be large, any clustering scheme may produce some clusters which contain only samples of one category, called *mono-cluster*. Since in the last step of the proposed algorithm individual multiclass classifiers, h^Ms, are trained on the produced clusters, we should be aware of mono-clusters, otherwise the algorithm will face difficulty when it is trying to train h^Ms. Therefore, to avoid having mono-clusters, we should consider a criterion in the clustering step of the proposed algorithm:

$$\forall\, \ell_i \in \mathcal{L}, \quad \psi(\ell_i) > 1, \tag{6}$$

where $\psi(.)$ is the number of different categories (classes) of the elements (samples) in a given set.

The criterion proposed in (6) is implemented via an aggregating procedure where all the produced mono-clusters are joined to their closest clusters. Note that even after joining a mono-cluster to its closest cluster, we may have another bigger mono-cluster. Thus, we repeat the procedure until no mono-clusters exist. This procedure is called *cluster post processing* and is shown in Fig. 2.

3.3. *Toy example*

To demonstrate how MPCC works, we employ a toy dataset containing 51 samples of three classes; A, B and C. The 2D scatterplot of the dataset is shown in Fig. 3.

The first step in MPCC is to project the original feature vectors to MPCs. To this end, three pairwise binary classifiers, $h_{A,B}$, $h_{A,C}$ and $h_{B,C}$, are induced according to

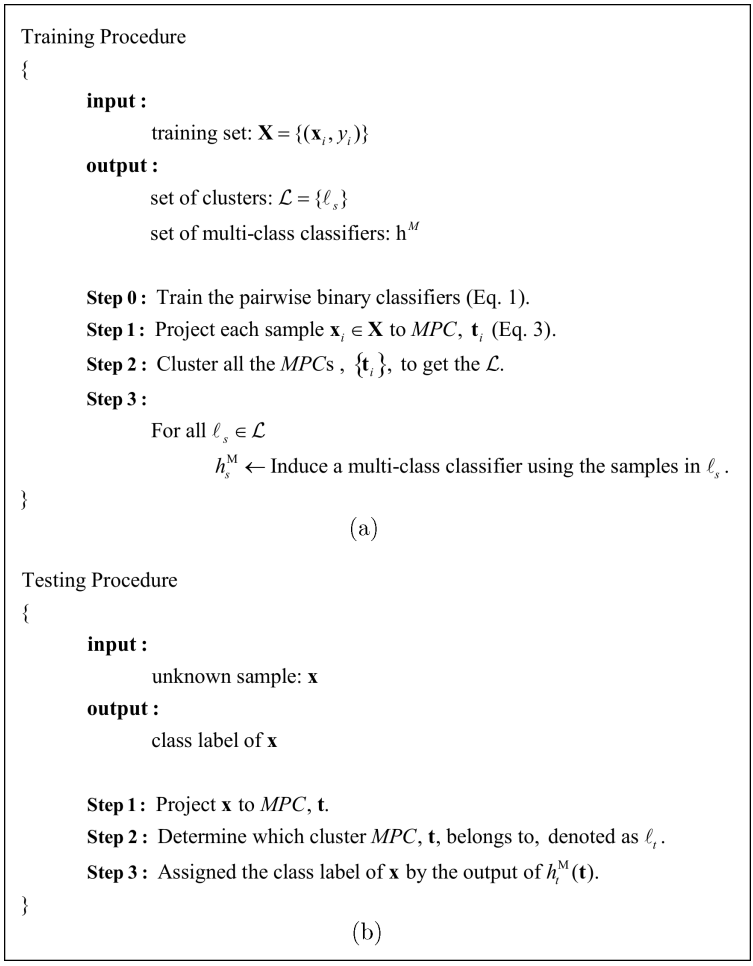

Fig. 1. MPC-Clustering algorithm. (a) Training procedure, (b) testing procedure.

Eq. (1).[a] Following Eq. (3), we feed each of the base learners with every sample in the toy dataset. Hence, for a given sample in the dataset, the real-valued outputs (class probabilities) of $h_{A,B}$, $h_{A,C}$ and $h_{B,C}$ together produce a new three-dimensional feature vector (MPC). Figure 4 shows a 3D scatterplot of the MPCs. It indicates that the MPCs are linearly separable while the original features, as can be seen from Fig. 3, are not.

The second step is to cluster the projected samples. Here, for simplicity, we use k-means clustering scheme where the value of k is set to 2. The produced clusters are also shown in Fig. 4 and are indicated by black ellipses. Applying the cluster post-processing procedure, the final clusters are produced. Since there are no

[a] The base learners are chosen to be Support Vector Machine. For more detailed information see Sec. 4.2.

```
Cluster Postprocessing Procedure
{
        input :
                raw cluster set: L = {ℓ1, ℓ2, ...}

        for all ℓi ∈ L
        {
                while(ℓi is a mono–cluster)
                {
                        ℓj ← Find closets cluster to ℓi .
                        ℓi ← Join ℓi and ℓj .
                        delete ℓj .
                }
        }
}
```

Fig. 2. Cluster post-processing procedure.

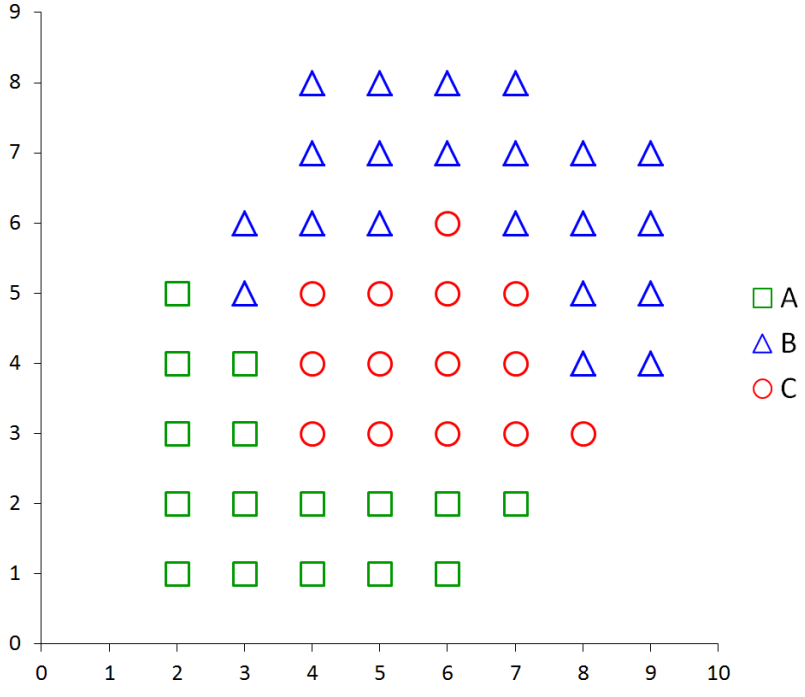

Fig. 3. 2D scatterplot of the toy dataset.

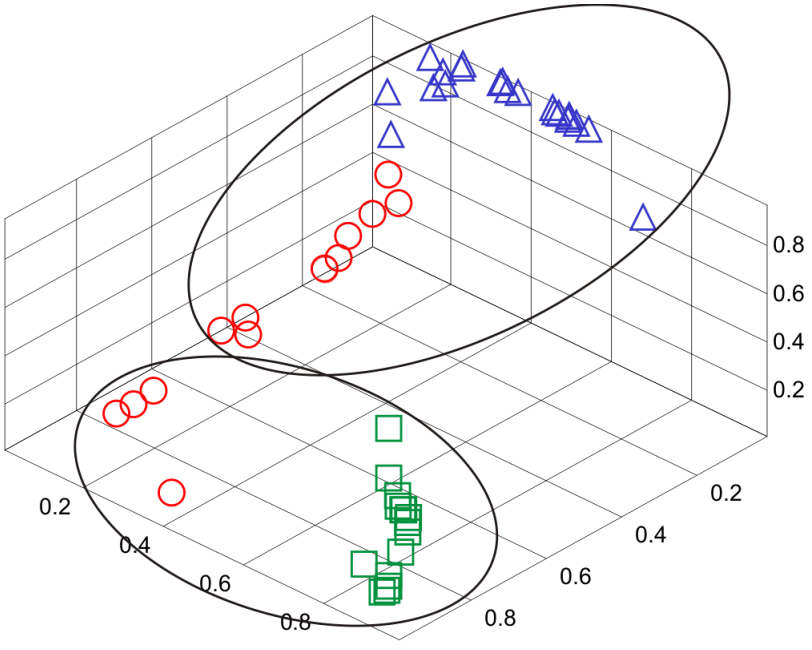

Fig. 4. 3D scatterplot of the toy dataset after projecting its features to MPC. Final clusters are indicated by black ellipses.

mono-clusters produced, the cluster post-processing procedure will not change any of the clusters.

To finalize MPCC, any known multiclass classification algorithm can be used to induce h^Ms [Eq. (5)].

4. Experiments and Results

4.1. *Datasets*

To investigate the performance of the proposed algorithm for multiclass classification, we conducted experiments on 20 datasets from UCI.[35] Table 1 shows a brief description of these datasets. The chosen datasets are from different categories with different levels of difficulty, which represent a wide range of domains and data characteristics. Meanwhile, we choose those datasets that have more than two types of patterns to be classified ($k > 2$). The entries that contain missing values are not considered in our experiments.

4.2. *Employed classifiers and clustering schemes*

To induce binary and multiclass classifiers in the experiments, we employ three classifiers from different categories: SVM,[6] Multi-Layer Perceptron,[8,15] and k-Nearest

Table 1. Description of the datasets used in the experiments.

No.	Dataset	No. of Classes	No. of Features	Missing Value	No. of Instances			No. of Train	No. of Test
					Min	Max	Total		
1	Abalone	3	8	NO	—	—	4177	3133	1044
2	Car	4	6	NO	65	1210	1728	—	—
3	Chess (King vs. King)	18	6	NO	27	4553	28056	—	—
4	Dermatology	6	34	YES	20	112	358	—	—
5	Glass	6	9	NO	9	76	214	—	—
6	Heart Disease Cleveland	5	13	YES	13	160	297	—	—
7	Iris	3	4	NO	50	50	150	—	—
8	Letter Recognition	26	16	NO	734	813	20000	—	—
9	Mfeat.FOU	10	76	NO	200	200	20000	—	—
10	Mfeat.MOR	10	6	NO	200	200	20000	—	—
11	Mfeat.ZER	10	47	NO	200	200	20000	—	—
12	Nursery	5	8	NO	2	4320	12960	—	—
13	Page Blocks	5	10	NO	28	4913	5473	—	—
14	Pen Digits	10	16	NO	719	780	7494	7494	3498
15	Sat Image	6	36	NO	215	1038	4435	4435	2000
16	Segment	7	19	NO	30	30	210	210	2100
17	Soybean	15	35	YES	10	40	291	291	341
18	Vehicle	4	18	NO	199	218	946	—	—
19	Wine	3	13	NO	48	71	178	—	—
20	Yeast	10	8	NO	5	463	1484	—	—

Neighbor. For clustering purposes, two clustering schemes, SOM[20] and k-means, are tested. In the following, a brief description of each of the classifiers and clustering schemes, along with their properties used in our experiments, are presented.

4.2.1. *Support vector machine*

Given a set of training sample pairs $(\mathbf{x}_i, y_i)$ where $\mathbf{x}_i \in \mathbb{R}^N$ and $y \in \{-1, +1\}$, the solution of the following optimization problem is required in SVM:

$$\begin{aligned} &\min_{w,b,\xi} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}\xi_i \\ &\text{subject to} \quad y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \end{aligned} \tag{7}$$

where l is the number of samples, C is the penalty parameter and ϕ is a kernel function. The kernel function maps training vectors to a higher dimension with the hope that there will be a linear separating hyperplane with the maximal margin. The radial basis function, also known as Gaussian function, is a commonly used kernel function and is as follows:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2), \quad \gamma > 0, \tag{8}$$

where γ is the kernel parameter.

Once the optimization problem in Eq. (7) is solved, the class of a given unknown sample $\mathbf{x}$ is determined with the following decision function:

$$\text{class of } \mathbf{x} \equiv \arg \max_{i \in \{-1,+1\}} ((\mathbf{w}^i)^T \phi(\mathbf{x}) + b^i). \tag{9}$$

In this paper, for SVMs' kernels, Gaussian function is chosen and the parameter selection is done based on grid optimization strategy.[16] That is, for a given problem, the generalization accuracy using different kernel parameters $\gamma = [2^{-5}, 2^{-4}, \ldots, 2^5]$, and cost parameters $C = [2^{-5}, 2^{-4}, \ldots, 2^{10}]$ are estimated. Thus, $11 \times 16 = 176$ combinations are tried to find the optimum parameters. Note that the parameter optimization is only done on the training samples via five-fold cross validation. The publicly available implementation of SVM, *libsvm*,[7] is employed. To produce the probability outputs, we also use the provided routines in *libsvm*. The routines have been implemented based on the work proposed by Wu *et al.*[42]

The second step in MPCC algorithm partitions all the given training samples into clusters. These clusters obviously will contain only a fraction of the classes and samples. Therefore, to simplify MPCC algorithm we consider the use of single machine approach for induction of h^Ms. To this end, an implementation of the work proposed in Ref. 10 (multiclass SVM), *bsvm*,[5] is used in our experiments.

4.2.2. *Multi-layer perceptron*

The Multi-Layer Perceptron (MLP)[8,15] is an artificial neural network which consists of more than one layer. The outputs of each layer are connected to one or more of the inputs of the next layer. The technique which MLP employs for training the network is called back-propagation.[23] Two main activation functions in this network are both sigmoid and are as follows:

$$\theta(y_i) = \tanh(v_i), \tag{10}$$

$$\theta(y_i) = (1 + e^{-v_i})^{-1}, \tag{11}$$

where y_i is the output of the ith node and v_i is the weighted sum of the input nodes.

The first function [Eq. (10)] ranges from -1 to $+1$ and is a tangent hyperbolic. The second function [Eq. (11)] is called logistic sigmoid and ranges from 0 to 1. The logistic sigmoid function, which is equal in shape with the tangent hyperbolic, allows the outputs of MLP to be interpreted as an estimated probability of the form $p(target = 1|\mathbf{x})$.[4]

In this paper, we use a three layer MLP. The second layer (hidden layer) uses tangent sigmoid and the last layer (output layer) uses logistic sigmoid in order to serve probability outputs of the base learners [Eq. (3)]. The initial weights are given according to the Nguyen−Widrow algorithm and the training is done based on the Levenberg−Marquardt Conjugate back-propagation algorithm. The number of hidden layer nodes is tuned and done by varying it from a tenth to a half of the number of features in steps of fifths.

4.2.3. *k-nearest neighbor*

The k-Nearest Neighbor (kNN) algorithm is one of the simplest classification algorithms in pattern recognition. In this algorithm, the label of an unknown sample is assigned by a majority vote of its neighbors. In other words, the class which is the most common class among the k nearest neighbors of a given sample's neighborhood is determined as the sample's class.

In our work, for kNN algorithm, Euclidean measure is used for distances, and tuning the value of k is done with values ranging from 1 to 10.

4.2.4. *Self organizing map*

One of the most popular neural network models is the Self-Organizing Map (SOM)[20] which belongs to the category of competitive learning networks. The training procedure in this network is unsupervised. Therefore, SOM is very suitable for clustering the data of which a little information about its characteristics is available.

In this paper, a two-dimensional structure for SOM is used, i.e. any feature vector from a high dimensional space is mapped to a 2D space. The size of the SOM network is chosen to be 200×200, and the Euclidean distance is used as a distance measure.

4.2.5. *k-means*

The k-means is a well-known clustering scheme in which it partitions samples into one of k groups. The partitioning procedure is iterative and it tries to minimize the overall within-cluster scatter by reallocating clusters' members. The value of k is chosen prior to the partitioning procedure. In this study, the proper number of partitions, k, is selected by cross validation.

4.3. *Effectiveness of the proposed approach*

Cluster-based multiclass classification algorithm tries to localize the classification task by furthering a clustering step and training individual multiclass classifiers based on the produced clusters. Therefore, as it should, the clustering step plays the most important role in this algorithm. The effectiveness of this step can be examined from two aspects: (1) the effectiveness of the clustering scheme itself, and (2) the effectiveness of the features being clustered. The former aspect was discovered by the study of Abbas[1] where k-means clustering, Hierarchal clustering, Self Organizing Map, and Expectation Maximization (EM) clustering schemes were compared from different points of view. However, in the original idea of cluster-based classification,[22] the authors used k-means. They mentioned that the classification results obtained employing k-means and SOM were similar, and the only reason that they selected k-means was due to its simplicity. Nevertheless, the effectiveness of these two clustering schemes for MPCC is examined and presented in Sec. 4.4.

In this section, our hope is to evaluate the effectiveness of the projected features MBCs and MPCs (Eqs. (3) and (4)), and compare them with the original features in

order to study the second aspect. The technique which we use to evaluate the produced clusters is the classes-to-clusters technique.[40] In this technique, after clustering with a clustering scheme, the majority class in each cluster is determined and its label is assigned to that cluster with the constraint that the label of a class can only be assigned to one cluster. Subsequently, all the instances are mapped to the labeled clusters and the number of correctly mapped instances is recorded.

In order to generalize and to fairly compare the performance of the features of interest for clustering, it is important to choose a proper number of clusters for a given problem. One solution is to run cross validation on a randomly drawn fraction of the dataset and find a proper number of clusters first, and then use it for the remaining (testing) samples.[12] To this end, in our experiment, the samples of every class are divided into two parts in a random manner. Subsequently, the first parts are collected as the training set, and the second parts are left for the evaluation purpose.

During the experiments, however, we observed that setting the number of clusters equal to the number of classes is an optimal choice for classes-to-clusters evaluation technique. Since for our case, in particular, the categories of instances for a given problem are known, we take this advantage and set the number of clusters equal to the number of classes for simplicity. In this experiment, the base learners and clustering scheme are chosen to be SVM and k-means respectively. Table 2 shows the results.

Table 2. Percentage of correctly mapped samples in the classes-to-clusters evaluation technique for the original and the projected features.

Dataset	Original Features	MPC	MBC
Abalone	52.36	65.87	68.40
Car	47.52	97.40	87.79
Chess	41.12	75.16	49.13
Dermatology	73.76	77.94	71.79
Glass	47.45	65.12	53.49
HDC	45.11	44.19	43.78
Iris	88.67	96.67	97.34
LR	72.64	76.11	81.41
Mfeat.FOU	54.65	68.45	69.35
Mfeat.MOR	62.36	66.55	58.82
Mfeat.ZER	62.20	68.05	56.00
Nursery	26.91	71.02	60.09
Page Blocks	39.14	41.79	49.36
Pen Digits	67.58	74.14	68.09
Sat Image	66.84	70.81	73.87
Segment	57.62	61.91	51.91
Soybean	60.69	78.97	68.28
Vehicle	34.99	75.89	74.34
Wine	94.39	97.73	94.39
Yeast	38.19	35.05	36.39
Average	56.71	70.44	65.55

As can be seen from Table 2, the percentage of the correctly clustered instances based on MPC 13.73% on average is higher than the original features. The effectiveness of MPC can be explained as follows: the MPC contains between-class discriminant information in which each of its components represents the probability of the corresponding sample being a member of a given pair. Therefore, it is more likely that the outputs of the pairwise binary classifiers (base learners) for all samples of a class are similar. As a result, the generated MPCs of a given class can be clustered well.

Table 2 also shows that the results obtained using the original features for two datasets, HDC and Yeast, are slightly better than MPCs, and for Page Block dataset the results are almost the same. We think that this is due to very few numbers of samples available for some classes in these datasets (see Table 1 and the datasets' descriptions from Ref. 35). Therefore, the base learners may not be trained accurately, and, as a consequence, the derived features based on them may not be appropriate enough for clustering as we had hoped for. However the number of samples for one of the classes in Nursery dataset is 2, the evaluation result based on MPC is 44.11% higher than the original features in this dataset. This is not in contrary to what we have concluded. As this class forms only 0.01% of the dataset, thus, its effect either in clustering or classification is obviously very low and negligible.[b]

In the following section the effectiveness of MPC and MBC is compared and discussed.

4.3.1. *MPC versus MBC*

As Table 2 indicates, the average evaluation result based on MBC is 8.84% higher than the original features and is close to MPC. It seems that projecting features using MBC, which consists of 1s and 0s, generates more distinguishable patterns compared to the original features. However, we cannot expect that using MBC will increase the classification accuracy more than the original features and probably more than MPC. This is due to the fact that the projected samples (MBCs) from different classes may overlap very closely. To demonstrate this problem, we use Wine dataset, which contains three different classes ($k = 3$). The proposed approach projects the original feature space from dimension $N = 13$ to dimension $K = 3 \times (3 - 1)/2$. Hence, we can plot a 3D scatter of the projected samples and illustrate how MPCs and MBCs perform the projection. Figure 5 shows 3D scatter plots of the MPCs and MBCs. The demonstrated plots data are drawn from a complete run of 10-fold cross validation.

From Fig. 5(a) it can be seen that some of the MBCs of class B are exactly overlapped with the MBCs of class C. On one hand, we can take advantage of this projection and produce very isolated clusters. On the other hand, we may face a serious problem while training h^Ms [Eq. (5)]. That is, the multiclass classifier in the

[b]The Nursery dataset consists of five different classes, each of which contain 4320, 2, 328, 4266 and 4044 samples respectively (see Ref. 35).

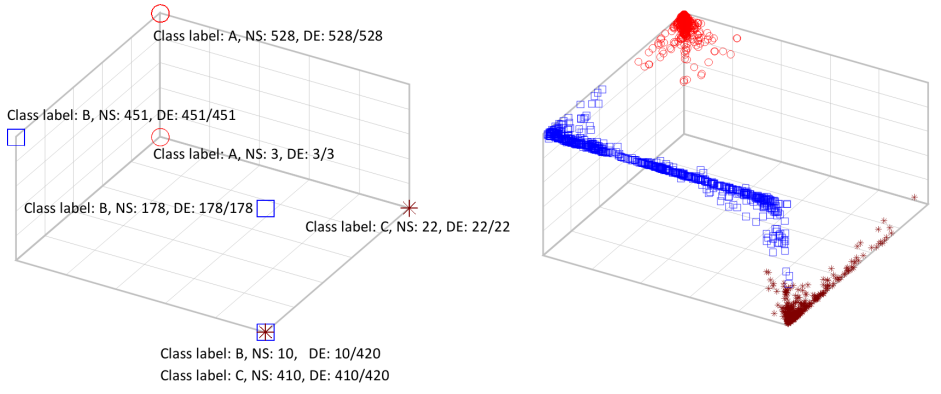

Fig. 5. 3D plots of the projected samples. (a) MBCs, (b) MPCs. NS: Number of samples, DE: Density of the overlapped samples.

last step of the algorithm will not be able to distinguish these samples as two different samples in either the training procedure or the testing procedure. Therefore, the classification rate may drop considerably. As Fig. 5(b) shows, there are no exact overlaps among the samples of the different classes for MPCs. Moreover, as can be seen, the projected samples are very easy to be clustered and for this dataset, in particular, are linearly separable.

4.4. *Classification results*

4.4.1. *Experimental framework and protocol*

To evaluate the performance of MPCC algorithm, SVM and MLP are used for the induction of the base learners. For the clustering step, SOM and k-means are employed. Hence, we conduct four experiments as follows: (1) base learners: SVM, clustering scheme: SOM, (2) base learners: SVM, clustering scheme: k-means, (3) base learners: MLP, clustering scheme: SOM, and (4) base learners: MLP, clustering scheme: k-means. To evaluate MBC-Clustering (MBCC), we conduct only one experiment where the base learners and clustering scheme are chosen to be SVM and SOM respectively. Note that for h^Ms, as it is mentioned in Sec. 4.2, we use multiclass SVM.

To investigate the performance of MPC in terms of classification, we conduct another experiment using the same datasets, where SVM and kNN are trained directly on MPCs, i.e. no clustering step is applied. Hereafter, we refer to these classifiers as MPC-Direct (MPCD).

In the experiments for the datasets in which the training and testing sets have already been partitioned, we use them accordingly. For the other datasets in which no training and testing sets have been provided, we use 10-fold cross validation. That is, the entire given samples are randomly partitioned into ten subsets, which are as

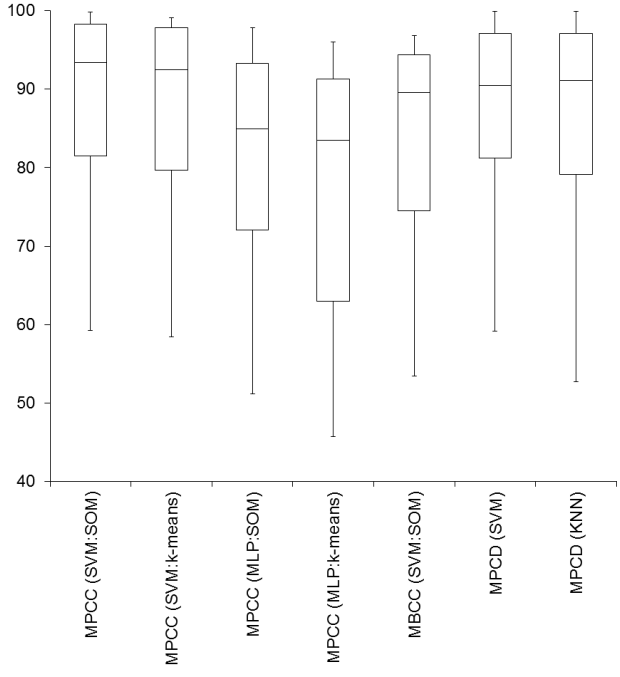

Fig. 6. Box-plot of the classification rates obtained with MPCC and MPCD algorithms.

closely as possible equally sized. Then we run the algorithm ten times in which at each run nine subsets are used for training and one set is left for testing.

In order to decrease any random effects of one single run, all the demonstrated results in this section are averages of ten runs of the proposed algorithms.

4.4.2. *Results*

Table 3 shows the classification results for MPCC, MBCC and MPCD algorithms on 20 different datasets. The box-plot of the results is also provided and demonstrated in Fig. 6.

As can be seen from Table 3, the average classification rate for the SVM:k-means pair is 87.18%, which is less than the classification rate obtained by the SVM:SOM pair (88.11%). Additionally, comparing the classification rates of MLP:SOM (81.30%) and MLP:k-means (77.57%) pairs, we arrive at the conclusion that the performance of SOM clustering scheme is better than k-means. We think that the reason for the effectiveness of SOM is due to its totally unsupervised algorithm, where for k-means one needs to adjust the number of clusters beforehand. From the classification point of view, we can conclude that the performance of SVM is

[c]In Fig. 7, CBC[22] is not included due to the low number of its reported results.

Table 3. Classification rates (%) for MPC-Clustering, MBC-Clustering and MPC-Direct algorithms.

	MPCC				MBCC	MPCD	
Dataset	SVM:SOM	SVM:*k*-Means	MLP:SOM	MLP:*k*-Means	SVM:SOM	SVM	*k*NN
Abalone	74.14	71.45	66.49	63.14	61.44	69.94	62.91
Car	99.65	99.07	95.14	91.10	94.28	99.61	99.47
Chess	90.63	89.52	81.45	62.19	85.91	88.01	86.12
Dermatology	97.56	97.51	92.11	87.12	95.94	97.52	98.33
Glass	73.39	73.12	61.98	51.19	68.73	72.51	73.16
HDC	58.29	58.46	51.19	45.74	53.46	59.37	57.37
Iris	96.67	96.66	94.00	93.33	96.00	96.21	95.33
LR	98.23	97.98	92.81	89.34	93.94	97.11	97.37
Mfeat.FOU	85.15	82.80	74.72	72.52	80.07	85.65	85.50
Mfeat.MOR	78.87	75.80	64.19	62.39	72.25	74.10	71.60
Mfeat.ZER	82.41	81.00	73.94	70.41	75.25	83.63	83.45
Nursery	99.81	98.62	91.59	87.37	94.81	99.80	99.90
Page Blocks	97.99	96.38	93.64	93.19	92.71	96.71	97.02
Pen Digits	98.89	98.99	97.83	94.19	93.29	97.82	98.25
Sat Image	91.45	91.10	85.75	84.95	89.25	90.75	91.75
Segment	98.41	97.80	93.19	92.04	96.24	93.23	93.14
Soybean	95.34	93.82	84.11	82.05	89.92	90.29	91.47
Vehicle	87.19	86.50	81.92	80.17	83.05	85.92	85.79
Wine	99.55	97.80	96.62	96.07	96.83	97.15	97.70
Yeast	58.48	59.25	53.43	52.98	55.39	59.16	59.59
Average	88.11	87.18	81.30	77.57	83.43	86.73	86.26

obviously better than MLP, since the average classification rate obtained using the MLP:SOM pair is 6.80% less than the SVM:SOM pair.

Table 3 also shows that the classification results obtained using MBCs, as it was expected, are lower than those of MPCs, and on average are 4.7% below them.

4.5. *Comparison with other multiclass classifiers*

To compare the performance of our proposed algorithms with other multiclass classification algorithms, we have collected the reported classification rates of different algorithms with our selected datasets. Table 4 shows the comparisons between the proposed algorithms and other algorithms. The best rates are bold-faced. The box-plot of the recognition rates is also shown in Fig. 7.[c]

The authors of cluster-based classification (CBC) provided their classification accuracy on eight different datasets, in which only three of them were multiclass classification problems. Since in this work we put our emphasis on multiclass classification problems, we can quote only results from their paper of Abalone, Letter Recognition and Nursery datasets. The average classification rate on Abalone, Letter Recognition and Nursery datasets for MPCC and CBC are 90.72% and 90.18% respectively. This implies that our proposed algorithm outperforms CBC on these three datasets.

[c] In Fig. 7, CBC[22] is not included due to the low number of its reported results.

Table 4. Comparison of the classification rates (%) with other multiclass classification algorithms.

Dataset	MPCC (SVM:SOM)	MPCD (SVM)	MPCD (kNN)	CBC[22]	kNN[3]	FM3[3]	RM[3]	SVM[3]	RBFNN[3]	SVM[27]	(R^3)[13]
Abalone	**74.14**	69.94	62.91	72.97	63.07	66.69	66.45	66.46	66.49	—	—
Car	**99.65**	99.61	99.47	—	94.76	93.03	87.57	98.76	93.63	99.59	97.80
Chess	**90.63**	88.01	86.12	—	74.30	39.80	31.91	81.33	—	—	—
Dermatology	97.56	97.52	**98.33**	—	96.50	—	96.78	97.51	65.10	—	—
Glass	73.39	72.51	73.16	—	66.94	67.64	63.98	68.64	68.75	71.96	**73.80**
HDC	58.29	**59.37**	57.37	—	56.45	46.36	56.30	57.50	54.68	—	—
Iris	96.67	96.21	95.33	—	95.68	**96.79**	96.70	96.19	96.03	—	—
LR	**98.23**	97.11	97.37	97.82	95.70	92.92	73.53	97.56	—	97.88	92.30
Mfeat.FOU	85.15	**85.65**	85.50	—	81.13	—	82.87	84.25	59.84	—	—
Mfeat.MOR	**78.87**	74.10	71.16	—	71.58	74.90	74.74	74.95	74.12	—	—
Mfeat.ZER	82.41	**83.63**	83.45	—	81.12	—	83.58	82.71	80.91	—	—
Nursery	99.81	99.80	**99.90**	99.75	97.25	95.96	90.98	99.58	—	—	—
Page Blocks	**97.99**	96.71	97.02	—	95.96	95.97	95.22	96.68	95.88	96.62	97.40
Pen Digits	98.89	97.82	98.25	—	99.38	99.56	95.76	**99.61**	—	—	—
Sat Image	91.45	90.75	91.75	—	90.70	—	88.04	**92.24**	—	91.85	87.80
Segment	**98.41**	93.23	93.14	—	96.18	95.94	92.52	96.75	95.09	97.53	—
Soybean	**95.34**	90.29	91.47	—	72.48	74.85	78.27	80.93	72.44	93.62	86.70
Vehicle	87.19	85.92	85.79	—	69.81	85.52	82.95	84.86	75.20	**87.47**	—
Wine	**99.55**	97.15	97.70	—	96.12	96.90	96.67	98.38	98.12	98.87	—
Yeast	58.48	59.16	59.59	—	57.88	59.64	59.76	59.84	**59.99**	58.96	58.40

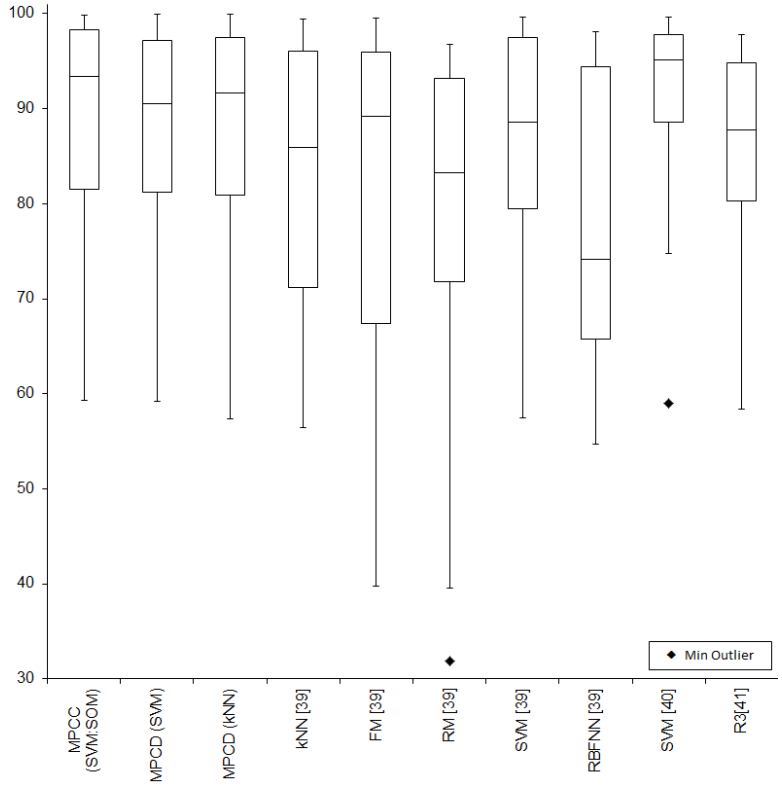

Fig. 7. Box-plot of the classification rates obtained with proposed algorithms and other algorithms.

As Table 4 shows, classification rates for all the datasets are only available for three algorithms; kNN,[3] RM[3] (for more information about RM see Ref. 34) and SVM.[3] The average classification rates for these algorithms are 82.64%, 79.72% and 85.73% respectively. Compared to MPCC, with an average classification rate of 88.11% on all datasets, we can see that MPCC algorithm outperforms these algorithms and surpasses the classification rate by almost 2.4%.

According to Table 4, an interesting conclusion can be made by comparing the results of MPCD(SVM) and MPCD(kNN) with SVM[3] and kNN[3]; applying SVM and kNN on the original features yield classification accuracies of 85.73% and 82.64% respectively. By contrast, the classification rates on the projected features (MPCs) for these algorithms are 86.72% and 86.26%. These results obviously show the effectiveness of MPC compared to the original features with an improvement of 0.99% in SVM and 3.62% in kNN.

From Table 4 it is also observed, however, that MPCD is a single stage algorithm compared to MPCC; its classification accuracy is close to MPCC, and, on average MPCD(SVM) is 1.38% lower than MPCC(SVM:SOM). Furthermore, we can see that MPCD(SVM), together with MPCD(kNN) outperforms MPCC(SVM:SOM) on

seven datasets: Dermatology, HDC, Mfeat.FOU, Mfeat.ZER, Nursery, Sat Image, and Yeast.

If we compare the classification rates individually with each other, it can be seen that MPCC algorithm obtains the best classification rates (except the results of MPCD(SVM) and MPCD(kNN)) in 13 datasets. Collecting the best results among the reported results for the remaining seven datasets (Glass: 73.80%, Iris: 96.79%, Mfeat.ZER: 83.58%, Pen Digits: 99.61%, Sat Image: 92.24%, Vehicle: 87.47%, and Yeast: 59.99%) and averaging them, we have an average accuracy of 84.78%. The average accuracy of these datasets for MPCC is 84.12%. It implies that while we collected the best rates among the available results, the difference between the average of the collected results and the average of MPCC's results is not considerable and is 0.66%.

5. Summary and Conclusion

The aim of our study was to find an optimal feature space for clustering and to improve cluster-based multiclass classification performance. Therefore, we introduced MPC as an optimal feature space and MPC-Clustering algorithm accordingly. During the experiments, our interest rested in discovering how well MPC can outperform the original features in terms of classification. Thus, we introduced MPC-Direct algorithm, where it trained a given classifier on MPCs. To investigate the performance of the proposed algorithms, we conducted extensive experiments on 20 different datasets from a wide range of domains. According to the results we obtained, we summarized our conclusions as follows:

- We showed that projecting an original feature space to MPC and MBC outperformed it in terms of clustering.
- We also showed that the classification performance of MPC-Clustering was remarkably better than MBC-Clustering.
- It was shown that employing SVMs as the base learners together with SOM as the clustering scheme in MPCC algorithm outperformed three other pairs: SVM:k-means, MLP:SOM and MLP:k-means.
- Considering all the datasets used in the experiments, we showed that MPCC improved the classification rate by almost 2.4%.
- It was shown that the classification performance of MPCD was comparable to that of existing algorithms and in some datasets outperformed them including MPCC.
- And finally, we conclude that the proposed approach for projecting original features to a new feature space has advantages for both cluster-based classification and direct classification.

Although in this paper the performances of the proposed algorithms were evaluated extensively, we would like to investigate their performance on the real world applications such as Face Recognition (FR) and Facial Expression Recognition (FER). Firstly, the problems mentioned are well-suited for pattern classification and

secondly, a variety of datasets are available for them. Most importantly, the numbers of samples per patterns (classes) in the datasets available for FR and FER are almost equal. Therefore, the base learners, which are the core components of the proposed approach, can be trained more accurately to yield a high classification rate. We expect that applying the proposed algorithms on FR and FER problems will improve the classification rates.

Acknowledgments

This work is partly supported by NSF of China (No. 61070067), National 863 High-Tech Programme (No. 2009AA011900), and Zhejiang Provincial Natural Science Foundation of China (No. Y1090690).

References

1. O. A. Abbas, Comparisons between data clustering algorithms, *Int. Arab J. Inform. Technol.* **5**(3) (2007) 325−330.
2. E. Alpaydin, *Introduction to Machine Learning*, 2nd edn. (MIT Press, 2010).
3. M. R. Bin Abdullah, K.-A. Toh and D. Srinivasan, A framework for empirical classifiers comparison, *Proc. 1st IEEE Conf. Industrial Electronics and Applications* (2006), pp. 1−6.
4. C. M. Bishop, *Pattern Recognition and Machine Learning* (Springer, 2006).
5. BSVM. http://www.csie.ntu.edu.tw/cjlin/bsvm.
6. H. Byun and S.-W. Lee, A survey on pattern recognition applications of support vector machines, *Int. J. Patt. Recogn. Artif. Intell.* **17**(3) (2003) 459−486.
7. C.-C. Chang and C.-J. Lin, LIBSVM: A Library for Support Vector Machines. http://www.csie.ntu.edu.tw/∼cjlin/libsvm.
8. D. Chen and P. Burrell, On the optimal structure design of multilayer feedforward neural networks for pattern recognition, *Int. J. Patt. Recogn. Artif. Intell.* **16**(4) (2002) 375−398.
9. W. W. Cohen, Fast effective rule induction, *Proc. Twelfth Int. Conf. Machine Learning* (1995), pp. 115−123.
10. K. Crammer and Y. Singer, On the algorithmic implementation of multiclass SVMs, *J. Mach. Learn. Res.* **2** (2001) 265−292.
11. T. G. Dietterich and G. Bakiri, Solving multiclass learning problems via error-correcting output codes, *J. Artif. Intell. Res.* **2** (1995) 263−286.
12. *Finding the Right Number of Clusters in k-Means and EM Clustering: v-Fold Cross-Validation*, Electronic Statistics Textbook. http://www.statsoft.com/textbook/cluster-analysis/#vfold.
13. J. Furnkranz, Round robin classification, *J. Mach. Learn. Res.* **2** (2002) 721−747.
14. K.-S. Goh, E. Chang and K.-T. Cheng, SVM binary classifier ensembles for image classification, *Proc. Tenth Int. Conf. Information and Knowledge Management* (2001), pp. 395−402.
15. S. Haykin, *Neural Networks: A Comprehensive Foundation* (Prentice Hall, 1999).
16. C.-W. Hsu, C.-C. Chang and C.-J. Lin, A practical guide to support vector classification, Department of Computer Science, National Taiwan University, Technical Report 2003.
17. C.-W. Hsu and C.-J. Lin, A comparison of methods for multiclass support vector machines, *IEEE Trans. Neural Networks* **13**(2) (2002) 415−425.

18. M. Hülsmann and C. M. Friedrich, Comparison of a novel combined ECOC strategy with different multiclass algorithms together with parameter optimization methods, *Proc. 5th Int. Conf. Machine Learning and Data Mining in Pattern Recognition* (2007), pp. 17−31.
19. B. Kijsirikul and U. Nitiwut, Multiclass support vector machines using adaptive directed acyclic graph, in *Proc. Int. Joint Conf. Neural Networks* (2002), pp. 980−985.
20. T. Kohonen, *Self-Organizing Maps*, 3rd edn. (Springer, 2001).
21. A. C. Lorena, A. C. P. L. F. de Carvalho and J. M. P. Gama, A review on the combination of binary classifiers in multiclass problems, *Artif. Intell. Rev.* **30** (2009) 19−37.
22. K. G. Mehrotra, N. E. Ozgencil and N. McCracken, Squeezing the last drop: Cluster-based classification algorithm, *Statist. Probab. Lett.* **77** (2007) 1288−1299.
23. A. Mitiche and J. K. Aggarwal, Pattern category assignment by neural networks and nearest neighbors rule: A synopsis and a characterization, *Int. J. Patt. Recogn. Artif. Intell.* **10**(5) (1996) 393−408.
24. M. Ozay and F. T. Vural, On the performance of stacked generalization classifiers, in *Proc. 5th Int. Conf. Image Analysis and Recognition* (2008), pp. 445−454.
25. A. Passerini, M. Pontil and P. Frasconi, New results on error correcting output codes of kernel machines, *IEEE Trans. Neural Networks* **15**(1) (2004) 45−54.
26. J. C. Platt, N. Cristianini and J. Shawe-Taylor, Large margin dags for multi-class classification, *Adv. Neural Inform. Process. Syst.* **12** (2000) 547−553.
27. R. Rifkin and A. Klautau, In defense of one-vs-all classification, *J. Mach. Learn. Res.* **5** (2004) 101−141.
28. L. Rokach, Ensemble-based classifiers, *Artif. Intell. Rev.* **33** (2009) 1−39.
29. L. Rokach, *Pattern Classification Using Ensemble Methods*. Series in Machine Perception and Artificial Intelligence, Vol. 75 (World Scientific, 2010).
30. Y. Sun, A. K. C. Wong and M. S. Kamel, Classification of imbalanced data: A review, *Int. J. Patt. Recogn. Artif. Intell.* **23**(4) (2009) 687−719.
31. S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, 2nd edn. (Academic Press, 2003).
32. K. M. Ting and I. H. Witten, Stacked generalization: When does it work? in *Proc. Int. Joint Conf. Artificial Intelligence* (1997), pp. 866−871.
33. K. M. Ting and I. H. Witten, Issues in stacked generalization, *J. Artif. Intell. Res.* **10** (1999) 271−289.
34. K.-A. Toh, Q.-L. Tran and D. Srinivasan, Benchmarking a reduced multivariate polynomial pattern classifier, *IEEE Trans. Patt. Anal. Mach. Intell.* **26**(6) (2004) 740−755.
35. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml/index.html.
36. H. Vafaie and K. De Jong, Robust feature selection algorithms, in *Proc. 5th Int. Conf. Tools and Artificial Intelligence* (1993), pp. 356−363.
37. V. N. Vapnik, *Statistical Learning Theory* (Wiley, NY, 1998).
38. B. Verma and M. Blumenstein, *Pattern Recognition Technologies and Applications: Recent Advances* (Information Science Reference, 2008).
39. J. Weston and C. Watkins, Multi-class support vector machines, in *European Symp. Artificial Neural Networks*, Vol. 4 (1999), pp. 219−224.
40. I. H. Witten and E. Frank, *Data Mining*, 2nd edn. (Morgan Kaufmann, 2005).
41. D. H. Wolpert, Stacked generalization, *Neural Networks* **5** (1992) 241−259.
42. T.-F. Wu, C.-J. Lin and R. C. Weng, Probability estimates for multi-class classification by pairwise coupling, *J. Mach. Learn. Res.* **5** (2003) 975−1005.

Nacer Farajzadeh received both the bachelor's and master's degrees in software engineering from Azad University, Iran, in 2002 and 2005 respectively. He is currently working toward the Ph.D. degree at the Biometrics Lab in the College of Computer Science and Technology, Zhejiang University, China.

His current research interests include machine learning for pattern recognition and machine vision.

Gang Pan received the B.S. and Ph.D. degrees in computer science from Zhejiang University, Hangzhou, China, in 1998 and 2004, respectively. Since then, he has been with Zhejiang University, where he has been an associate professor of computer science from 2006.

His research interests include pervasive computing, computer vision, and pattern recognition. He has served as a PC member for numerous international conferences, such as ICCV, CVPR, UIC, and as a reviewer for more than ten journals, such as TPAMI, TIP, TVCG, and TSMC-B.

Zhaohui Wu received the Ph.D. degree in computer science from Zhejiang University, Hangzhou, China, in 1993. From 1991 to 1993, he was with the German Research Center for Artificial Intelligence (DFKI) as a joint Ph.D. student in the area of knowledge representation and expert system. Currently he is a professor of computer science with Zhejiang University and the Director of the Institute of Computer System and Architecture. He has authored four books and more than 100 refereed papers.

His major interests include intelligent systems, semantic grid, and ubiquitous embedded systems. He is on the editorial boards of several journals and has served as PC member for various international conferences.

Min Yao received the B.E. degree in radio technique from Hefei University, China, in 1982, the M.E. degree in computer science from Hefei University of Technology, China, in 1986, and the Ph.D. degree in biomedical engineering and instrumentation from Zhejiang University, China, in 1995. He is currently a professor in the College of Computer Science and Technology at Zhejiang University, Hangzhou, China.

His research interests include pattern recognition, computational intelligence and data mining.