Searching for Optimal Classifier Using a Combination of Cluster Ensemble and Kernel Method

Vladimir B. Berikov[1,2] and Lyailya Sh. Cherikbayeva[3]

[1] Sobolev Institute of Mathematics, Novosibirsk, Russia
berikov@math.nsc.ru
[2] Novosibirsk State University, Novosibirsk, Russia
[3] Al-Farabi Kazakh National University, Almaty, Kazakhstan
lailash01@gmail.com

Abstract. This work introduces a supervised classification algorithm based on a combination of ensemble clustering and kernel method. The main idea of the algorithm lies behind the expectation that the ensemble clustering as a preliminary stage would restore more accurately metric relations between data objects under noise distortions and existence of complex data structures, eventually rising the overall classification quality. The algorithm consists in two major steps. On the first step, the averaged co-association matrix is calculated using cluster ensemble. It is proved that the matrix satisfies Mercer's condition, i.e., it defines symmetric non-negative definite kernel. On the next step, optimal classifier is found with the obtained kernel matrix as input. The classifier maximizes the width of hyperplane's separation margin in the space induced by the cluster ensemble kernel. Numerical experiments with artificial examples and real hyperspectral image have shown that the proposed algorithm possesses classification accuracy comparable with some state-of-the-art methods, and in many cases outperforms them, especially in noise conditions.

Keywords: Kernel based learning · Cluster ensemble · Co-association matrix · Support vector machine

1 Introduction

Kernel based learning and collective decision making (ensemble approach) stay among top trends influencing the progress in machine learning.

Kernel (potential) function defines an implicit non-linear mapping of the initial feature space into a new space with larger or even infinite dimensionality.

In: S. Belim et al. (eds.): OPTA-SCL 2018, Omsk, Russia, published at http://ceur-ws.org

In the new space, initial configurations of patterns are transformed (with so called "kernel trick") into the structures which often are more compact and linearly separable. To calculate distances or average values in the new space, it is not necessary to determine coordinates of the transformed points; it is enough to only know the values of kernel function. Such methods as Support Vector Machine (SVM), Kernel Fisher Discriminant (KFD), Kernel K-means, Kernel Principal Component Analysis, etc., are based on this methodology [1]. The decision function is defined by a linear combination of kernels determining distances to a number of sample elements. In the existing algorithms, the kernel matrix is usually calculated with use of the Euclidean distance between objects in the input feature space.

Ensemble approach exploits the idea of collective decision making by usage of algorithms working on different settings such as subsets of parameters, subsamples of data, combinations of features, etc. Ensemble based systems usually yield robust and effective solution, especially in case of uncertainty in data model or when it is not clear which of algorithm's parameters are most appropriate for a particular problem. As a rule, properly organized ensemble (even composed of "weak" predictors) significantly improves the overall quality of decisions [2–6].

Cluster analysis aims at determining a partition of a dataset on natural clusters using objects descriptions and a certain criterion of compactness-remoteness of groups. Ensemble clustering is one of the successful implementations of the collective methodology. There are a number of major techniques for constructing the ensemble decision [7–9]. Following *evidence accumulation* approach [10], the clustering partition is found in two steps. On the first step, a number of clustering results are obtained (for example, by usage of K-means for different number of clusters or random initializations of centroids). For each partition variant, the co-association boolean matrix is calculated. The matrix elements correspond to the pairs of data objects and indicate if the pair belong to the same cluster or not. On the second step, the averaged co-association matrix is calculated over all variants; it is used for constructing the resultant partition: the matrix elements are considered as distances or similarity measures between data points and any clustering algorithm designed for such type of input information is applied to get the final clustering partition.

This paper introduces an algorithm of classifier construction using a combination of ensemble clustering and kernel based learning. The proposed methodic is based on the hypothesis that the preliminary ensemble clustering allows one to restore more accurately metric relations between objects under noise distortions and existence of complex data structures. The obtained kernel matrix depends on the outputs of clustering algorithms and is less noise-addicted than conventional similarity matrix. Clustering with sufficiently large number of clusters can be viewed as Learning Vector Quantization methodic [11] known for lowering the average distortion in data. These reasons, as supposed, eventually result in an increase of recognition accuracy of the combination. The outline of the method is as follows. First of all, a number of variants of a dataset partitioning are obtained with base clustering algorithm. Then the averaged co-association

matrix is calculated, where the averaging is performed with weights dependent on the obtained ensemble's characteristics. The matrix elements play the role of similarity measures between objects in the new feature space induced by implicit non-linear transformation of input features. On the second stage, a kernel classifier is found by usage of the obtained co-association matrix as input kernel matrix (we used SVM in numeric experiments).

The aim of this paper is to verify the practicability of the suggested methodology with theoretical analysis and experimental evaluation.

There are two main types of cluster ensembles: homogeneous (when a single algorithm partitions data by varying its working settings) and heterogeneous ones (which includes a number of different algorithms). Heterogeneous cluster ensemble was considered in [12, 13], where methods for its weights optimization were suggested. Homogeneous cluster ensemble was investigated in [14] with use of the probabilistic model assuming the validity of some key assumptions. In the current work, we follow a scheme of homogeneous ensemble and perform theoretical investigation of some of its properties using less restrictive assumptions.

The rest of the paper is organized as follows. Section 2 briefly overviews related works. Section 3 introduces necessary notions in the field of kernel based classifiers and ensemble clustering. In The Next Section We Prove That the Weighted Co-association Matrix obtained with cluster ensemble is a valid kernel matrix. The proposed algorithm of classifier design KCCE is also presented and some details of the optimization procedure are given. Section 5 provides a probabilistic analysis of the ensemble clustering stage. The Final Section Describes the Results of Numerical Experiments with KCCE. The conclusion summarizes the work and describes some of the future plans.

2 Related Works

The idea of combining cluster analysis and pattern recognition methods is rather well-known in machine learning [15]. There are several natural reasons for the combination:

- Cluster analysis can be viewed as a tool for data cleaning to eliminate outliers or noisy items from learning sample [16].
- Joint learning and control sample provides additional information on data distribution which can be utilized to improve the classifier performance (this way of reasoning is sometimes called the *trunsductive learning*). For example, the authors of [17] make a partition of the united sample into clusters which are used to design more accurate decision rule.
- In semi-supervised learning context [18], usage of small amount of labeled data in combination with a large volume of unlabeled examples is useful for constructing more efficient classifier.

A connection between cluster analysis and kernel based classifiers was established in [19], where *cluster kernels* were proposed implementing the *cluster assumption* in the form: "two points are likely to have the same class label if

there is a path connecting them passing through regions of high density only". Three types of kernels were presented: kernels from mixture models, random walk kernels and kernels induced by a cluster representation with spectral clustering [20].

The usage of a certain similarity function (which not necessarily possesses positive semi-definiteness property) instead of kernel function was proposed in [21]. A classifier is finding in two stages. On the first stage, the choice of some "supporting" points is performed. With regard to these points, according to the defined similarity function, initial observations are mapped into metric space of small dimensionality. On the second stage, a linear classification rule is constructed in the new space implementing SVM-type algorithm to find the classification margin of maximum width.

Following the idea of combining cluster ensembles and supervised classification, the authors of [22] construct new feature space by usage of the degree of belonging of objects to clusters in the obtained variants of data partitioning with cluster ensemble. The transomed feature matrix is utilized as input training set for classification using conventional techniques such as Decision Tree, Naive Bayes, K-nearest neighbors, Neural Network. The method showed its effectiveness in comparison with a number of state-of-the-art procedures.

Unlike the above mentioned works, we apply completely different combination scheme based on the notion of kernel function.

3 Basic Preliminaries

Suppose we are given a data set $A = \{a_1, \ldots, a_N\}$ consisting of N objects (examples), $A \subset \Gamma$, where Γ is a statistical population. Information about the objects is presented in the form of a feature matrix $\mathbf{Z} = (\mathbf{X}, \mathbf{Y}) = (x_i, y_i)_{i=1}^N$, where $x_i = (x_{i,1}, \ldots, x_{i,d}) \in \mathbf{R}^d$ is input feature vector (d is feature space dimensionality), $x_{i,m} = X_m(a_i)$ is a value of feature X_m for object a_i; y_i is a class label attributed to ith object, $i = 1, \ldots, N$. For binary classification task we assume $y_i \in \{-1, 1\}$. In multi-class classification problem, an arbitrary finite set of unordered class labels is defined.

On the basis of the information about A (training sample), it is required to find a classifier (predictor, decision function) $y = f(x)$, optimal in some sense, e.g. having minimal expected losses for unseen examples. To examine the performance of the classifier, it is possible to use test sample $B = \{b_1, \ldots, b_{N_t}\}$, $B \subset \Gamma$ described with feature matrix $\mathbf{X_{test}}$. We shall presume that the objects in A and B are independent and identically distributed (iid), that is, the sets are collected on the basis of independent random choice of objects from Γ without replacement following a fixed distribution.

Kernel classifiers [1] make use of the notion of kernel function $K(x_i, x_j) \geq 0$, where K is a kind of similarity measure between two data points. Linear kernel classifier exemplifies a binary decision function introduced within this approach: $f(x) = sign(\sum_{x_i \in \mathbf{X}} \alpha_i y_i K(x, x_i))$, where $sign$ is the sign function, $\alpha_1, \ldots, \alpha_N$ are

non-negative weights. A number of methods for determining weights (Support Vector Machine, Kernel Fisher Discriminate, etc.) exist.

For the SVM classifier, the weights are found as a solution to the constrained quadratic optimization problem of maximizing the width of the margin (separation region) between two classes in Hilbert's space induced by kernel mapping.

KFD is a kernelized version of Fisher's linear discriminant analysis (LDA) which aims at finding such a position of a straight line in feature space, for which the object's projections are separated as better as possible according to a functional minimizing within-class scatter of projections and maximizing between-class distance.

The general multi-class classification problem can be solved by the application of a series of binary classification tasks for SVM or KFD, e.g., one-against-all, one-against-one or Error Correcting Output Codes (ECOC) schemes [23].

Kernel k-NN classifier assigns data points according to k Nearest Neighbor rule, where neighboring points are determined with respect to similarity measure defined by kernel function.

Consider a scheme of homogeneous cluster ensemble. Let a clustering algorithm μ be running a number of times under different conditions such as initial cluster centroids coordinates, subsets of features, number of clusters or other parameters. The joined data set $A \cup B$ is the input for the algorithm (if test sample is unavailable in the moment of classifier design, then set A is the input). In each lth trial, algorithm μ creates a partition of the given dataset composed of K_l clusters, where $l = 1, \ldots, L$, and L is the given number of runs. For each variant of clustering, we define the evaluation function γ_l (cluster validity index or diversity measure). We suppose that the values are scaled to non-negative quantities and the better is the found variant according to certain criterion, the larger is the quantity.

For a pair of different data objects $(a_i, a_j) \in A \cup B$, we define the value $h_l(i,j) = \mathbf{I}[\mu_l(a_i) = \mu_l(a_j)]$, where $\mathbf{I}[\cdot]$ is the indicator function: $\mathbf{I}[true] = 1$; $\mathbf{I}[false] = 0$; $\mu_l(a)$ is the cluster label assigned by algorithm μ to object a in lth run. Ensemble matrix M stores the results of clusterings: $M = (\mu_l(a_i))_{i=1,\ldots,N+N_t}^{l=1,\ldots,L}$.

The averaged co-association matrix $\mathbf{H} = (\bar{h}(i,j))$ is defined over all generated variants: $\bar{h}(i,j) = \sum_{l=1}^{L} u_l \, h_l(i,j)$, where the standardized weights $u_1, \ldots, u_L$ indicate the quality of clustering for the given variants: $u_l = \frac{\gamma_l}{\sum \gamma_{l'}}$, $l = 1 \ldots, L$.

4 Kernel Classification with Averaged Co-association Matrix

Let $K(x, x')$: $D \times D \to \mathbf{R}$ be a symmetric function, either continuous or having a finite domain, D be a closed subset in $\mathbf{R}^d$. According to Mercer's theorem, $K(x, x')$ is kernel function (i.e., it defines inner product in some metric space), if and only if for any finite set of m points $\{x_i\}_{i=1}^m$ in D and real num-

bers $\{c_i\}_{i=1}^m$, matrix $\mathbf{K} = (K(i,j)) = (K(x_i, x_j))_{i,j=1}^m$ is nonnegativity definite: $\sum\limits_{i,j=1}^{m} c_i c_j K(i,j) \geq 0$. Let us prove the following

Proposition 1. *The averaged co-association matrix satisfies Mercer's condition.*

Proof. The symmetric property of $\mathbf{H}$ is obvious. The domain of $\mathbf{H}$ is a finite set $A \cup B$. Let $I_r^{(l)}$ be the set of indices for data points belonging to rth cluster in lth variant of partitioning. Then for any $\{c_i\}_{i=1}^m$ it holds true: $\sum\limits_{i,j=1}^{m} c_i c_j \bar{h}(i,j) =$

$$\sum_{i,j=1}^{m} c_i c_j \sum_{l=1}^{L} u_l \, h_l(i,j) = \sum_{l=1}^{L} u_l \sum_{i,j=1}^{m} c_i \, c_j \, h_l(i,j) = \sum_{l=1}^{L} u_l \sum_{k=1}^{K_l} \sum_{i,j \in I_k^{(l)}} c_i c_j$$

$$= \sum_{l=1}^{L} u_l \sum_{k=1}^{K_l} \Big(\sum_{i \in I_k^{(l)}} c_i \Big)^2 \geq 0.$$

From this property, it follows that the averaged co-association matrix is a valid kernel matrix and can be used in kernel based classification methods.

Let us describe the main steps of the proposed algorithm KCCE (Kernel Classification with Cluster Ensemble).

Algorithm KCCE.
Input:
training data set $\mathbf{Z} = (\mathbf{X}, \mathbf{Y}) = (x_i, y_i)$, $i = 1, \ldots, N$;
test data set $\mathbf{X_{test}}$;
L: number of runs for base clustering algorithm μ;
$\mathbf{\Omega}$: set of allowable parameters (working conditions) of μ.
Output:
decision function $y = f(x)$; class labels attributed to $\mathbf{X_{test}}$.
Steps:
1. Generate L variants of clustering partition of $\mathbf{X} \cup \mathbf{X_{test}}$ using algorithm μ with randomly chosen working parameters; calculate evaluation functions and weights;
2. For each pair $(x_i, x_j) \in \mathbf{X} \cup \mathbf{X_{test}}$ $(i \neq j)$, if the pair are assigned to the same group in lth variant, then $h_l(i,j) := 1$, otherwise $h_l(i,j) := 0$;
3. Calculate the averaged co-association matrix $\mathbf{H}$;
4. Find decision function with the preset type of kernel classifier and matrix $\mathbf{H}$;
5. Classify test sample $\mathbf{X_{test}}$ using the found decision function and matrix $\mathbf{H}$;
end.

In this paper, we use K-means as base clustering algorithm, however it is possible to apply any other clustering technique. As the kernel classifier, we utilize soft margin version of SVM which aims at optimizing the following objective

function:

$$\begin{array}{l} \frac{1}{2}||w||^2 + C\sum\limits_i \xi_i \rightarrow \min\limits_{w,b,\xi} \\ \text{subject to:} \\ y_i\left(\langle w, x_i\rangle + b\right) \geq 1 - \xi_i \\ \xi_i \geq 0, \quad i = 1, ..., N, \end{array}$$

where w is normal vector to the separating hyperplane in the space induced by kernel, b is hyperplane's bias, ξ_i is a penalty imposed on ith example violating the separation margin, $C \geq 0$ is soft margin parameter. By solving for the Lagrangian dual, one obtains the quadratic optimization problem:

$$W(\alpha) = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \;\rightarrow\; \max$$

$$\text{subject to: } \sum_i \alpha_i y_i = 0,\; 0 \leq \alpha_i \leq C,\; i = 1, \ldots, N,$$

where $K(\cdot,\cdot)$ is kernel function. One may substitute this kernel by the cluster ensemble kernel $\mathbf{H}$ and get:

$$W(\alpha) = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \sum_l u_l\, h_l(i,j)$$

$$= \sum_i \alpha_i - \frac{1}{2}\sum_l u_l \sum_{k=1}^{K_l} \sum_{i,j \in I_k^{(l)}} \alpha_i \alpha_j y_i y_j = \sum_i \alpha_i - \frac{1}{2}\sum_l u_l \sum_{k=1}^{K_l} (\sum_{i \in I_k^{(l)}} \alpha_i y_i)^2.$$

We search for the optimal solution using Sequential Minimal Optimization (SMO) method [24]. A point x_i for which $\alpha_i > 0$ is called *support vector*. The bias term b is determined by any support vector x_{i^*}: $b = y_{i^*} - \sum\limits_i y_i \alpha_i \mathbf{H}(x_i, x_{i^*})$.

The decision for $x_j \in \mathbf{X_{test}}$ is calculated using the found multipliers: $f(x_j) = sign(\sum\limits_i y_i \alpha_i \mathbf{H}(x_j, x_j) + b)$.

One can see that there is no need to store the kernel matrix: the objective function is computed using the ensemble matrix M kept in memory. Therefore KCCE has linear storage complexity with respect to data size. The time complexity depends on the type of utilized clustering and kernel algorithms and is linear with respect to data matrix dimensionality in case of K-means and SVM-SMO.

In some classification tasks, test data are unavailable in the moment of classifier design. To find the decision function $f(x)$ for any new feature vector x, this observation should be attributed to clusters according to the obtained partition variants. It is possible to make this assignment using cluster centroid coordinates which can be stored in memory during the implementation of Step 1 in KCCE. The observation is assigned to the nearest centroid's label for each clustering variant.

5 On the Reliability of Ensemble Clustering

The suggested combined method includes two main phases: ensemble clustering and kernel classifier design. An important question is the reliability of the first step: are the elements of the obtained similarity matrix $\mathbf{H}$ fit true relationships between object pairs (e.g., belonging to same or different classes)? To study this problem, we use the methodology described in [12, 14, 13].

In the clustering process, true class labels are unavailable. Following a probabilistic approach, one may suppose that data sample is composed of a finite number of components. A latent groundtruth variable Y' defines the class number to which an object belongs. Denote

$$v(i,j) = \mathbf{I}[\, Y'(a_i) = Y'(a_j)\,], \tag{1}$$

where a_i and a_j are arbitrary objects from input sample. This quantity determines the true status of the pair (i.e., if a_i and a_j indeed belong to the same class).

Function $c(i,j) = \mathbf{I}\,[\sum\limits_{l:h_l(i,j)=1} u_l > \sum\limits_{l:h_l(i,j)=0} u_l]$ will be called the ensemble decision for a_i and a_j following weighted voting procedure.

For a pair (a_i, a_j), their ensemble's margin is defined as

$$mg(i,j) = \{ \sum_{l:h_l(i,j)=v(i,j)} u_l - \sum_{l:h_l(i,j)\neq v(i,j)} u_l\}$$

and can be rewritten in the form:

$$mg(i,j) = \sum_{l=1}^{L} u_l \left\{\mathbf{I}[h_l(i,j) = v(i,j)] - \mathbf{I}[h_l(i,j) \neq v(i,j)]\right\}.$$

This value indicates to what extend the number of right decisions for (a_i, a_j) exceed the number of wrong ones. Evidently, it equals:

$$mg(i,j) = \sum_{l=1}^{L} u_l (2v(i,j) - 1)(2h_l(i,j) - 1). \tag{2}$$

The margin can not be calculated if the true partition is unknown. However, it was shown in [12, 14] that some of margin's characteristics can be evaluated using a number of assumptions on the behavior of clustering algorithms:
A1) Under the condition that input data matrix is fixed, for any pair of objects (a_i, a_j) their true status is a random value $V(i,j)$ with values defined in (1).
A2) Algorithm μ is randomized, i.e. it depends on the vector Ω chosen at random from the set of parameters $\mathbf{\Omega}$. For fixed input data, μ is running L times with i.i.d. parameters $\Omega_1, \ldots, \Omega_L$ being statistical copies of Ω.

Conditional probabilities of correct decisions (partition or union of a_i and a_j under fixed $V(i,j)$) are denoted as

$$q_0(i,j) = P[h(i,j,\Omega) = 0 | V(i,j) = 0], \;\; q_1(i,j) = P[h(i,j,\Omega) = 1 | V(i,j) = 1],$$

where $h(i,j,\Omega)$ is the decision for (a_i, a_j) made by algorithm μ with random parameters Ω. It should be noted that the work [14] makes use of more restrictive assumption: $q_0(i,j) = q_1(i,j)$, i.e. it presumes that the conditional probabilities of correct assigning of both kinds coincide. This assumption could be used for the qualitative analysis of cluster ensemble's behavior; however, numerical results of [13] demonstrate that it can be violated.

The measure of clustering validity, estimated with algorithm μ, is represented as a random value $\gamma(\mathbf{X}, \Omega)$. Because the quality criterion is determined on the whole data set, one may consider this value practically independent on $V(i,j)$ and $h(i,j,\Omega)$.

The weights $u_1, \ldots, u_L$ are random values following identical distribution defined by the distribution of $\gamma(\Omega)$ and its statistical copies $\gamma_1(\Omega_1), \ldots, \gamma_L(\Omega_L)$. The weights are dependent on each other, and for any pair (u_{l_1}, u_{l_2}) the degree of their dependence is characterized with the covariance coefficient $\sigma = cov[u_{l_1}, u_{l_2}]$.

From i.i.d. assumption, and because of $\sum\limits_l E[u_l] = E\left[\sum\limits_l u_l\right] = 1$, it follows that $E[u_l] = \dfrac{1}{L}$. Let us denote by $s = Var[u_l]$ the variance of weights. From

$$0 = Var\left[\sum_l u_l\right] = \sum_l Var[u_l] + \sum_{l_1, l_2\ (l_1 \neq l_2\ )} cov[u_{l_1}, u_{l_2}],$$

one may conclude that $LVar[u_l] + L(L-1)cov[u_{l_1}, u_{l_2}] = 0$. Thus we get

$$\sigma = -\frac{s}{L-1}. \tag{3}$$

Proposition 2. *Given the assumptions A1), A2) be valid, conditional mathematical expectation of ensemble margin for (a_i, a_j) under $V(i,j) = v$ equals:*

$$E_{\bar{\Omega}}[mg(i,j) \mid V(i,j) = v] = 2Q(v;i,j) - 1,$$

where $\bar{\Omega} = (\Omega_1, \ldots, \Omega_L)$, $Q(v;i,j) = (1-v)\, q_0(i,j) + v\, q_1(i,j)$, $v \in \{0,1\}$.

Proof. Let us denote by $h_l(i,j,\Omega_l) = \mathbf{I}[\mu(i,\Omega_l) = \mu(j,\Omega_l)]$ the decision for (a_i, a_j), where $\mu(i,\Omega_l)$ is the cluster label assigned to object a_i by algorithm μ in its lth run with usage of parameter vector Ω_l.

Until the proof end, arguments i, j are skipped for short: $mg(i,j) = mg$, $V(i,j) = V$, etc. From (2) we have:

$E_{\bar{\Omega}}[mg|V = v] = \sum\limits_l E_{\bar{\Omega}}[u_l\,(\Omega_l)(2v-1)(2h_l(\Omega_l)-1)]$.

Because Ω_l and Ω are equally distributed and $u_l(\Omega_l)$ is independent on $h_l(\Omega_l)$, it holds true that

$E_{\bar{\Omega}}[mg|V = v] = \sum\limits_l E[u_l(\Omega)]\,(2v-1)(2E_\Omega[h(\Omega)]-1) = (2v-1)(2E_\Omega[h(\Omega)]-1)$.

For $v = 0$ we have: $(2v-1)(2E_\Omega[h(\Omega)]-1) = -(2P[h(\Omega) = 1|V = 0]-1) = 2q_0 - 1$; and for $v = 1$: $(2v-1)(2E_\Omega[h(\Omega)]-1) = 2P[h(\Omega) = 1|V = 1] - 1 = 2q_1 - 1$. Therefore $E_{\bar{\Omega}}[mg|V = v] = 2Q(v;i,j) - 1$. This completes the proof.

Proposition 3. *Given the assumptions A1), A2) be valid, conditional variance of ensemble margin for* (a_i, a_j) *under* $V(i,j) = v$ *equals:*

$$Var_{\bar{\Omega}}[mg(i,j) \mid V(i,j) = v] = 4Q(v;i,j)(1 - Q(v;i,j))\,(L\,s + \frac{1}{L}).$$

Proof. Let us again skip indices i, j for simplicity; and also let h_l denote $h_l(\Omega_l)$, $h = h(\Omega)$, $u_l = u_l(\Omega)$. From the properties of variance, it follows that under condition $V = v$ it holds true:

$$Var_{\bar{\Omega}}[mg] = (2v-1)^2 Var\left[\sum_l E[u_l]\,(2E[h_l] - 1)\right] = Var\left[\sum_l 2u_l h_l - 1\right] =$$

$$4Var\left[\sum_l u_l h_l]\right] = 4\sum_l Var[u_l h_l] + 4\sum_{l_1,l_2\,(l_1 \neq l_2)} cov[u_{l_1}h_{l_1},\, u_{l_2}h_{l_2}] =$$

$$4\sum_l \left(Var[u_l]Var[h_l] + (E[u_l])^2\,Var[h_l] + (E[h_l])^2 Var[u_l]\right) +$$

$$4\sum_{l_1,l_2\,(l_1 \neq l_2)} cov[u_{l_1}h_{l_1},\, u_{l_2}h_{l_2}] =$$

$$4\,L\,s\,Var[h] + 4\,Var[h]/L + 4\,L\,s(E[h])^2 + 4\sum_{l_1,l_2\,(l_1 \neq l_2)} cov[u_{l_1}h_{l_1},\, u_{l_2}h_{l_2}]. \quad (4)$$

Evidently, $Var[h \mid V = v] = Q(v)(1 - Q(v))$. From the independence of all pairs h_{l_1} and h_{l_2}, we have: $\sum_{l_1,l_2\,(l_1 \neq l_2)} cov[u_{l_1}h_{l_1},\, u_{l_2}h_{l_2}] =$

$$\sum_{l_1,l_2\,(l_1 \neq l_2)} (E[u_{l_1}u_{l_2}]E[h_{l_2}h_{l_1}] - E[u_{l_1}h_{l_1}]E[u_{l_2}h_{l_2}]) =$$

$$\sum_{l_1,l_2\,(l_1 \neq l_2)} (E[u_{l_1}u_{l_2}](E[h])^2 - (E[h])^2 E[u_{l_1}]E[u_{l_2}]) =$$

$$(E[h])^2 \sum_{l_1,l_2\,(l_1 \neq l_2)} (E[u_{l_1}u_{l_2}] - E[u_{l_1}]E[u_{l_2}]) = (E[h])^2\,L(L-1)\,\sigma.$$

Using (3), (4) we see that

$$Var\,[mg] = 4Q(1-Q)\,L\,s + \frac{4}{L}Q(1-Q) + 4(E[h])^2 L\,s - 4\,(E[h])^2\,L\,s =$$

$$4Q(1-Q)\,(L\,s + \frac{1}{L}). \qquad \blacksquare$$

Now we consider the upper bound for the weight's variance.

Proposition 4. *Under the validity of the assumptions A1) and A2), the variance* $s = Var[u_l]$ *is upper bounded with the expression:* $s \leq \frac{1}{L^2}(E[\gamma(\Omega)^{-2}] - 1)$.

The proof technically repeats the one for analogous statement in [14], p. 430.

Our main objective is finding dependencies between the observed ensemble characteristics and the probability of directly unobserved classification error for a pair a_i, a_j:

$$P_{err}(i,j) = P_{\Omega_1,\ldots,\Omega_L,U(i,j)}[\, c(i,j) \neq V(i,j)].$$

It is clear that the probability of error in ensemble classification of a pair equals $P_{err}(i,j) = P[mg(i,j) < 0]$.

Now let us consider the conditional probability of error under given state of objects: $P_{err}(v;i,j) = P_{\Omega_1,\ldots,\Omega_L,V(i,j)}[\, c(i,j) \neq V(i,j) \,|\, V(i,j) = v], \;\; v \in \{0,1\}$.

Proposition 5. *Let the above introduced model assumptions A1), A2) be valid, and also let $E_{\bar{\Omega}}[mg(i,j) \mid V(i,j) = v] > 0$ for each (i,j). Then the conditional probability of error in classification of a given pair is upper bounded by the expression:*

$$P_{err}(v;i,j) < \frac{Var_{\bar{\Omega}}[mg(i,j) \mid V(i,j) = v]}{(E_{\bar{\Omega}}[mg(i,j) \mid V(i,j) = v])^2},$$

where conditional mathematical expectation and variance of margin are given by Propositions 2,3.

This property directly follows from the Tchebychev's inequality. The proof is similar to one given in [14], p. 433, and skipped in this work for the sake of brevity.

From Proposition 2, it is clear that the margin expectation takes positive value if the following assumption is valid:

$$A3)\; \forall i,j \;(i \neq j), \;\; 0.5 < q_0(i,j) \leq 1, \; 0.5 < q_1(i,j) \leq 1.$$

It means that the base clustering algorithm has at least slightly better classification quality than just random assignment of a pair to the same or different clusters. In machine learning theory, similar conditions are known as algorithm's *weak learnability.*

Let us formulate an important consequence of the obtained results.

Proposition 6. *Holding all other factors constant, under validity of assumptions A1), A2) and A3), conditional probability of error converges to zero as the ensemble size increases.*

This property follows from Propositions 2-5 taking into account that the weight's variance s is of order $O(L^{-2})$.

6 Numerical Experiments

This Section Describes Numerical Experiments with Kcce. In the First experiment, we used Monte Carlo statistical modeling technique to evaluate the quality of classification. We consider a simple case of two spherical Gaussian classes $N(m_1, \Sigma)$ and $N(m_2, \Sigma)$, each having diagonal covariance matrix $\Sigma = \sigma \mathbf{I}$,

where $\sigma > 0$ is a parameter, m_1 and m_2 are population means (in our experiments, $m_1 = \mathbf{0}$ and $m_2 = \mathbf{1}$). The classes are of equal prior probabilities $P_1 = P_2 = 0.5$. In this simple case, it is possible to derive the Bayes probability of error $P_B = \Phi(-\delta/2)$, where $\delta = (m_1 - m_2)^T \Sigma^{-1}(m_1 - m_2)$ is the Machalanobis distance, Φ is the standard Gaussian cumulative distribution function. Moreover, the expected generalization error for the optimal sample-based linear classifier [25] asymptotically equals $P_N = \Phi\left(-\frac{\delta}{2}\frac{1}{\sqrt{1+\frac{1}{N}(1+\frac{2d}{\delta^2})+\frac{d}{N^2\delta^2}}}\right)$.

In Monte Carlo modeling, we repeatedly generate data sets according to the given distributions. For each class, training sample size equals N. Each data set is analyzed in Matlab environment with SVM, KFD and KCCE. For SVM and KFD, radial basis function $\varphi(x, x') = e^{\frac{||x-x'||}{2\sigma_r}^2}$ with $\sigma_r = 5$ is used as a kernel. The soft margin constant is $C = 10$. The cluster ensemble is generated for KCCE by random initialization of centroids in K-means (number of clusters equals 2). Ensemble size equals 10. The weights of partition variants are constant values.

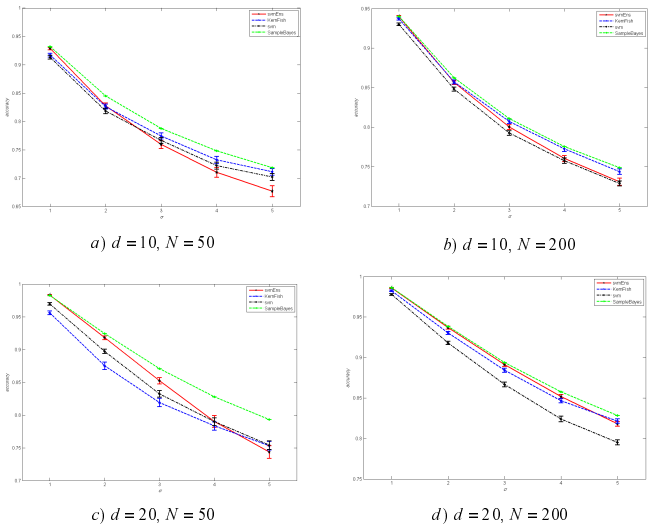
The accuracy (probability of correct classification) of each algorithm is estimated by independent test sample of size $N_t = 1000$. To make the results more statistically sound, we averaged the accuracy estimates over 100 Monte Carlo repetitions. The 95% confidence intervals for the probability of correct classification are evaluated for each algorithm.

Figure 1 presents the results of modeling. The plots display the dependencies between algorithm's accuracy and standard deviation σ for different combinations of feature space dimensionality and training sample size.

The results show that there exist such examples of data distribution for which the proposed method demonstrates substantially more accurate classification than SVM or KFD, and approaches to the optimal Bayes classifier.

In the second experiment we consider a real hyperspectral satellite image "Indian Pines" taken from [26]. This scene was gathered by AVIRIS sensor over the Indian Pines test site in North-western Indiana. The image size is 145×145 pixels; each pixel is characterized by the vector of 224 spectral intensities in 400-2500 nm range. The image includes 16 classes describing different vegetation types, as one can see in Figure 2. There are unlabeled pixels not assigned to any of the classes. These pixels are excluded from the analysis. To study the effect of noise on the performance of the algorithms, randomly selected $100r\%$ of the spectral intensity values have experienced a distorting effect: the corresponding value x is replaced by the quantity generated from the interval $[x(1-p), x(1+p)]$, where r, p are preset parameters. The dataset has been randomly divided on training and test sample in proportion 1:3.

We use multiclass SVM following "one-against-one" strategy. Cluster ensemble size is $L = 200$. For the construction of each variant, three hyperspectral channels are randomly chosen. To obtain more diverse results of K-means, the number of its iterations is limited to 1, and the initial centroids are randomly sampled from data. In the ensemble generation, data matrix $\mathbf{X_{test}}$ is not used. The number of clusters in each variant equals $\lceil\sqrt{N}\rceil$. The weights of clusterings are constant values.



a) $d = 10$, $N = 50$ *b*) $d = 10$, $N = 200$

c) $d = 20$, $N = 50$ *d*) $d = 20$, $N = 200$

Fig. 1. Results of Monte Carlo experiments for a number of combinations of feature space dimensionality d and training sample size N. "svmEns": averaged accuracy of the proposed algorithm KCCE; "svm": averaged accuracy of SVM; "KernFish": averaged accuracy of KFD algorithm; "SampleBayes": accuracy of optimal sample-based linear classifier

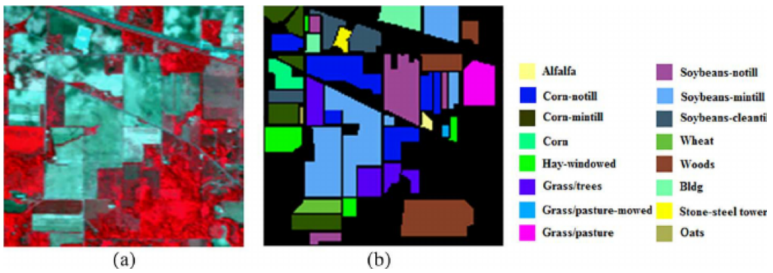


Fig. 2. Indian Pines hyperspectral image: (a) Composite image of hyperspectral data; (b) Ground-truth map

We compare the proposed algorithm with conventional SVM using Euclidean metric, under similar conditions (the parameters are chosen as recommended default values in Matlab environment; RBF kernel with $\sigma = 10$ gives the best results). Table 1 shows the accuracy of classification (rate of correctly predicted class labels) on test sample for some of the noise parameters. The running time on a dual-core Intel Core i5 processor with a clock frequency of 2.8 GHz and 4 GB RAM is about 50 sec in average for KCCE and 14 sec for SVM (note that an unoptimized code is used in KCCE implementation, in contrast with efficient implementation of SVM). One can see that KCCE has revealed itself as more noise resistant than SVM, especially under large distortion rates.

Table 1. Accuracy of KCCE and SVM on Indian Pines hyperspectral image

Noise parameters r, p	$0.05, 0.05$	$0.1,\ 0.1$	$0.15, 0.15$	$0.2, 0.2$
KCCE accuracy	0.777	0.742	0.720	0.686
SVM accuracy	0.767	0.618	0.543	0.503

7 Conclusion

In this work, we have introduced a supervised classification algorithm using a combination of ensemble clustering and kernel based classification. In the clustering ensemble, we used a scheme of a single clustering algorithm that constructs base partitions with parameters taken at random. It was verified that the weighted co-association matrix obtained with a clustering ensemble is a valid kernel matrix. The proposed combined approach experimentally has been proven to be successful when comparing with Support Vector Machine and Kernel Fisher Discriminant. Monte-Carlo experiments demonstrated that there exist examples of data distribution for which the proposed method gets significantly more accurate predictions. The experiment with a real hyperspectral satellite image has shown that the suggested algorithm is more accurate than SVM under noise distortion.

In the future, we plan to continue working under improving the performance of the suggested approach. For example, it will be useful to filter out points with unstable clusterings before using the kernel classifier. We expect that applying cluster ensemble with optimized weights [12] will further improve the accuracy. It will be interesting to apply the introduced approach for the solution of other types of machine learning problems such as regression or transfer learning.

Acknowledgement. The work was carried out according to the scientific research program "Mathematical methods of pattern recognition and prediction" in the Sobolev Institute of mathematics SB RAS. The research was partly supported by RFBR grant 18-07-00600 and partly by the Russian Ministry of Science and Education under the 5-100 Excellence Programme.

References

1. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press (2004)
2. Zhuravlev, Y.I.: Principles of construction of justification of algorithms for the solution of badly formalized problems. Mathematical Notes of the Academy of Sciences of the USSR. 23(6), 493–501 (1978)
3. Schapire, R., Freund, Y., Bartlett, P., Lee, W.: Boosting the margin: a new explanation for the effectiveness of voting methods. Annals of Statistics 26(5), 1651–1686 (1998)
4. Breiman, L.: Random forests. Machine Learning 45(1), 5–32 (2001)
5. Kuncheva, L.: Combining Pattern Classifiers. Methods and Algorithms. Wiley, NJ (2004)
6. Ajdarkhanov, M.B., Amirgaliev, E.N., La, L.L.: Correctness of algebraic extensions of models of classification algorithms. Kibernetika i Sistemnyj Analiz 5, 180–186 (2001)
7. Jain, A.K.: Data clustering: 50 years beyond k-means. Pattern Recognition Letters 31(8), 651–666 (2010)
8. Ghosh, J., Acharya, A.: Cluster ensembles. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 1(5), 305–315 (2011)
9. Vega-Pons, S. Ruiz-Shulcloper, J.: A survey of clustering ensemble algorithms. IJPRAI 25(3), 337–372 (2011)
10. Fred, A., Jain, A.: Combining multiple clusterings using evidence accumulation. IEEE Transaction on Pattern Analysis and Machine Intelligence 27, 835–850 (2005)
11. Gray, R.M.: Vector quantization. IEEE ASSP Magazine 1(2), 4–29 (1984)
12. Berikov, V.: Weighted ensemble of algorithms for complex data clustering. Pattern Recognition Letters 38, 99–106 (2014)
13. Berikov, V.: Cluster ensemble with averaged co-association matrix maximizing the expected margin. In: 9th International Conference on Discrete Optimization and Operations Research and Scientific School (DOOR 2016). pp. 489–500 (2016)
14. Berikov, V., Pestunov, I.: Ensemble clustering based on weighted co-association matrices: Error bound and convergence properties. Pattern Recognition 63, 427–436 (2017)
15. Zhuravlev, Y.I., Yunusov, R.: A method of improving the taxonomy algorithm by pattern recognition methods of the voting type. USSR Computational Mathematics and Mathematical Physics 11(5), 327–333 (1971)
16. Jaing, M., Tseng, S. and Su, C.: Two-phase clustering process for outlier detection. Pattern Recognition Letters 22(67), 691–700 (2001)
17. Rahman, A., Verma, B.: Cluster-based ensemble of classifiers. Expert Systems 30, 270–282 (2013)
18. Chapelle, O., Zien, A., Scholkopf, B. (Eds.): Semi-Supervised Learning. MIT Press (2006)
19. Chapelle, O., Weston, J., Scholkopf, B.: Cluster kernels for semi-supervised learning. Adv. Neural Inf. Process. Syst. 15, 601-608 (2002)
20. Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In Advances in Neural Information Processing Systems 14 (2001)
21. Balcan, M.F., Blum, A., Srebro, N.: A theory of learning with similarity functions. Machine Learning 72, 89–11 (2008)
22. Iam-On, N., Boongoen, T.: Diversity-driven generation of link-based cluster ensemble and application to data classification. Expert Systems with Applications 42(21), 8259–8273 (2015)

23. Dietterich, T., Bakiri, G.: Solving multiclass learning problems via error-correcting output codes. Journal of Artificial Intelligence Research 2, 263-282 (1995)
24. Platt, J.: Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. Advances In Kernel Methods - Support Vector Learning (1998)
25. Raudys, S.: Statistical and Neural Classifiers: An integrated approach to design. London: Springer-Verlag (2001)
26. Hyperspectral Remote Sensing Scenes: http://www.ehu.es/ccwintco/index.php/HyperspectralRemote SensingScenes, [On-line; accessed 09-February-2018]