Cluster Ensemble with Averaged Co-Association Matrix Maximizing the Expected Margin

Vladimir Berikov[1,2]

[1] Sobolev Institute of mathematics, Novosibirsk, Russia
[2] Novosibirsk State University, Novosibirsk, Russia
berikov@math.nsc.ru
http://www.math.nsc.ru

Abstract. The problem considered is cluster analysis with usage of the ensemble approach. The paper proposes a method for finding optimal weights for the averaged co-association matrix applied to the construction of the ensemble partition. The main idea is to find such weights for which the expectation of ensemble margin takes its maximum value. A latent variable pairwise classification model is used for determining margin characteristics dependent on cluster validity indices. To construct the ensemble partition, we apply minimum spanning tree found on the averaged co-association matrix as an adjacency matrix. The efficiency of the method is confirmed by Monte-Carlo simulations with artificial data sets.

Keywords: cluster ensemble, weighted co-association matrix, latent variable model, cluster validity index, margin expectation, minimum spanning tree

1 Introduction

The problem of cluster analysis consists in finding a partition $P = \{C_1, \ldots, C_K\}$ of data set $A = \{a_1, \ldots, a_n\}$ on a relatively small number of homogeneous clusters. The result can be either hard (each data object is attributed to one cluster) or fuzzy (for each object, a cluster membership function is defined); this paper considers hard clustering only.

Let the data sample be described with a table $\mathbf{X} = (x_{i,m})$, where $x_{i,m} = X_m(a_i) \in \mathbb{R}$ is a value of feature X_m for object a_i ($m \in \{1, \ldots, d\}$, $i \in \{1, \ldots, n\}$), d is feature space dimensionality. As a criterion of homogeneity, one may understand certain functional dependent on the scatter of observations within groups and the distances between clusters. The number of clusters is either a predefined parameter or should be found in a best way. In the given work it is assumed that the value of K should be determined automatically.

It is well-known that clustering is a NP-hard problem essentially [1,9]. Because of large computational complexity of the search process, a majority of existing algorithms [3]-[5] apply approximate iterative procedures to find locally optimal partition. In each

Copyright © by the paper's authors. Copying permitted for private and academic purposes.

In: A. Kononov et al. (eds.): DOOR 2016, Vladivostok, Russia, published at http://ceur-ws.org

step of the algorithm, the current partition is updated to improve the quality functional. As a rule, the process is guided by certain user-specified parameters.

Collective decision making (ensemble clustering) is a comparatively new approach in cluster analysis [6,7]. Following the ensemble framework, a number of base partitions are obtained firstly with the collection of different algorithms (or with a single algorithm under different parameters or other working settings). On the second step, the given variants are combined to yield the final consensus partition. The ensemble approach, as a rule, allows one to increase the stability of clustering results in case of uncertainty on data model or when it is not clear which of algorithm's parameters are most appropriate for a particular problem.

A number of ways to find the ensemble solution exist. In the co-association (CA) matrix based approach, each pair of objects $a_i, a_j \in A$ are associated with the matrix element indicating how often the objects are assigned to different clusters over all partition variants. This value is considered as analog of distance between a_i and a_j. To construct the final partition, any procedure based on a pairwise distance matrix can be used, for example, hierarchical agglomerative clustering (HAC) algorithm.

Some authors suggest *weighted* cluster ensemble aimed at improving its characteristics. The weights depend on the estimated quality (cluster validity indices, diversity measures, etc.) of base partitions [8,9].

The work [10] considerers weighted cluster ensemble composed of different algorithms. For determining the ensemble solution, it is proposed to use weighted averaged CA matrix, where the weights are found according to the principle of minimum variance of ensemble margin (closely related to the maximization of the expected margin). This principle is substantiated by the analysis of probabilistic properties of cluster ensembles in the framework of *a latent variable pairwise classification* (LVPC) model. The suggested method (Pairwise Weighted Ensemble Clustering, PWEC) has demonstrated an ability to find complex data structures under noise conditions; however, it considers only variations of cluster labels and disregards potentially useful information on other characteristics of base partitions such as cluster validity indices. Another disadvantage is that it can not automatically determine the number of clusters.

The proposed work aims at eliminating the limitations mentioned above: we suggest a method that

a) assigns weights taking into account both stability measures of base clusterings and the obtained quality estimates;

b) finds the optimal number of clusters using minimum spanning tree (MST) algorithm.

The rest of the paper is organized as follows. The second section briefly introduces the necessary notions. In the third section we describe the concept of ensemble margin and obtain an expression for the expected margin with use of LVPC model. This section also suggests a solution to the problem of assessing the characteristics of the ensemble and finding the optimal weights which give the maximum value to the expected margin. In the forth section we formulate the algorithm for constructing the ensemble clustering partition with MST approach. The fifth section presents the results of numerical experiments using Monte-Carlo simulations with data sampled from a given probability distribution. The conclusion summarizes the work.

2 Basic concepts

This section describes the method of ensemble clustering with weighted CA matrix and provides examples of cluster validity indices used in this work.

2.1 Cluster ensemble with usage of weighted co-association matrix

In the framework of the ensemble approach, we consider a collection of different clustering algorithms $\mu_1, \ldots, \mu_M$ applied to data set A. Suppose that each algorithm μ_m, $m = 1, \ldots, M$ is enabled to work a number of times under different conditions such as initial centroids coordinates, subsets of features, number of clusters, etc. In each lth run, it generates a partition composed of $K_{l,m}$ clusters, where $l = 1, \ldots, L_M$, L_M is a given number of runs for algorithm μ_m.

The quality of each variant is evaluated with some criterion (e.g., cluster validity index) $\gamma_{l,m}$. It is allowable to suppose that the indices are properly standardized so that a) $0 \leq \gamma_{l,m} \leq 1$, and b) the more compact and distant are the found clusters, the larger is the index value.

For a pair of different objects a_i, a_j, we define the value

$$h_{l,m}(i,j) = \mathbb{I}[\mu_{l,m}(a_i) \neq \mu_{l,m}(a_j)],$$

where $\mathbb{I}[\cdot]$ is indicator function: $\mathbb{I}[true] = 1$; $\mathbb{I}[false] = 0$; $\mu_{l,m}(a)$ is a cluster label assigned by algorithm μ_m to object a in lth run.

The averaged weighted CA matrix $\mathbf{H} = (\bar{h}(i,j))$ is calculated over all found variants, taking into account validity indices. We set matrix elements as

$$\bar{h}(i,j) = \sum_{m=1}^{M} \sum_{l=1}^{L_m} w_{l,m}(i,j)\, h_{l,m}(i,j), \tag{1}$$

where $w_{l,m}(i,j) \geq 0$ are weight constants. The weights should account for the degree of confidence to the results of cluster assignments for a given pair. In the suggested algorithm (see below) the weights depend on validity indices and other characteristics of the ensemble.

It is easy to prove that $\mathbf{H}$ defines a pseudo metric on A, thus matrix elements can be considered as analogs of distances between pairs of objects. To find the final partition of A on a predefined number of clusters using matrix $\mathbf{H}$, one can apply HAC algorithm with average linkage rule for the definition of between-group distances.

2.2 Cluster validity indices

Cluster validity indices give one a possibility to compare the obtained partition with some "etalon" variant (*external indices*); or to assess the result of clustering with various measures of cluster compactness and separability (*internal indices*). A majority of existing indices require quadratic time for their calculation; however, indices of linear complexity exist.

External indices estimate the degree of correspondence between two partitions $P_1 = \{C_{1,1}, \ldots, C_{k,1}, \ldots, C_{K_1,1}\}$ and $P_2 = \{C_{1,2}, \ldots, C_{l,2}, \ldots, C_{K_2,2}\}$, where $C_{k,1} = \{a_{i_1}, \ldots, a_{i_{N_{k,1}}}\}$, $C_{l,2} = \{a_{j_1}, \ldots, a_{j_{N_{l,2}}}\}$, $N_{k,1}$ is a number of objects in kth cluster of the first partition, $N_{l,2}$ is a number of objects in lth cluster of the second partition.

The *Rand index* is defined as

$$\varphi_R(P_1, P_2) = (S + D)/G_0,$$

where S is a number of pairs belonging to the same cluster in P_1 and P_2, D is a number of pairs belonging to different clusters, $G_0 = \binom{n}{2}$ is total number of pairs. The index belongs to interval $[0, 1]$ and defines the portion of correctly classified pairs; the value $\varphi_R = 1$ indicates perfect agreement between two partitions.

The *adjusted Rand index* [11] is defined with the following expression:

$$\varphi_{AR}(P_1, P_2) = \frac{\sum\limits_{k,l} \binom{N^{(k,l)}}{2} - Q_1 Q_2 / G_0}{\frac{1}{2}(Q_1 + Q_2) - Q_1 Q_2 / G_0},$$

where $Q_1 = \sum\limits_k \binom{N_{k,1}}{2}$, $Q_2 = \sum\limits_k \binom{N_{l,2}}{2}$ and $N^{(k,l)} = \|C_{k,1} \cap C_{l,2}\|$. The maximum value of φ_{AR} is 1; the index may take negative values. The value close to 0 indicates that the obtained cluster partition is mostly probably accidental.

Internal indices are determined for a single partition $P = \{C_1, \ldots, C_K\}$. This work uses two examples of internal indices: Hubert Gamma index and cophenetic correlation coefficient.

Hubert Gamma index, $\Gamma(P)$, is specified as the linear correlation coefficient between elements of matrices $R(i, j)$ and $U(i, j)$ $(i < j)$, where R is a matrix of pairwise Euclidian distances between data points, U is unweighed co-association matrix: $U(i, j) = 0$ if $\exists C_q : a_i, a_j \in C_q$; otherwise $U(i, j) = 1$. An increase in index value typically indicates that the quality of the partition is improving.

Cophenetic correlation coefficient, $Coph(P)$, is defined with a dendrogram of clustering partition. The dendrogrammatic distance $\tau(i, j)$ between a_i, a_j is the height of the node at which these two objects are first joined together. $Coph(P)$ equals the linear correlation coefficient between the Euclidian distances $R(i, j)$ and dendrogrammatic distances $\tau(i, j)$ over all i, j $(i < j)$.

Both $\Gamma(P)$ and $Coph(P)$ belong to interval $[-1, 1]$; it will be more convenient for us to replace their negative values with zeros.

3 Margin of cluster ensemble

Suppose that data sample is a mixture of a finite number of components (classes). A latent groundtruth variable Y defines a class number to which an object belongs to. Matrix Z with elements

$$Z(i, j) = \mathbb{I}[\, Y(a_i) \neq Y(a_j) \,], \tag{2}$$

where a_i and a_j are arbitrary data objects, defines their true status (i.e., if a_i and a_j indeed belong to separate classes). Let us define *margin matrix* dependent on the

variants of cluster partitions. For a pair a_i, a_j, their margin is

$$mg(i,j) = \{\text{weighted number of votes for } Z(i,j) - \\ \text{weighted number of votes against } Z(i,j)\}$$

and can be written as

$$mg(i,j) = \sum_{m=1}^{M} \sum_{l=1}^{L_m} w_{l,m}(i,j) \left\{ I[h_{l,m}(i,j) = Z(i,j)] - I[h_{l,m}(i,j) \neq Z(i,j)] \right\}.$$

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:

$$mg(i,j) = \sum_{m=1}^{M} \sum_{l=1}^{L_m} w_{l,m}(i,j)(2Z(i,j) - 1)(2h_{l,m}(i,j) - 1). \tag{3}$$

Margin matrix can not be calculated if the true partition is unknown. However, it was shown in [10] that some of margin's characteristics can be assessed using a few basic assumptions on the statistical behavior of clustering algorithms. Furthermore, there was found an upper bound for error probability (at assigning objects in a pair to different clusters) and it was proved that an increase in the expected margin and a decrease in margin variance reduces the error resulting in improving the quality of cluster ensemble.

3.1 Conditional expectation of margin

Consider a specific form of (1): let each weight be a function of validity index and a quantity $\alpha_m(i,j) \geq 0$:

$$w_{l,m}(i,j) = \frac{\alpha_m(i,j)\ \gamma_{l,m}}{L_m}, \tag{4}$$

where $\sum_m \alpha_m(i,j) = 1$ for all i, j (in (4), we separate a component dependent on validity index).

Below we formulate basic assumptions which allow us assessing characteristics of cluster ensemble:
a) Data table $\mathbf{X}$ is arbitrary and fixed; for any pair of data objects a_i, a_j, chosen at random from A, their true status is a random value $Z(i,j)$ defined with (2).
b) Each algorithm μ_m is randomized, i.e. it depends on a random vector Ω_m from the set of parameters $\mathbf{\Omega}_m$, $m = 1, \ldots, M$. For data table $\mathbf{X}$ as input, μ_m is running L_m times with i.i.d. parameters $\Omega_{1,m}, \ldots, \Omega_{L_m,m}$ being statistical copies of Ω_m.

It should be noted that the i.i.d. condition must be ensured through a mechanism of the ensemble generation.

For any algorithm μ_m, the conditional probabilities of correct decisions (either separation or joining of a_i, a_j under fixed $Z(i,j)$) are denoted as

$$q_m^1(i,j) = P[h_m(i,j,\Omega_m) = 1 | Z(i,j) = 1], \tag{5}$$

$$q_m^0(i,j) = P[h_m(i,j,\Omega_m) = 0 | Z(i,j) = 0]. \quad (6)$$

The measure of cluster validity, obtained by algorithm μ_m on Ω_m, is considered as a random value $\gamma_m(\mathbf{X}, \Omega_m)$. Because the quality criterion is determined on the whole data set, one may understand this value as practically independent on quantities $Z(i,j)$, $h_{l,m}(i,j)$, etc., defined for a specific object pair.

`Proposition 1.` *Given the assumptions a), b) be valid, conditional mathematical expectation of ensemble margin for a_i, a_j under $Z(i,j) = z$ is:*

$$E_{\bar{\Omega}}[mg(i,j)|Z(i,j) = z] = \sum_m \alpha_m(i,j)\, E[\gamma_m]\, (2p_m(z;i,j) - 1),$$

where $\bar{\Omega} = (\Omega_{1,1}, \ldots, \Omega_{L_M,M})$, $E[\gamma_m] = E_{\Omega_m}[\gamma_m(\Omega_m)]$, $p_m(z;i,j) = (1-z)\, q_m^0(i,j) + z\, q_m^1(i,j)$.

`Proof.` Let us denote by

$$h_{l,m}(i,j,\Omega_{l,m}) = \mathbb{I}[\mu_m(i,\Omega_{l,m}) \neq \mu_m(j,\Omega_{l,m})]$$

the decision for a pair a_i, a_j, where $\mu_m(i,\Omega_{l,m})$ is a cluster label assigned to object a_i by algorithm μ_m in its lth run with parameter vector $\Omega_{l,m}$, and also denote by

$$h_m(i,j,\Omega_m) = \mathbb{I}[\mu_m(i,\Omega_m) \neq \mu_m(j,\Omega_m)]$$

the decision under vector Ω_m.

Until the proof end, let us skip arguments i, j for short: $mg(i,j) = mg$, $Z(i,j) = Z$, etc. From (3) and (4) we have:

$$E_{\bar{\Omega}}[mg|Z = z] = \sum_m \frac{\alpha_m}{L_m} \sum_l E_{\bar{\Omega}}[\gamma_{l,m}\,(\Omega_{l,m})(2z-1)(2h_{l,m}(\Omega_{l,m}) - 1)].$$

Because $\Omega_{l,m}$ and Ω_m are equally distributed and $\gamma_{l,m}$ is independent on $h_{l,m}$, it holds true that

$$E_{\bar{\Omega}}[mg|Z = z] = \sum_m \alpha_m E[\gamma_m(\Omega_m)]\, (2z-1)(2E_{\Omega_m}[h_m(\Omega_m)] - 1).$$

For $z = 0$ we have:

$$(2z-1)(2E_{\Omega_m}[h_m(\Omega_m)] - 1) = -(2P[h_m(\Omega_m) = 1|Z = 0] - 1) = 2q_m^0 - 1;$$

and for $z = 1$:

$$(2z-1)(2E_{\Omega_m}[h_m(\Omega_m)] - 1) = 2P[h_m(\Omega_m) = 1|Z = 0] - 1 = 2q_m^1 - 1.$$

Therefore

$$E_{\bar{\Omega}}[mg|Z = z] = \sum_m \alpha_m E[\gamma_m(\Omega_m)]\, (2p_m(z;i,j) - 1).$$

This completes the proof.

In this work we also assume that each algorithm μ_m creates a partition for which the probability of correct decision for any pair a_i, a_j is greater than 0.5, i.e.

$$q_m^0(i,j) > 0.5,\ q_m^1(i,j) > 0.5. \tag{7}$$

It means that clustering algorithms included into the ensemble possess at least slightly better classification accuracy than a trivial procedure based on random assignment of a pair to the same or different clusters.

3.2 Evaluation of the expected margin and its optimization

The main idea of our method is to control the behavior of cluster ensemble by changing weights $\alpha_1, \ldots, \alpha_M$ looking for the maximum conditional expectation of margin. The expression for the expected margin depends on a number of characteristics ($p_m(z;i,j)$, $E[\gamma_m]$) which should be evaluated from the results of base clustering algorithms, i.e., a number of partitions along with evaluated validity indices $\gamma_{l,m}$, $l = 1 \ldots, L_m$, $m = 1, \ldots, M$.

To evaluate $p_m(z;i,j)$, one may consider the obtained partitions and calculate co-association matrices with elements

$$\bar{h}_{l,m}(i,j) = \mathbb{I}[\mu_{l,m}(a_i) \neq \mu_{l,m}(a_j)].$$

For each (i,j), the division frequency is $\bar{P}_m(i,j) = \dfrac{1}{L_m} \sum\limits_l \bar{h}_{l,m}(i,j)$. Each $\bar{P}_m(i,j)$ is an estimate of conditional probability

$$P[h_m(i,j,\Omega_m) = 1 | Z(i,j) = z]$$

depending on the realization of $Z(i,j)$. Given the assumption (7) be valid, one may assess $p_m(z;i,j)$ with quantity

$$\bar{p}_m(i,j) = \max(\bar{P}_m(i,j), 1 - \bar{P}_m(i,j)).$$

This value can be interpreted as an estimate of *local stability* of cluster decisions for a pair a_i, a_j.

To evaluate theoretical means $E[\gamma_m]$, $m = 1, \ldots, M$, one may use sample averages of validity indices $\bar{\gamma}_m = \dfrac{1}{L_m} \sum\limits_{l=1}^{L_m} \gamma_{l,m}$.

Let us consider a problem of maximization of the expected margin:

$$\text{for each } i,j\ (i<j),\ \text{find } \alpha_1(i,j), \ldots, \alpha_M(i,j):$$

$$\sum_m \alpha_m(i,j)\bar{\gamma}_m(2\bar{p}_m(i,j) - 1) \ \rightarrow \max_{\alpha_1(i,j),\ldots,\alpha_M(i,j)},$$

$$\text{s.t. } \alpha_1(i,j) \geq 0, \ldots, \alpha_M(i,j) \geq 0,\ \sum_m \alpha_m(i,j) = 1.$$

The solution to this linear programming problem is rather straightforward:

$$\alpha^*_{m^*}(i,j) = \begin{cases} 1, & m^* = arg \max\limits_{m=1,\ldots,M} \{\bar{\gamma}_m \bar{p}_m(i,j)\}; \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

As one can see, in the resulting optimization strategy only the best algorithm from the ensemble (considering quality and stability estimates) is taken into account. For different pairs a_i, a_j, different algorithms may come out on top.

4 Finding ensemble partition with minimum spanning tree

Let $G = (V, E)$ be an undirected graph with non-negative edge weights w, where V is set of vertices (in our case, $V = A$, i.e., the set of data objects); E is the set of edges. Matrix $w = (w_{i,j})$ defines weights of edges (a_i, a_j), where $i, j = 1, \ldots, n$. As a weight matrix, we consider the averaged CA matrix $\mathbf{H}$ with elements $\bar{h}(i,j)$ defined according to (1), (4), (8).

A spanning tree is a tree with $n-1$ edges, i.e. a tree that connects all the vertices. The total weight of a spanning tree T is defined as $w_T = \sum_{e \in T} w(e)$. A minimum spanning tree is a tree of minimum total weight.

A large number of methods for MST construction exist [12]. Their running time is of polynomial order (in some special cases, near to linear).

In a graph-theoretic framework, computationally efficient MST-based algorithms are widely used in cluster analysis [13]. Once the MST T is built for a given data set, first $k-1$ edges with largest weights are removed, which creates a partition P_k of A on k clusters. The obtained variant is evaluated with an objective function; the best partition for different $k \in \{2, \ldots, K_{\max}\}$ forms an output ($K_{\max}$ is algorithm's parameter). MST clustering algorithms are capable of finding complex non-spherical clusters; they can automatically determine the number of clusters using a predefined quality functional.

In this paper, we apply well known Kruskal's algorithm [14] for finding MST for data set A and matrix $\mathbf{H}$. Using the obtained MST, the final ensemble partition is created. As a quality qriterion, we use the following functional:

$$F(P_k) = k \prod_{c=1}^{k} N_c^{w_c / w_T}, \tag{9}$$

where N_c is size of cth cluster, w_c is total weight of edges which enter into cth cluster.

MST-based ensemble clustering algorithm (MSTEClust) has the following main steps.

Algorithm MSTEClust.
Input:
$A = \{a_1, \ldots, a_n\}$: dataset described with data table $\mathbf{X}$;
$K_{\max}$: maximum number of clusters;
$L_1, \ldots, L_M$: number of runs for base algorithms $\mu_1, \ldots, \mu_M$;

$\mathbf{\Omega_1}, \ldots, \mathbf{\Omega_M}$: sets of allowable parameters (working conditions) of algorithms.
Output:
partition of A into optimal number K of clusters ($2 \leq K \leq K_{\max}$).
Steps:
1. Generate $L_1, \ldots, L_M$ variants of cluster partition with algorithms $\mu_1, \ldots, \mu_M$ for randomly chosen working parameters; calculate validity indices $\gamma_{l,m}$, $l = 1, \ldots, L_m, m = 1, \ldots, M$.
2. Calculate averaged co-association matrix $\mathbf{H}$ with optimal weights according to (1), (4), (8).
3. Construct MST on graph $(V, G) = (A, \mathbf{H})$ using Kruskal's algorithm.
4. Find a partition of A on optimal number of clusters K using MST and criterion (9).
end.

The running time of steps 1-2 linearly depends on the complexity of base clustering algorithms and validity estimators. Constructing of the final partition needs $O(n \log(n))$ operations. One may conclude that MSTEClust is more readily applicable to large amount of data than PEWEC [10] which has time complexity of order $O(n^2)$.

5 Numerical experiments

This section describes numerical experiments with MSTEClust algorithm with Monte Carlo modeling. In the experiments, two base clustering algorithms are included into the ensemble: k-means (KM) and hierarchical agglomerative clustering algorithm with single linkage rule (HACSL).

We choose the following distribution model. In 16-dimensional feature space four classes are defined. First and second classes are of normal distribution $N(\nu_1, \Sigma)$, $N(\nu_2, \Sigma)$, where $\nu_1 = (0, ..., 0)^T$, $\nu_2 = (6, ..., 6)^T$, $\Sigma = \sigma\mathbf{I}$ is diagonal covariance matrix, $\sigma = 1$. The coordinates of objects from other two classes are determined recursively: $x_{k_{i+1}} = x_{k_i} + \theta_1 \cdot \mathbf{1} + \theta_2 \cdot \varepsilon$, where $\mathbf{1} = (1, \ldots, 1)^T$, $\theta_1 = \theta_2 = 0.25$, ε is Gaussian random vector $N(0, \mathbf{I})$, $k = 3, 4$. For class 3, $x_{3_1} = (-6, 6, ..., -6, 6)^T + 0.1 \cdot \varepsilon$; for class 4, $x_{4_1} = (6, -6, ..., 6, -6)^T + 0.1 \cdot \varepsilon$. The number of objects of each class equals 25. Figure 1 illustrates sampled data.

The random subspace method is used for the ensemble construction: each base cluster partition is built on d_{ens} randomly chosen variables ($d_{ens} = 3$ in this experiment). Besides that, for each base algorithm the number of clusters is chosen at random from range $\{2, \ldots, K_{\max}\}$, $K_{\max} = 5$. The obtained clustering results are assessed with cluster validity indices: Hubert Gamma index estimates the quality of KM; the results of HACSL are evaluated with cophenetic correlation index. The number of ensemble variants for each algorithm is set to $L_1 = L_2 = 25$.

To study the behavior of the ensemble algorithm in the presence of noise, some of the features, whose indexes are determined by chance and their total number equals parameter d_0, are replaced with random values uniformly distributed over feature range.

In the process of Monte Carlo modeling, artificial data sets are repeatedly generated according to the specified distribution model. For each data set, a number of base partitions are obtained by KM and HACSL; the ensemble solution is found according to MSTEClust.

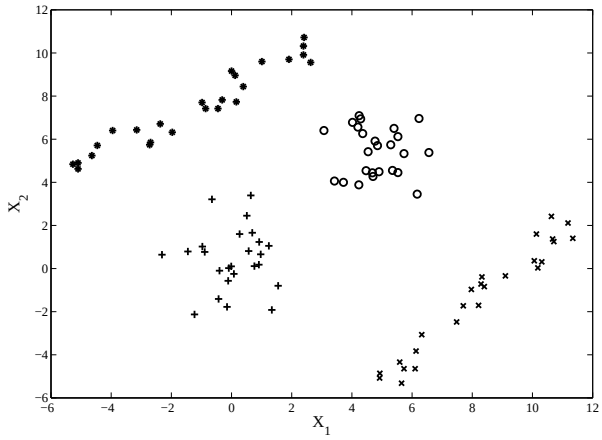

Fig. 1. An example of sampled data (4 classes marked with $\times$, $\circ$, $+$, $*$) ; projection on axes X_1, X_2.

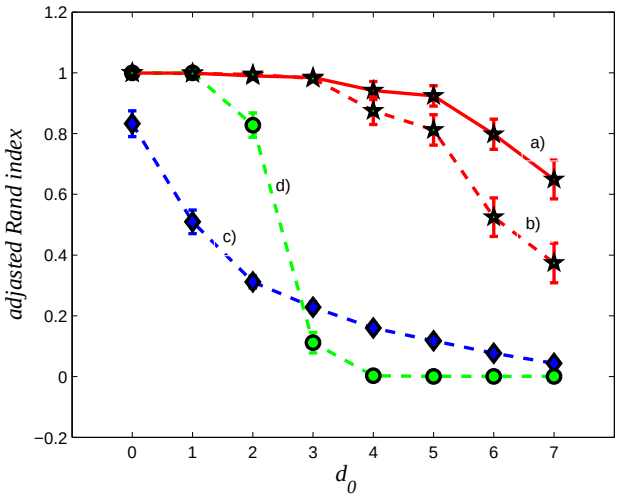

Fig. 2. Dependence of averaged *ARI* from the number of noise features d_0: a) MSTEClust, b) PEWEC, c) KM, d) HACSL. 95% confidence intervals for estimated characteristic are also shown.

Figure 2 presents the results of the experiments (Adjusted Rand index averaged over 100 random samples). For comparison purposes, the results of KM and HACSL (for which the true number of clusters is given) are plotted as well. To illustrate the effect of weights optimization taken validity indices into account, Figure 2 displays the

evaluations of the performance of PEWEC [10] modification in which the MST is used to construct the ensemble partition instead of a dendrogram algorithm.

The plots are positive evidence for the suggested ensemble algorithm with optimized weights. Beginning with sufficiently large number of noise features, MSTECLust gives significantly better clustering quality than other examined procedures. In contrast, non-ensemble algorithms totally fail to recover heavily distorted data structure.

An important problem is experimental verification of theoretical suppositions which lie at the basis of the suggested method. The validity of assumption (7) is crucial because it determines the way of assessing the expected margin. In Monte Carlo simulation, the true status of each object pair is known, thus it is possible to evaluate conditional probabilities $q_m^1(i,j)$, $q_m^0(i,j)$ in (5),(6) using available frequencies. In the experiment, all above-described settings are unchanged. Table 1 presents the results of estimations. From this table, one can see that in the condition of small and moderate noise (up to three noise features) the estimates stay above 0.5, thus confirming assumption (7). Under increasing noise, violation of the inequalities is seen, evidently resulting in the decrease of the algorithm's accuracy.

Table 1. Averaged estimates of $q_m^1(i,j) = P(h_m = 1|Z(i,j) = 1$ and $q_m^0(i,j) = P(h_m = 0|Z(i,j) = 0)$. Averaging is done over all $i < j$ and 100 generated data samples under given number of noise features d_0. For KM, $m = 1$; for HACSL, $m = 2$. Standard deviations of the estimated quantities are shown in parenthesis.

	d_0				
	0	1	2	3	4
$q_1^0(\cdot,\cdot)$	0.957 (0.008)	0.862 (0.023)	0.773 (0.029)	0.698 (0.03)	0.630 (0.026)
$q_1^1(\cdot,\cdot)$	0.746 (0.017)	0.723 (0.015)	0.703 (0.017)	0.681 (0.015)	0.659 (0.013)
$q_2^0(\cdot,\cdot)$	0.99 (0.003)	0.969 (0.01)	0.949 (0.013)	0.936 (0.014)	0.923 (0.013)
$q_2^1(\cdot,\cdot)$	0.716 (0.021)	0.642 (0.028)	0.575 (0.035)	0.506 (0.034)	0.433 (0.043)

Conclusion

In this work we have suggested a method of ensemble clustering using a number of different base algorithms. The ensemble partition is found by the weighted average of co-association matrices, where the weights depend on cluster validity indices. A mathematical methodology of determining optimal weights is proposed. As an optimization functional, we utilize the estimate of the ensemble's expected margin. To construct the final partition, it is suggested to use minimum spanning tree built on the averaged co-association matrix as an adjacency matrix.

Unlike other existing methods of ensemble clustering, the proposed one takes into account both stability measures of base partitions and the obtained quality estimates; it is capable of finding the optimal number of clusters.

The efficiency of the suggested MSTEClust algorithm is confirmed experimentally with Monte-Carlo modeling. For a given distribution model, the simulations under

noise distortions have demonstrated significant improvement of clustering quality for MSTEClust in comparison with other considered algorithms. Monte-Carlo experiments have confirmed the principal feasibility of theoretical assumptions used in this work for the estimation of cluster ensemble characteristics.

In the future works, the author plans to continue studying theoretical properties of clustering ensembles and developing algorithms for real world applications such as hyperspectral images analysis and segmentation.

Acknowledgements

This work was partially supported by the Russian Foundation for Basic Research, project 14-07-00249a.

References

1. Falkenauer E.: Genetic Algorithms and Grouping Problems. Wiley, New York (1998)
2. Naldi M., Campello R., Hruschka E., Carvalho A.: Efficiency of evolutionary k-means. Applied Soft Computing. 11, 1938–1952 (2011)
3. Duda R.O., Hart P.E., Stork D.G.: Pattern Classification. Second Edition. Wiley, New York (2000)
4. Jain A.K., Dubes R.C.: Algorithms for clustering data. Prentice Hall, NJ (1988)
5. Jain A.K.: Data clustering: 50 years beyond k-means. Pattern Recognition Letters. 31 (8), 651–666 (2010)
6. Ghosh J., Acharya A.: Cluster ensembles. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. Vol. 1, 305–315 (2011)
7. Vega-Pons, S. Ruiz-Shulcloper, J.: A Survey of Clustering Ensemble Algorithms. IJPRAI 25(3), 337–372 (2011)
8. Vega-Pons, S., Ruiz-Shulcloper, J.: Clustering Ensemble Method for Heterogeneous Partitions. CIARP'09, 481–488 (2009)
9. Naldi, M. C., Carvalho, A. C. P. L. F., Campello, R. J. G. B.: Cluster ensemble selection based on relative validity indexes. Data Mining and Knowledge Discovery. Vol. 27, 259—289 (2013)
10. Berikov V.: Weighted ensemble of algorithms for complex data clustering. Pattern Recognition Letters. Vol. 38, 99–106 (2014)
11. Hubert L., Arabie P.: Comparing partitions. Journal of Classification. Vol. 2, 193–218 (1985)
12. Gabow, H. N., Galil, Z., Spencer, T., Tarjan, R. E.: Efficient algorithms for finding minimum spanning trees in undirected and directed graphs. Combinatorica. Vol. 6(2), 109 – 122 (1986)
13. Gower J., G. Ross.: Minimum spanning trees and single linkage cluster analysis. Applied Statistics. Vol. 18, 54-–64 (1969)
14. Kruskal J.: On the shortest spanning subtree and the traveling salesman problem. Proceedings of the American Mathematical Society, pp. 48–50 (1956)