Clustering combination based on co-association matrices

Combinaison des clusterings basée sur les matrices de co-association

Ivan O. Kyrgyzov
Henri Maître
Marine Campedel

2008C002

Septembre 2008

Département Traitement du Signal et des Images
Groupe TII : Traitement et Interprétation des Images

Clustering combination based on co-association matrices

Combinaison des clusterings base sur les matrices de co-association

Ivan O. Kyrgyzov *, Henri Maître, Marine Campedel

Competence Centre for Information Extraction and Image Understanding for Earth Observation,
Institut TELECOM, TELECOM ParisTech, LTCI CNRS,
46, rue Barrault, 75013, Paris, France

Abstract

A new method of "clustering combination" is presented in this paper the purpose of which is to benefit from several clusterings made in parallel in a previous stage. The guideline of the proposed combination is to group data samples which appear frequently in the same cluster. First, we develop a hierarchical algorithm to optimise the objective function which qualifies the grouping. The algorithm is competitive compared to existing combination algorithms but in spite of its good results it does not guarantee the convergence to a global unique solution. Based on the analysis of the objective function a second method is proposed which provides a global solution with a guaranteed convergence. This combination is expressed as the minimisation of the square distances among samples. We prove in this paper that the global minimum may be found using the gradient density function estimation by the mean shift procedure. Local optimal modes of this function form groups of samples and consequently constitute a global solution of combination. Advantages of this method are a fast convergence and a linear complexity. The combination of different clusterings is performed on synthetic as well as real data-bases and shows the effectiveness of the proposed method and its superiority with respect to others combination approaches.

Key words: Clustering combination, Co-association matrix, Least square error, Mean shift

Résumé

Dans ce rapport, nous proposons d'étudier les solutions de combinaison de clustering en utilisant la matrice de co-association. Nous présentons également de nouvelles méthodes pour la combinaison afin d'éviter les inconvénients des approches existantes. L'idée de la combinaison proposée est de regrouper les échantillons qui sont dans le même cluster dans la plupart des clusterings. Tout d'abord, nous montrons une fonction objective pour combiner différentes clusterings. Ensuite, nous développons un algorithme hiérarchique pour optimiser la fonction objective. Un tel algorithme est compétitif par rapport aux autres algorithmes de combinaison, mais en dépit de ses très bons résultats, il ne garantit pas la convergence vers la solution globale. Après une analyse de la fonction objective, nous proposons une méthode améliorée qui donne la solution globale. De plus, nous décrirons les conditions d'une telle convergence. L'un des avantages d'une telle méthode est que l'algorithme a la convergence rapide et la complexité linéaire.

Mot-Clés : Combinaison des clusterings, Matrices de co-association, Erreur quadratique, Mean shift

1 Introduction

Clustering algorithms are one of the basic tools in pattern recognition. They are used for data mining in unsupervised learning tasks [1]. Clustering consists in obtaining groups of similar samples for further data exploration or retrieval in supervised or semi-supervised classification [2].

It is a common practice that several clustering steps are performed in parallel, either because different algorithms are used, or, because different parameters of the same algorithm provide complementary results which may be profitable [3–15].

For supervised classification there are two main approaches to infer one single classification from multiple ones: (i) selecting the best classification and (ii) combining classifications [16]. The first approach is rather applied to classify a particular type of data for a given class model when strong evidence exists on the kind of result we expect. The second approach has become popular (e.g., AdaBoost [16,17]) for large data sets, complex and multiple classification criteria and unmodelled data. The strong interest in combination techniques rather than in classification selection is due to their ability to better take

* Corresponding author. Tel.:+33 1 45 81 76 40; fax: +33 1 45 81 37 94.
Email addresses: kyrgyzov@enst.fr (Ivan O. Kyrgyzov), maitre@enst.fr (Henri Maître), campedel@enst.fr (Marine Campedel).

benefit from the many different classifications of the complex data. These two approaches concern supervised tasks when classes are a priori known [18].

For unlabelled data and unknown classes, unsupervised clustering is preferably used. Clustering is generally carried out on numerous non organised data when unknown clusters have to be estimated [1,16]. It is often the case for satellite or multimedia image indexing. The purpose of clustering is to group data to facilitate their interpretation by a user. But often the exact purpose of the user is not known [19]. Moreover, there are hundreds of developed clustering algorithms. Some methods make the hypotheses that data are almost Gaussian distributions and therefore look for dense kernels [20]. Other methods are looking for discriminant features or combination of the features which provide boundaries between clusters [16]. There are also methods which suppose the data to be hierarchically dependent [2]. Some other are only based on the local proximity [20]. For each of these hypotheses, some algorithms are known as to be efficient. When using these efficient algorithms, we obtain a set of clusterings which will hopefully well respect the diversity and the richness of the classifications we may obtain from the data. Clustering algorithms are pertinent with respect to a given criterion, but none is absolutely superior since no measure exists to qualify an absolute quality [21]. Selecting a good classification method among hundreds proposed without a unique and exact objective function is a difficult problem [19].

Our paper refers to the problem of unsupervised clustering combination. In this case, in general, there is no correspondence among clusters in different clusterings, contrary to supervised classifications. Moreover, the number of clusters may be different from one clustering to another.

At this point the main difficulty is to determine a judicious criterion in order to combine elementary clusterings and to obtain a final clustering solution. Another problem is how to efficiently implement the chosen method in case of very large data-bases. The contribution of this paper is to address these two issues.

Many different methods may be used in order to merge information issued from different clustering [3–15,22]. But, here, we pay attention to methods which are based on the property of two samples to belong or not to the same cluster, depending on the clustering. A review of these methods is given in Section 2; the combination criterion that we chose is formulated in Section 3, with some mathematical developments which make it easy to manipulate; Section 4 describes the proposed algorithm along with an improvement to efficiently process large data sets. In Section 5, a global optimum of the combination criterion formulated in Section 3 is exactly found by using iterative mean shift. Results on toy as well as real world data are presented in Section 6, which also provide a comparison with existing methods, and demonstrates the efficiency of this approach.

2 Clustering Combination

There exist many methods to aggregate information pieces issued from different clustering techniques. Classical approaches of clustering combination are based on Bayesian or Dempster-Shafer theories [23]. But these approaches have exponential computational complexity. Therefore, their application may be cumbersome for more than several clusterings of large data sets.

One of the most attractive clustering combination methods is based on the use of a co-association matrix [3–8]. An element of this symmetric square matrix is the number of occurrences of two samples in the same cluster depending on the clustering. The co-association matrix will be introduced in Section 3.

In [3,5] and later in [8], authors propose a methodology that inspired this paper to combine several clusterings. A number of clusterings are obtained by *K-means* algorithm with random initialisations and a random number of clusters. The co-association matrix is built by collecting the clusterings. A hierarchical single-link method is then applied to the matrix in order to group samples which appear the most frequently together. In [8], the final number of combined clusters is taken either as the one that corresponds to the longest lifetime on the dendrogram or as the one which provides the highest mutual information measure between the combined clusters and given clusterings. In this case, normalised mutual information (denoted *NMI*) [8] is the objective criterion of the method. It expresses a global quality of the final partition. This criterion is different from the distance criterion which was used in [4], and that we also use. Therefore we comment here on the approach [8], when the discussion on [3,4] is kept in Section 3. This method, only based on the frequency of association of different samples to the same cluster, is interesting for the user who does not need to care about the elementary clustering methods. It makes no assumption on the reasons for which samples have been grouped and does not question about the pertinence of the initial clustering stage. However it suffers from several limitations: (i) it requires some prior knowledge on the approximate number of clusters, (ii) it does not guarantee any optimality of the final classification and (iii) it may face storage and computational problems when dealing with large sample sets.

The first limitation comes from the initial clustering stage. If the number of initial clusters is sequentially increased from 2 to the number of samples, the co-association matrix tends to be a near diagonal matrix with small values out of diagonal. Therefore, the more clusters used to build the co-association matrix, the more clusters result from the combination. To limit this trend, following the method presented in [8], one should constrain the initial parameter of the *K-means* to values close from the targetted number of clusters, *e.g.,* as in [20,24].

The third limitation is due to the single-link algorithm used to obtain the combined clusters (or similarly to the complete-link or to the average-link algorithms which are proposed as alternatives in [8]). This algorithm requires the storage of the complete co-association matrix (or of its upper-part). In case of thousands of samples, this may create storage and computational difficulties.

To address the second limitation, the method proposed in [10] may be used. In order to optimise the final classification, the authors consider the clustering combination in the framework of finite mixture models of clustering ensembles and solve it according to the maximum likelihood criterion with the Expectation-Maximisation (EM) algorithm. Another solution to overcome this second limitation may be found in [9]. The authors propose the mutual information measure as an objective function, but optimise it with a greedy combinatorial algorithm. Unfortunately, its complexity is exponential in the number of samples. Both methods [10] and [9] require a predetermined number of final clusters. We propose a way to overpass this constraint in Sections 4.1 and 5.

In [14], clustering labels are combined jointly with a feature space of data. We do not consider such an approach here because it is often difficult to combine unambiguously criteria of different clustering algorithms. In addition, for this approach, several prior parameters should be tuned in order to combine clustering results. Ayad and Kamel [11] combine clusterings generated by *K-means* algorithm with the same predetermined number of clusters. The authors argue that the representation of clustering labels by a co-association matrix is cumbersome and propose to analyse a matrix of pairwise distances between clusters, instead. They find the correspondence between clusters from different clustering results. Then a group-average hierarchical clustering is applied in order to group elements of this matrix. In such a way, they always combine clusterings with the same number of clusters. The authors do not provide any objective function to estimate the quality of combination and the number of clusters after combination.

In [12], a matrix of sample associations is used in order to represent different clusterings. Then their combination is obtained by clustering this matrix. In this approach, the final number of clusters should be known a priori that imposes the first limitation given above. Lange and Buhmann [25] make use of a probabilistic model of the co-association matrix. The EM-algorithm optimises model parameters. It requires $O(I^2)$ operations for each iteration, where I is the number of data samples making it difficult to apply it to a high volume of data. The memory complexity relates to the third limitation when the co-association matrix is stored.

Clustering combination is a recent interesting topic in data mining and it appears up to now weakly exploited. A recent survey, [22], only reviews the few methods of clustering combination here presented, and even references in [3–7] were omitted.

As we have seen, many methods need to know a priori information about data in order to combine clusterings or to manually fix some parameters for the combination scheme. This motivates us to put the problem in a form free from any parameter and prior knowledge. The adopted formulation is also based on the co-association matrix. It allows to process a large volume of data as well as large numbers of final clusters without using the co-association matrix explicitly. We estimate clustering combination via an objective function proposed in [4–6] and introduce two algorithms to optimise this criterion. The first algorithm uses a hierarchical approach and shows competitive performances compared to existing ones. It combines clusterings in an unsupervised way for a large volume of data. Unfortunately, there is no proof that it always achieves a global optimum. The second algorithm is a fast iterative combination algorithm for which we prove the convergence to a global optimum of the objective function.

3 Problem statement

Let us consider the case where we have a large set of samples and different clusterings, each of them providing a partition of the sample set into a specific number of clusters. Let I be the number of samples and P the number of clusterings. Each clustering associates each sample u with one and only one cluster. The elementary co-association matrix A^p of clustering p $(p = 1, ..., P)$ collects the information on which sample u belongs to the same cluster that sample v:

$$A^p_{uv} = \begin{cases} 1, & \text{if } u \text{ and } v \text{ are in the same cluster,} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

where $u, v = 1, ..., I$. A^p is a binary symmetric square matrix of size I. We may similarly describe the p^{th} clustering by binary matrix B^p with I rows and J^p columns, where J^p is the number of clusters in the p^{th} clustering, so that:

$$B^p_{uj} = \begin{cases} 1, & \text{if sample } u \in j, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

where $j = 1, .., J^p$. B^p is called a partition matrix. We have:

$$A^p = B^p B^{p\prime}, \tag{3}$$

where $'$ denotes the matrix transposition.

For the P clusterings, we can compute the average matrix A as:

$$A = \frac{1}{P} \sum_{p=1}^{P} A^p = \frac{1}{P} \sum_{p=1}^{P} B^p B^{p\prime}. \tag{4}$$

A is the global co-association matrix or, in short, the co-association matrix. Each clustering presented by binary matrix A^p has weight $1/P$ in Eq. (4). For large P, we may say that two elements u and v have a probability A_{uv} to belong to the same cluster.

Let us denote the consensus partition having J unknown clusters as matrix B of size $I \times J$. This partition of the samples reflects at best the point of view of all clusterings. Our goal is to obtain such a consensus partition B from the co-association matrix A. From B, we may compute a square matrix D of size I as:

$$D = BB'. \tag{5}$$

Such a matrix D would be the binary co-association matrix corresponding to the consensus clustering. For any problem, where P different clusterings are performed, we may observe one matrix A, but the consensus partition B is unknown as well as D. The purpose of the clustering combination is to derive these unknown matrices.

Several matrices D could be obtained, depending on the criterion chosen to derive D from A. For instance, in [8] the matrix D is obtained from A by maximising Normalised Mutual Information (NMI) criterion based on information theory. We follow here the formulation of the problem given in [4] as the one which minimises the square error between D and A:

$$E(D) = \|D - A\|^2, \tag{6}$$

which can be rewritten (since D Eq. (5) is a binary symmetric matrix) as:

$$\begin{aligned} E(B) &= \sum_{u=1}^{I}\sum_{v=1}^{I}\left(\sum_{j=1}^{J}(B_{uj}B_{vj}) - A_{uv}\right)^2 \\ &= \sum_{u=1}^{I}\sum_{v=1}^{I} D_{uv}(1-2A_{uv}) + \sum_{u=1}^{I}\sum_{v=1}^{I} A_{uv}^2, \qquad (7) \\ &\text{subject to} \quad B'B = \mathbf{I}, \ \sum_{j=1}^{J}\mathbf{I}_{jj} = I, \ B_{uv} \in \{0,1\}, \end{aligned}$$

where $\mathbf{I}$ is a diagonal matrix of size J with diagonal elements equal to the clusters' cardinalities. Unknown consensus partition B has size $I \times J$, where the number of concensus clusters J has to be estimated. The first term in the second part of Eq.(7) has no square degree because D is a binary matrix. In addition, the last term in the second part of Eq.(7) is a constant and does not influence on the minimisation of error E. The quadratic objective function Eq. (7) may be solved exactly for small data sets using efficient methods [5], in contrast to the optimisation of *NMI* criterion in [8,9].

4 Proposed solution

4.1 Combination algorithm

In order to combine clusterings and find B that minimises E Eq. (7) we propose to use a single-link merging algorithm [1]. This algorithm optimises Condorcet criterion [7] which equals the quadratic criterion Eq. (7) up to a constant. The single-link method gives experimentally very good results when compared to other hierarchical algorithms such as average-link, Ward, complete-link, etc., [8]. The motivation of using single-link algorithm is based on the previous remark that the general term A_{uv} of matrix A may be considered as the probability of two samples to belong to the same cluster. Of course we do not know the memberships of u and v and the actual number of clusters J, but it is reasonable to group in the same cluster elements of A that have the highest probability of co-association, that is the way single-link works [1]. We propose the Least Square Error Combination (*LSEC*) algorithm for solving Eq. (7) (see Algorithm 1). The optimal number of clusters J is found when the error E in Eq. (7) is minimum. At the first step we initialise B as the identity matrix supposing that each cluster has only one sample. Error $E^{(1)} = I^2$ is initialised to have its maximal value. A partition presented by matrix B is stored to matrix B^* before merging two clusters. Merging is continued till minimising error $E^{(i)}$.

Algorithm 1
Pseudo code of *LSEC*-algorithm

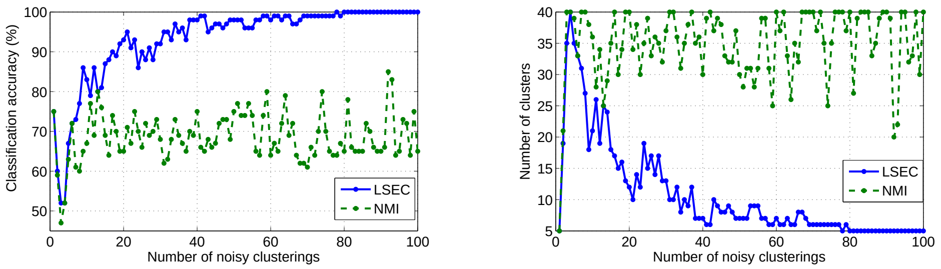
```
1:  Set $B$ as the identity matrix, $J \leftarrow I$, $i \leftarrow 1$ and $E^{(i)} \leftarrow I^2$.
2:  Find clusters' indexes $(j,k) = \arg \max_{u \in j, v \in k} A_{uv}$; $j,k = 1,...,J$, $j \neq k$.
3:  Set $B^* \leftarrow B$.
4:  Merge two clusters $j$ and $k$ by $B_{uj} \leftarrow (B_{uj} + B_{uk})$.
5:  Remove column $k$ from matrix $B$.
6:  $E^{(i+1)} \leftarrow \sum_{u=1}^{I} \sum_{v=1}^{I} \left( \sum_{j=1}^{J-1} (B_{uj} B_{vj}) - A_{uv} \right)^2$.
7:  if $E^{(i+1)} \leq E^{(i)}$, then
8:      $i \leftarrow i + 1$,
9:      $J \leftarrow J - 1$,
10:     go to Step 2;
11: else $B \leftarrow B^*$, $B$ is the optimal partition, stop.
```

Simulated example

In order to demonstrate the efficiency of this algorithm, it has been experimented on synthetic noisy data. The experiment was carried out on a data set of $I = 100$ samples with $J^p = 5$ classes each of which has 20 samples. We simulate a clustering by randomly changing 25% of the samples from one class to another. The noisy sets, so constructed are considered as the result of one clustering. We repeat this experience P times to simulate $p = 1, ..., P$ independent clusterings. Matrix B^p Eq. (2) is constructed for each of p noisy clusterings. From these clusterings, matrix A Eq. (4) is estimated from B^p and following *LSEC* algorithm we determine the consensus classification. For each class of the consensus classification, we compute the class accuracy (expressed in percentage) as the ratio of the number of the samples issued from the majority class to the total number of samples. The consensus accuracy is expressed as the mean accuracy over all the classes. Figure 1 shows comparison of *LSEC*-algorithm with *NMI* criterion for $single-link$ algorithm [8] for different values of P from 1 to 100. In the Figure 1a for $P = 1$ both curves start with a 75% accuracy since 25% noise was added to a single clustering. When P increases, *LSEC* curve has a chaotic behaviour first, then it converges towards 100% accuracy. On the contrary, *NMI* never benefits from the many noisy clusterings to improve the global accuracy.

For $P = 100$ noisy clusterings, *LSEC*-algorithm accuracy is 100%, contrary to *NMI* criterion which provides about 70% of accuracy. Figure 1b shows the accuracy versus the cluster number. We see that for a large number of noisy clusterings the accuracy of *LSEC*-algorithm in estimating the actual number of clusters is good whereas it fails for the *NMI* criterion.

Matrix A is computed in $I(I-1)/2$ iterations. In order to combine clusters, I iterations are needed and error E is calculated in $I(I-1)/2$ iterations for each



a

b

Fig. 1. Comparison of *LSEC* algorithms and *NMI* criterion (*single-link* algorithms). a - Combination accuracy versus the number of noisy clusterings, b - Estimated number of clusters versus the number of noisy clusterings

combination. The time complexity of such an algorithm being in $O(I^3)$, it is prohibitive for large volumes of data. To overcome this problem we propose an efficient initialisation procedure in Section 4.2 at first and a fast version of the algorithm for large data sets in Section 4.3.

4.2 Initialisation

Algorithm 1 starts with an initialisation of the matrix B as the identity matrix. A better initialisation of B can accelerate the convergence. A gradient like method which iteratively modifies B and minimises the error E Eq. (7) can reduce the computation time. Consider an iteration of optimisation which switch the label j_0 of sample q to j. Let B^{j_0} and B^j be the partition matrices before and after this allocation. The variation of E Eq. (7) is given by:

$$\Delta E(q|j_0 \to j) = \sum_{u=1}^{I}\sum_{v=1}^{I}(D^j_{uv} - D^{j_0}_{uv})(1 - 2A_{uv}), \tag{8}$$

where $D^j = B^j B^{j\prime}$, $D^{j_0} = B^{j_0} B^{j_0\prime}$ as in Eq. (5) and $j_0 \to j$ is an operation of label changing. The change is accepted if and only if $\Delta E(q|j_0 \to j)$ is not positive, and the process is iterated until no change minimises E. As the variation of the error E Eq. (8) depends only on the difference between D^j_{uv} and $D^{j_0}_{uv}$, ΔE could be written as:

$$\Delta E(q|j_0 \to j) = 2\sum_{k\in j, q\notin j}(1 - 2A_{qk}) - 2\sum_{l\in j_0, q\notin j_0}(1 - 2A_{ql}), \tag{9}$$

Let us explain equations (8) and (9). Binary co-association matrices D^j and D^{j_0} are square and symmetric. D^{j_0} has two binary square matrices on diagonal: elements of the first matrix are associated with clusters j_0 and elements of the second matrix are associated with cluster j. Let d^{j_0} and d^j be the first and the second diagonal matrices of D^{j_0}, respectively. When label j_0 of element q is switched to label j, then D^j equals D^{j_0} and is changed as: row q and column q of matrix d^{j_0} equal 0 (except for the diagonal element q, q), and row q and column q of matrix d^j equal 1. Thus, the difference between matrices D^j and D^{j_0} Eq. (8) equals Eq (9). Multiplier 2 comes from symmetry of matrices D^j and D^{j_0}.

At the initial step, when each cluster contains only one sample, matrix B is the identity matrix. Sample q is moved to the cluster which minimises the error Eq. (9). In this case cluster j_0 has only one sample q and l is an empty set. Then the error in Eq. (9) has the following form:

$$\Delta E(q|j_0 \to j) = 2(1 - 2A_{qk}). \tag{10}$$

Minimising ΔE is equivalent to finding the maximum of A_{qk}, excepting the diagonal elements of A. Using the nonpositiveness condition of the error variation in Eq. (10), the necessary condition to examine points A_{qk} is:

$$A_{qk} \geq 0.5. \tag{11}$$

Condition (11) means that two points could be combined if they are in the same cluster in more than half of the cases. This optimisation procedure is equivalent to building nearest-neighbour subgraphs. It avoids the storage of the square matrix A. It is very important when processing a large amount of data. Points belonging to each subgraph are assigned to the same cluster. Now, such clusters will be considered as the initialisation matrix B for *LSEC*-algorithm, instead of the new identity matrix, so we obtain a noticeable gain of processing time.

4.3 Gradient descent optimisation and storage reduction

In the proposed Algorithm 1, matrix A should be computed at **Step 1**. This step may be difficult for real applications such as clustering of large databases, because of the dimension of matrix A. For instance, when processing images, we often have to deal with thousands of pixels. This would involve millions of terms for matrix A. Instead of calculating the error at each step of the optimisation procedure, we suggest to use the optimisation error gradient as proposed in Eq. (8), and follow a descending approach as an optimisation strategy. The error gradient reduces both the computation time and the volume of storage and processing.

Let k and l be indexes of samples belonging to two clusters j_0 and j, respectively, with n_{j_0} and n_j samples each. Let D^{j_0} be the binary co-association matrix before combination and D^j after combination. All elements of D^j are either equal to 1 or to 0. Let E^{j_0} and E^j be errors as in Eq. (7) before and after combination. We obtain the difference ΔE between errors E^j and E^{j_0} by substituting matrices D^{j_0} and D^j in Eq. (8):

$$\Delta E = 2n_{j_0}n_j - 4\sum_{k}^{I}\sum_{l}^{I} A_{kl}. \tag{12}$$

A new condition for subcluster combination is obtained when the gradient error is non positive, *i.e.*:

$$\frac{\sum_{k}^{I}\sum_{l}^{I} A_{kl}}{n_{j_0}n_j} \geq 0.5. \tag{13}$$

Property (13) states that two subclusters j_0 and j are combined if the sum of their connection probabilities is greater than half of all possible connections of their points. We say that the normalised sum of their connections is greater than 0.5. The last term in the gradient ΔE in Eq. (12) allows us to calculate a double sum without storage of whole matrix A.

4.4 A complete iterative algorithm

Now let use the results presented in Section 4.2 which provide a good initialisation of the algorithm by an initial clustering based on nearest neighbour graphs. Let J^g be the number of these initial clusters. From J^g, a binary matrix B^g is built according to Eq. (2) and matrix $\mathbf{B} = [B^1, ..., B^p]$ is a concatenation of B^p. A is derived by Eq. (4) as:

$$A = \frac{1}{P}\mathbf{B}\mathbf{B}'. \tag{14}$$

Matrix S of size J^gxJ^g can be computed as the sum of connections between all pairs of J^g clusters:

$$S = B^{g\prime} A B^g = \left(\frac{B^{g\prime}\mathbf{B}}{\sqrt{P}}\right)\left(\frac{B^{g\prime}\mathbf{B}}{\sqrt{P}}\right)'. \tag{15}$$

Let each element N_{kl} of a matrix N correspond to the number of all possible connections of two clusters k and l:

$$N_{kl} = n_k n_l, \tag{16}$$

where $k, l = 1, ..., J^g$ and n_k,n_l are the numbers of samples in clusters k and l, respectively. The normalised sum of connections between two clusters k and l allows building matrix $\overline{S}$ where each element $\overline{S}_{kl}$ is expressed as:

$$\overline{S}_{kl} = S_{kl}/N_{kl}, \tag{17}$$

with $0 \leq \overline{S_{kl}} \leq 1$. From matrix $\overline{S}$ we may propose a generalisation of condition (13): if $\overline{S}_{kl} \geq 0.5$, clusters k and l should be combined to reduce the error E in Eq. (7) for *LSEC*-algorithm. Ranking $\overline{S}_{kl}$ elements in a descending order determines clusters that should be grouped at **Step 2**. This algorithm, called *DLSEC* (differential *LSEC*) significantly reduces computations and may be applied to large volumes of data.

We compare in Figure 2 processing time for a direct search presented in Section 4.1 with the optimised search proposed in Sections 4.2-4.4. An ideal clustering with $J^p = 6$ clusters is taken as an example. Random changes of labels are performed on 20% of the samples. By repeating this procedure 100 times matrix B^c is built. We can see from Figure 2, the proposed optimised search decreases significantly the processing time. Moreover, after combination of noisy clusterings, a perfect solution with 6 classes is always obtained. The bootstrapping method [22] is one of the possible applications of the *DLSEC*-algorithm. For this experiment, we set randomly 60% of samples with initial clustering labels and 40% as unclassified labels to which the same label "unclassified" is attributed. After 100 steps of boosting the combination finds exactly the initial clustering. It could be one of the issues for a parallel clustering of large amounts of data or for improving clustering.

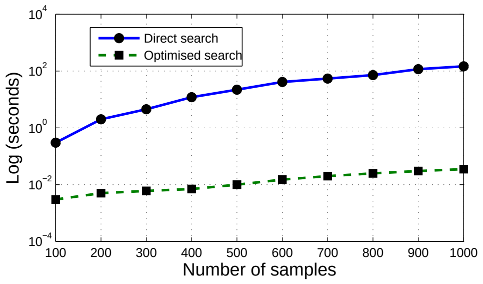


Fig. 2. Logarithm of time computation for direct and optimised search vs. the number of samples, in the case of synthetic data.

4.5 Complexity

To compute J^g clusters initialising *DLSEC* algorithm as described in Section 4.2, $I(I-1)/2$ operations at most are needed. The combination of these clusters as presented in Section 4.3 requires $J^g - 1$ operations, where $J^g \ll I$. The time complexity of optimised *DLSEC*-algorithm is approximately $O(I^2 + J^g)$. Note, that the method in [25] requires $O(I^2)$ operations at each step of the optimisation process.

We demonstrated the objective function and introduced the hierarchical algorithm to find the optimal concensus clustering. Unfortunately there is no clear proof that the hierarchical algorithm may achieve a global optimum of the objective function. To overcome this limitation we reformulate the optimisation process as well as the optimality conditions and propose an exact algorithm to find the global optimum for E Eq. (7).

5 Mean shift combination

In this section P clusterings are considered as labels coded by p binary matrices B^p Eq. (2), where $p = 1, ..., P$. The matrices are concatenated into a single matrix $\mathbf{B}$ and form space $\Re^d$, where $d = \sum_{p=1}^{P} J^p$. We propose to search a consensus clustering which, as previously, minimises the square error E Eq. (7). We prove in this section that this minimisation is equivalent to the minimisation of the square error among samples b_u, where b_u is a row of $\mathbf{B}$ and $u = 1, .., I$.

All samples $\{b_u\}$ are located on a hyper circle, since they simultaneously satisfy a hyper plane equation $\sum_{j=1}^{d} b_{uj} = d = const$ and a hyper sphere equation $\sum_{j=1}^{d} b_{uj}^2 = d = const$. Therefore vectors $\{b_u\}$ may be normalised by a constant $\sqrt{d}$ such that their square norm is 1.

Let us write the minimisation of square error E Eq. (7) as:

$$\begin{aligned} \min_B E = \min_B \sum_{u=1}^{I} \sum_{v=1}^{I} D_{uv}(1 - 2A_{uv}) = \min_{\{J, \{C_j\}_{j=1}^{J}\}} \sum_{j=1}^{J} \sum_{u \in C_j} \sum_{v \in C_j} (1 - 2A_{uv}) \\ = \min_{\{J, \{C_j\}_{j=1}^{J}\}} \sum_{j=1}^{J} n_j^2 (1 - \frac{2}{n_j^2} \sum_{u \in C_j} \sum_{v \in C_j} A_{uv}) \end{aligned} \tag{18}$$

where consensus cluster C_j, $j = 1, ..., J$ has the unknown number of samples n_j and J is the unknown number of consensus clusters. Set $\{C_j\}$ corresponds to binary matrix B of size $I \times J$.

As all elements verify $0 \leq A_{kl} \leq 1$ and $A_{uv} = b_u b_v'$ we may derive a condition to guarantee that the error Eq. (18) is always minimised:

$$\|b_u - b_v\|^2 < 1 \Rightarrow \frac{1}{n_j^2} \sum_{u=1}^{n_j} \sum_{v=1}^{n_j} A_{uv} > 0.5, \tag{19}$$

where $u, v \in C_j$. This condition shows the expression in the parenthesis of the last part of Eq. (18) is always negative. We may also say that if during the estimation of consensus clusters $\{C_j\}$ the condition (19) is hold and the number of samples n_j is growing then error E Eq. (18) is always minimised.

5.1 Proving convergence with mean shift

Let μ_j be the mean vector of cluster C_j, $\mu_j = \sum_v b_v/n_j$, $v \in C_j$. The square norm of μ_j is:

$$\|\mu_j\|^2 = \frac{1}{n_j^2} \left\| \sum_v b_v \right\|^2 = \frac{1}{n_j^2} \sum_v (\|b_v\|^2 + 2 \sum_u b_v b_u') = \sum_{u \in C_j} \sum_{v \in C_j} A_{uv}/n_j^2. \tag{20}$$

The square error σ_j^2 of cluster C_j with mean μ_j is:

$$\sigma_j^2 = \frac{1}{n_j} \sum_{u=1}^{n_j} \|b_u\|^2 - \left\| \frac{1}{n_j} \sum_{u=1}^{n_j} b_u \right\|^2 = 1 - \|\mu_j\|^2. \tag{21}$$

where $\|b_u\|^2 = 1$. Minimising the last term in Eq. (18) is equal to maximising both $\|\mu_j\|^2$ and the number of samples n_j in cluster C_j.

Proposition 1 *A global minimum of the error E in Eq. (18) is achieved by maximising the norms of local mean vectors μ_j in Eq. (20) or/and minimising square distances σ_j^2 in Eq. (21) jointly with maximising the number of samples n_j in clusters:*

$$\begin{aligned} &\min E = \min \sum_j n_j^2 (1 - 2\|\mu_j\|^2) = \min \sum_j n_j^2 (2\sigma_j^2 - 1), \\ &\textit{under conditions} \parallel \mu_j \parallel^2 > 0.5, \sigma_j^2 < 0.5. \end{aligned} \tag{22}$$

Written in this way, **Proposition 1** may be seen as a problem of parameter estimation. The problem can be solved via the estimation of a probability density function. The base of such an approach in regard to the pattern recognition is the nonparametric density estimation by its gradient [26,27], so-called the density estimation by mean shift vectors.

The multivariate kernel density estimation with kernel $K(b)$ and window radius h, computed at point b takes form [26]:

$$\hat{f}(b) = (Ih^d)^{-1} \sum_{u=1}^{I} K(h^{-1}(b - b_u)) \tag{23}$$

An appropriate kernel K should be selected to approximate the density. If the kernel has unknown parameters they should also be estimated. One of the popular kernels is the Gaussian kernel with the width of the kernel window [31] as parameter. This kernel is not appropriate for the problem at hand because it makes the assumption that the more data are available the denser the distribution is. In the case of our normalised samples $\{b\}$, a higher number of samples does not guarantee a higher density. We aim to group samples $\{b\}$ which are located on the different positive axes.

We propose to use the multivariate Epanechnikov kernel [28] in order to minimise of the average global error between the estimated and the true density [29]. The profile of the kernel is the function $k : [0, \infty) \rightarrow R$ such that $K(b) = k(\| b \|)$:

$$k(b) = \begin{cases} \frac{(d+2)}{2c_d}(1 - b), \text{if } b \leq 1, \\ 0, \text{ otherwise.} \end{cases} \tag{24}$$

where c_d is the volume of the unit d-dimensional sphere of radius 1.

The density estimation Eq. (23) is obtained through its gradient as shown in [27]:

$$\hat{\nabla} f_{h,K}(b) = \frac{2c_{k,d}}{Ih^{d+2}} \left[\sum_{i=1}^{I} k\left(\left\| \frac{b - b_i}{h} \right\|^2 \right) \right] \left[\frac{\sum_{i=1}^{I} b_i k\left(\left\| \frac{b-b_i}{h} \right\|^2 \right)}{\sum_{i=1}^{I} k\left(\left\| \frac{b-b_i}{h} \right\|^2 \right)} - b \right]. \tag{25}$$

The second term in Eq. (25) is the mean shift:

$$\mathbf{m}_{h,k}(b) = \frac{\sum_{i=1}^{I} b_i k\left(\left\| \frac{b-b_i}{h} \right\|^2 \right)}{\sum_{i=1}^{I} k\left(\left\| \frac{b-b_i}{h} \right\|^2 \right)} - b, \tag{26}$$

which expresses the difference between point b and the mean of the samples weighted by kernel k. It also shows the direction in which the density is increasing and where the weighted mean value should be replaced. The *mean* *shift* estimation always converges [27]. It proceeds in two steps: (i) compute the mean shift vector $\mathbf{m}_{h,k}$; (ii) move kernel $k(b)$ by $\mathbf{m}_{h,k}$.

Let us note two very important properties of mean shift algorithm applied to data $\{b\}$.
Property 1. *All $\{b\}$ vectors have positive values, consequently the cosine between successive mean shift vectors always remains positive [27], guaranteeing a fast and good convergence rate and no chaotic descent.*
Property 2. *As the mean shift algorithm converges [27] and all data $\{b\}$ have values from a finite set, the mean shift estimation of μ_j is obtained in a finite number of iterations. In practice, the iteration number for convergence is very small (some units).*

Condition (19) to achieve a global minimum of error E in Eq. (22) shows that the maximal distance among samples $\{b_u\}$ is less than 1. From this condition, the distance from mean vector μ_j to any point of cluster C_j is less than 1. The Epanichnekov kernel is differentiable in a sphere of radius 1; therefore optimisation converges to a global optimum [29]. We demonstrate a theorem which asserts the global optimality of Epanechnikov kernel to minimise E in Eq. (22).

Theorem 1 *Epanechnikov kernel is the optimal kernel to find a global minimum for error E in Eq. (22) by the mean shift algorithm.*

A proof of the theorem is given in the Appendix.

5.2 Optimal adaptive radius for mean shift combination

We proved in Section 5.1 that the mean shift combination with the Epanechnikov kernel finds an optimal solution for error E in Eq. (18). Because the starting point is a data sample $\mu_j = b_i$ the threshold is set to 1, so (19) satisfies the condition of the Epanechnikov kernel with a radius 1. As $\|\mu_j\|^2$ is changed during the search, an optimal radius should be estimated. Condition (19) shows that the optimal solution of the error (18) is found when $A_{uv} > 0.5$. In such a case using the square norm of mean vector μ_j in Eq. (20) calculated on n_j samples and in the worst case when $A_{uv} = b_u b'_v = 0.5 : u \neq v$; $A_{uu} = 1$, then $\|\mu_j\|^2 = (0.5n_j(n_j-1)+n_j)/n_j^2 = 0.5(1+1/n_j)$. To optimise the error E in Eq. (18) the optimal adaptive radius r_j (or similarly the minimal distance from any sample $b_u : u \notin j$ to the mean vector μ_j) should be:

$$r_j = \sqrt{\|b_u - \mu_j\|^2} = \sqrt{1 - 2b_u \sum_{v \in j} b_v/n_j + \|\mu_j\|^2} = \\ \sqrt{\|\mu_j\|^2} = \sqrt{0.5(1+1/n_j)} \tag{27}$$

From this equation it may be verified that :

$$\begin{aligned}&- \text{ when, } \mu_j = b_v \text{ then } r_j = 1 \text{ therefore satisfying (19),}\\ &- \lim_{n_j \to \infty} r_j = \lim_{n_j \to \infty} \sqrt{\|\mu_j\|^2} = \lim_{n_j \to \infty} \sqrt{0.5(1 + 1/n_j)} = \sqrt{0.5} \approx 0.7071.\end{aligned} \tag{28}$$

From this limit we obtain a low bound for the square norm of the mean vector $\mu_j : 0.5 < \|\mu_j\|^2$. This value always guarantees the minimisation of error E in Eq. (18). We may now present the algorithm of the Mean Shift Combination (MSC) with Epanechnikov kernel and adaptive radius r_j (see Algorithm 2).

Algorithm 2
Pseudo code of MSC-algorithm

```
   **Initialise** $j = 1, c_i = 0, i = 1, ..., I$
1: Find $i : c_i \equiv 0$, else stop, $c$ has labels of combination.
2: Initialise $r_j = 1, k = 1,\ y_k = b_i$.
3: Compute $y_{k+1} = \frac{1}{n_k} \sum_{b_i \in W(y_k, r)} b_i$,
4:             $r_k = \sqrt{0.5(1 + 1/n_k)}$,
5:             $k \leftarrow k + 1$ till convergence.
6: Assign $r_j = r_k, c_i = j, \forall i : \sqrt{\|b_i - y_{conv}\|^2} < r_j,\ j = j + 1$. Go to **Step 1**.
```

Where n_k is the number of points in the window $W(y_k, r_k)$ of radius r_k with centre y_k. After converging, points falling into window $W(y_{conv}, r_j)$ belong to cluster C_j. Vector c has nonzero labels of J clusters. All local optimal modes form groups of samples and consequently constitute a global solution of combination.

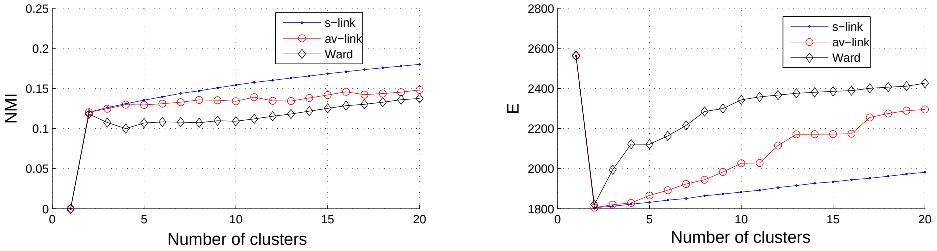
6 Experimental results

6.1 *Synthetic clustering combination*

In this subsection we present different clustering combination criteria and algorithms on synthetic data. To generate simulations we take one clustering and exchange randomly samples from true clusters to false ones. From these clusterings, several classes are extracted by different methods: hierarchical *single-link*, *Ward* and *average-link* algorithms [1] for the average normalised mutual information *NMI* criterion [8], $LSEC$-algorithm and MSC-algorithm as presented in this paper, $AUTOCLASS$ clustering [30] (that cluster labels by mixtures of multinomial models with Expectation-Maximisation algorithm).

The first experiment is made for 2 classes each of them containing 50 samples.

30% randomly selected samples are changed to the other class. Each labelling is represented as binary matrix B (see Eq. (2)). We collect 100 of such noisy labelings and construct co-association matrix A Eq. (1). Figure 3 shows two criteria to determine the optimal number of clusters: information *NMI* and square error E Eq. (7). The optimal cluster number is obtained for the maximum *NMI*, or for the minimum square error E. For such an elementary example *NMI* criterion is increasing with a growing number of clusters for all hierarchical algorithms. On the contrary error E in Eq. (7) provides always the true number of clusters whatever the fusion algorithm. *AUTOCLASS* gives the true solution, but should be initialised with the a prior information on the number of clusters and needs a large number of restarting (about 100). It is well known that clustering methods based on *EM*-algorithm do not guarantee a global optimum [2,16,20]. The best solution is selected among many trials using restarting (*e.g.,* with random parameter initialisation). The proposed *LSEC*-algorithm as well as the *MSC*-algorithm give 2 clusters without errors. This experiment was widely extended to many other synthetic cases with the same conclusion: the error E Eq. (7) indicates more precisely the true number of clusters than *NMI*.



a

b

Fig. 3. Combination of clusterings with 30% of noised labels by single-link, average-link and Ward hierarchical algorithms: a - *NMI* criterion, b- square error E in Eq. (7).

6.2 UCI data

We perform experiments of "clustering combination" on real datasets taken from the UCI machine learning repository [1] and compare results with the work of [8] where the normalised *NMI* criteria is studied. The goal of these experiments is to show that the proposed combination algorithms are competitive and may even outperform averaged *NMI* criterion in [8].

Real data from UCI repository are the same as in [8]: 1. Iris data (150 samples);

[1] http://kdd.ics.uci.edu/

2. Breast Cancer (683 samples); 3. Optical Digits (3823 samples); 4. Log yeast (384 samples); 5. Std Yeast (384 samples).

To obtain clusterings of data we use *K-means* algorithm for fixed and random number of clusters. The fixed number of clusters k^* is the "natural" known number of classes and the random number is chosen randomly near k^*. Estimating the optimal number of clusters is a classical problem for K-means which is rather successfully solved [20]. We do not address this problem in our paper.

Error E (Eq.(7) and Eq.(18)) estimates the optimal number of clusters for hierarchical and mean shift combination algorithms, respectively. We note that combination depends on clusterings which depend on the number of clusters. The more clusters clusterings have, the more clusters in the combined clustering. After the combination we estimate its quality as the percentage of missclassified samples. The largest number of samples in a combined class was set as the true one and all other samples in this class are set as misclassified. The minimum value of this error is used to indicate the best clustering for 100 random initialisations of *K-means* algorithm.

LSEC-algorithm, *MSC*-algorithm and *AUTOCLASS* (AC) were used to combine different clusterings and their results are compared to the best Evidence Accumulation Clustering (EAC) with single-link or average-link approaches (EAC-SL, EAC-CL), Table 3 and Table 2 in [8] for fixed and random k^*, respectively. Here again for *AUTOCLASS* combination we should always provide a priori number of clusters and a large number of restartings to obtain a good solution. Results of combination of clusterings is presented in Table 1 as error rates of classification (in percentage).

Table 1
Error (in percentage) of the clustering combination

		Fixed k^*					Variable k^*			
Data set	k^*	KM	Jain[8]	AC	LSEC	MSC	Jain[8]	AC	LSEC	MSC
Iris	3	10.7	11.1	**10.7**	**10.7**	**10.7**	**10.0**	**10.0**	**10.0**	**10.0**
Brest Cancer	2	3.9	4.0	**3.9**	**3.9**	**3.9**	**2.9**	**2.9**	**2.9**	**2.9**
Optical Digits	10	13.1	23.2	17.3	17.1	**15.7**	21.0	11.8	11.1	**10.5**
Log Yeast	5	58.6	66.6	59.4	58.8	**58.8**	59.0	**49.2**	52.8	50.3
Std Yeast	5	26.1	31.8	**31.2**	32.8	32.5	33.0	26.5	26.8	**26.3**

The first set of experiments is done with a fixed number of clusters, and a random initialisation of K-means. From Table 1 (Fixed k^*) we see that *LSEC* and *MSC* algorithms have lower errors (columns LSEC and MSC) in most cases comparing to *NMI* criterion (column Jain[8]). *AUTOCLASS* (column AC) justifies good performance of *LSEC* and *MSC* algorithms with near the same error. The same values of error for Iris and Brest Cancer data are explained by the fact that these data have small size therefore clusterings and combinations are the same.

In addition we conclude that *MSC*-algorithm outperforms *LSEC*-algorithm as expected from the theory. We observe that in several cases *MSC*-algorithm has significantly lower errors than *NMI* [8] (less than 7.8% for Log Yeast and 7.5% for Optical Digits). The best *K-means* clustering errors are less than several combinations for fixed k^* because of the presence of many low quality clusterings which degrade the fusion. Note however that this criterion (best *K-means*) is only acceptable in the case where the exact number of clusters is known.

The second set of experiments is done with a varying number of clusters, and a random initialisation of K-means. Columns Jain[8], AC, LSEC and MSC of Table 1 (Variable k^*) show clustering errors after the combination by Jain, *AUTOCLASS*, *LSEC* and *MSC* algorithms, respectively. In such an experiment with the combination we find "stable" clusters instead of the natural clusters. Therefore the estimated numbers of clusters k' may differ from a priori known k^*. Here again, we see that the performances of the proposed combination algorithms (columns LSEC and MSC) are still very good and better than EAC-SL or EAC-AL (column 7) in [8, Table 2]. Interesting to note, that in [8] there is no definitive decision about which combination algorithm is the best.

Experiments on synthetic examples as well as on real data bases show better performance of our combination algorithms than in [8].

We compare clustering combination algorithms via different criteria: computational and memory complexities, the achievement of the global optimum, the necessity of multiple restarts and the fixed number of clusters. The algorithms and criteria are presented in Table 2.

Table 2
Criteria for comparing combination algorithms

	Computational complexity	Memory complexity	Global optimum	Multiple starts	Fixed number of clusters
NMI, Jain[8]	$O(I^2)$	$O(I^2)$	No	No	No
AUTOCLASS	$O(I)$	$O(I)$	No	Yes	Yes
LSEC	$O(I^3)$	$O(I^2)$	No	No	No
DLSEC	$O(I^2)$	$O(I)$	No	No	No
MSC	$O(I)$	$O(I)$	Yes	No	No

The computational complexity shows the number of iterations which are needed to obtain a solution as a function of the data set size. The memory complexity indicates the size of allocated memory as a function of data set size. These two complexities have to be linear when processing large data sets. Global optimum criteria shows whether an algorithm achieves the global optimal combination. Here are two notions of the global optimum : theoretical and practical. A combination algorithm can achieve the global optimum theoretically, but practically very often it is not the case and vice-versa. Multiple restarts are needed to select the best clustering combination. The fixed number of clusters shows whether a user should fix this number for an algorithm or to estimate it on the fixed set of numbers.

From Table 2 we see that *NMI* and *LSEC* approaches have square (or even more) computational and memory complexities that makes difficult their application for large data sets. *DLSEC* approach has square computational complexity and linear size of memory that may facilitate data processing. AUTOCLASS and MSC approaches have linear complexities which are preferable in practical tasks.

There is no proof for *NMI* criteria to get the global combination, however in practice, it may be possible for some cases. $AUTOCLASS$ with unsupervised clustering can infer theoretically the global combination, however in practice it is not the case, except simple and trivial combinations. *LSEC* and *DLSEC* algorithms combine clusterings obtaining the local optimum because it uses the gradient descend as the optimisation method. However, for some cases these two approaches can achieve global optimum.

NMI, *LSEC* and *DLSEC* approaches estimate the optimal number of clusters without multiple starts. On the contrary, $AUTOCLASS$ with the multinomial mixture model should be restarted to estimate the parameters of the model. The number of clusters should also be fixed during estimation. Finally, the best combination is selected from the set of solutions corresponding to different numbers of clusters.

As we demonstrated in this paper mean shift combination (*MSC*) converges to the global solution theoretically and practically under specified conditions.

From the comparison of different approaches for clustering combination in Table 2 we conclude that mean shift combination *MSC* is the best on them.

7 Conclusions

In this paper, we proposed two efficient algorithms for the combination of optimal clusterings based on the examination of the co-association matrix of the data as suggested in [3–5]. The combination algorithms have no parameters to tune, do not need multiple restarts and determine the number of combined clusters in an unsupervised way. We showed the objective function and conditions for its optimality. The first method uses single-link algorithm to find the optimal solution. Such an algorithm was chosen experimentally because of its goods results compared to other hierarchical algorithms. But it does not guarantee the convergence to the global optimum. To avoid this problem a new combination approach is proposed based on a mean shift procedure. It has been proven that mean shift minimises the square distance between clusterings, achieves the global optimum and has a linear complexity. Mean shift method is able to process large set of samples, without facing problems of memory or time complexity. In addition, it has the elegant formulation and the simple realisation.

The combination of clusterings is able to improve unsupervised data mining. To analyse data it is preferable to apply different clustering algorithms. Thus, we can combine clusterings issued from incomparable methods. The combination may be used for many different applications of data mining: clustering of nominal data (*e.g.,* text documents), combination of different clusterings or segmentations of the same scene (*e.g.,* by clustering different groups of features or clustering time-series images), video clustering and motion detection. It can stabilise a clustering result for an algorithm which depends on the choice of the set of initial parameters.

For a future work, it would be preferable to take into account the "weakness" or "strength" of clusterings and combine weighted clusterings as well as consider their dependency to improve clustering combination.

Acknowledgements

The authors are thankfull to reviewers for their very constructive comments, remarks and suggestions that helped us to improve our paper. This work was done with the financial support of Centre National d'Etudes Spatiales (CNES-France). The authors are grateful to Hichem Sahbi from TELECOM ParisTech, France, for his comments and to Jean-François Marcotorchino (Thales) for pointing out references [3–7].

Appendix. Proof of Theorem 1

Firstly, we show the maximisation of the mean shift vector norm. **Proposition 1** is a particular case of the theorem proposed in [31, pp. 282] that establishes the optimum solution is found when the mean shift procedure maximises the norm of the mean shift vector.

Secondly, we prove that during optimisation the number of points n_j falling into cluster j is a strictly monotonic increasing sequence. Let y_k be a point where density is estimated within the d-dimensional window $W(y_k)$. Let the density estimation $\hat{f}$ in Eq. (23) with Epanechnikov kernel in Eq. (24) for k and $k+1$ consecutive steps be $\hat{f}_k$ and $\hat{f}_{k+1}$ respectively:

$$\begin{aligned} &\hat{f}_k = \frac{1}{(Ih^d)} \sum_{b_u \in W(y_k)} K\left(\frac{b-b_u}{h}\right) = \\ &\frac{(d+2)}{2Ic_d} \sum_{b_u \in W(y_k)} (1 - \|y_k - b_u\|^2) = \frac{(d+2)}{2Ic_d} \frac{1}{n_k} \sum_{b_u, b_v \in W(y_k)} b_v b'_u \end{aligned} \tag{29}$$

and

$$\begin{aligned} &\hat{f}_{k+1} = \frac{(d+2)}{2Ic_d} \sum_{b_u \in W(y_{k+1})} (1 - \|y_{k+1} - b_u\|^2) = \\ &\frac{(d+2)}{2Ic_d} \frac{1}{n_{k+1}} \sum_{b_u, b_v \in W(y_{k+1})} b_v b'_u. \end{aligned} \tag{30}$$

The theorem in [32, pp. 1198] shows that the positive sequence $\{\hat{f}_k\}$ of density estimation by mean-shift algorithm and Epanechnikov kernel is converging and

$$\hat{f}_{k+1} - \hat{f}_k \geq \frac{d+2}{2Ic_d} n_k \|y_{k+1}\|^2, \tag{31}$$

consequently the condition $\hat{f}_{k+1} > \hat{f}_k$ holds. Using this condition we may prove that $n_{k+1} > n_k$. Let us rewrite inequality (31) by substituting equations (20), (29) and (30):

$$\begin{aligned} &\frac{(d+2)}{2Ic_d} \left(\frac{1}{n_{k+1}} \sum_{b_u, b_v \in W(y_{k+1})} b_v b'_u - \frac{1}{n_k} \sum_{b_u, b_v \in W(y_k)} b_v b'_u \right) \geq \\ &\frac{(d+2)}{2Ic_d} \frac{n_k}{n_{k+1}^2} \sum_{b_u, b_v \in W(y_{k+1})} b_v b'_u. \end{aligned} \tag{32}$$

Dividing inequality (32) by $\hat{f}_{k+1}$ in Eq. (30) and using $0 < \hat{f}_k / \hat{f}_{k+1} < 1$ the inequality (32) is written as:

$$1 - \frac{\hat{f}_k}{\hat{f}_{k+1}} \geq \frac{n_k}{n_{k+1}} > 0 \Rightarrow 0 < 1 - \frac{n_k}{n_{k+1}} < 1 \Rightarrow 0 < n_k < n_{k+1}. \tag{33}$$

When the optimal value is achieved, then $\hat{f}_{k+1} \equiv \hat{f}_k$ and $n_j = n_{k+1} \equiv n_k$. We proved here that the number of samples n_j^2 is strictly increasing. The condition

$\|\mu_j\|^2 > 0.5$ in Eq. (19) provides strictly negative values during minimising error E in Eq. (22) by the mean-shift algorithm with Epanechnikov kernel in Eq. (24).

References

[1] A. Jain, R. C. Dubes, 1988, Algorithms for Clustering Data. Prentice-Hall, Englewood Cliffs, NJ.

[2] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, John Wiley and Sons, 2001.

[3] E. Diday, Optimisation en classification automatique. Tome 1, 2. (French) [Optimization in automatic classification. Vol. 1, 2], Institut National de Recherche en Informatique et en Automatique (INRIA), 1979.

[4] P. Michaud, F. Marcotorchino, Modèles d'optimisation en analyse des donnés relationnelles. Mathématiques et Sciences Humaines, 67 (1979), p. 7-38.

[5] F. Marcotorchino, P. Michaud, Agrégation de similarités en classification automatique. Revue de Statistique Appliquée, 30 no. 2 (1982), p. 21-44.

[6] F. Marcotorchino, N. El Ayoubi, Paradigme logique des écritures relationnelles de quelques critères fondamentaux d'association. Revue de Statistique Appliquée, 39 no. 2 (1991), p. 25-46

[7] H. Benhadda, F. Marcotorchino, Introduction á la similarité régularisée en analyse relationnelle. Revue de Statistique Appliquée, 46 no. 1 (1998), p. 45-69.

[8] A.L.N. Fred, A.K. Jain, 2005, Combining Multiple Clusterings Using Evidence Accumulation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(6), 835-850, IEEE Computer Society.

[9] A. Strehl, J. Ghosh, 2003, Cluster ensembles - a knowledge reuse framework for combining multiple partitions. J. Mach. Learn. Res., 3, pp. 583-617, MIT Press.

[10] A.P. Topchy, A.K. Jain, W.F. Punch, 2004, A Mixture Model for Clustering Ensembles. Proceedings of the Fourth SIAM International Conference on Data Mining, USA.

[11] H. Ayad, M.S. Kamel, 2005, Cluster-Based Cumulative Ensembles. Multiple Classifier Systems, 6th International Workshop, MCS 2005, USA, pp. 236-245.

[12] C. Boulis, M. Ostendorf, 2004, Combining multiple clustering systems. 8th European conference on Principles and Practice of Knowledge Discovery in Databases(PKDD), LNAI 3202, pp. 63-74.

[13] T. Li, M. Ogihara, S. Ma, 2004, On combining multiple clusterings. CIKM '04: Proceedings of the thirteenth ACM international conference on Information and knowledge management, ACM Press, pp. 294-303.

[14] Y. Qian, C. Suen, 2000, Clustering Combination Method. ICPR, 02, IEEE Computer Society.

[15] A. Topchy, B. Minaei-Bidgoli, A.K. Jain, W.F. Punch, 2004, Adaptive Clustering Ensembles. ICPR, 01, pp. 272-275, IEEE Computer Society.

[16] C.M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), Springer, 2006.

[17] Y. Freund, R. Schapire, Experiments with a New Boosting Algorithm, Proceedings of the Thirteenth International Conference on Machine Learning (ICML), pp. 148-156, 1996.

[18] M. Collins, R.E. Schapire, Y. Singer, Logistic Regression, AdaBoost and Bregman Distances. Machine Learning, vol. 48, pp. 253-285, 2002.

[19] A.K. Jain, M.H.C. Law, Data Clustering: A User's Dilemma, Pattern Recognition and Machine Intelligence, pp. 1-10, 2005.

[20] K. Fukunaga, Introduction to Statistical Pattern Recognition, Academic Press, 1990.

[21] C. Fraley, A.E. Raftery, How Many Clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis. The Computer Journal, August, 8, pp. 578-588, vol. 41 (8), 1998.

[22] L.I. Kuncheva, Combining Pattern Classifiers: Methods and Algorithms, 2004, Wiley-Interscience.

[23] S. Le Hegarat-Mascle, I. Bloch, D. Vidal-Madjar, Application of Dempster-Shafer evidence theory to unsupervised classification in multisource remote sensing, IEEE Transactions on Geoscience and Remote Sensing, vol. 35(4), pp. 1018-1031, 1997.

[24] I. O. Kyrgyzov, O. O. Kyrgyzov, H. Maître, M. Campedel, 2007, Kernel MDL to determine the number of clusters. In International Conference on Machine Learning and Data Mining MLDM 2007, Leipzig, Germany, pp. 203-217.

[25] T. Lange, J.M. Buhmann, 2005, Combining partitions by probabilistic label aggregation. KDD '05: Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, pp. 147-156.

[26] K. Fukunaga, and L.D. Hostetler, The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition, IT, 21, 1975, v. 1, January, pp. 32-40.

[27] D. Comaniciu, P. Meer, Mean Shift: A Robust Approach Toward Feature Space Analysis, IEEE PAMI, 24, 2002, 5, May, pp. 603-619.

[28] V. A. Epanechnikov, "Nonparametric estimation of a multivariate probability density", Theory Prob. Appl. (USSR), vol. 14, pp. 153-158, 1969

[29] D. Comaniciu, V. Ramesh, Real-Time Tracking of Non-Rigid Objects using Mean Shift, 2003, v. 2, pp. 142-149.

[30] P. Cheeseman, J. Stutz, Bayesian Classification (AUTOCLASS): Theory and Results, Advances in Knowledge Discovery and Data Mining, AAAI Press/MIT Press, 1996, U. M. Fayyad and G. Piatetsky-Shapiro and P Smyth and R. Uthurusamy, pp. 153-180.

[31] D. Comaniciu, An algorithm for data-driven bandwidth selection, IEEE PAMI, 25, 2003, v. 2 (25), February, pp. 281-288.

[32] D. Comaniciu, P. Meer, Mean Shift Analysis and Applications, ICCV '99: Proceedings of the International Conference on Computer Vision-Volume 2, 1999, pp. 1197-1203, IEEE Computer Society, Washington, DC, USA.

Dépôt légal : 2008 – 3ème trimestre
Imprimé à l'Ecole Nationale Supérieure des Télécommunications – Paris
ISSN 0751-1337 ENST C (Paris) (France 1983-9999)

TELECOM ParisTech

Institut TELECOM - membre de ParisTech

46, rue Barrault - 75634 Paris Cedex 13 - Tél. + 33 (0)1 45 81 77 77 - www.telecom-paristech.frfr

Département TSI