A Method of Clustering Combination Applied to Satellite Image Analysis

Ivan O. Kyrgyzov, Henri Maître and Marine Campedel
Competence Centre for Information Extraction and Image Understanding for Earth Observation,
GET/Télécom Paris - LTCI, UMR 5141, CNRS
46, rue Barrault, 75013, Paris, France
name@enst.fr

Abstract

An algorithm for combining results of different clusterings is presented in this paper, the objective of which is to find groups of patterns which are common to all clusterings. The idea of the proposed combination is to group those samples which are in the same cluster in most cases. We formulate the combination as the resolution of a linear set of equations with binary constraints. The advantage of such a formulation is to provide an objective function for the combination. To optimize the objective function we propose an original unsupervised algorithm. Furthermore, we propose an extension adapted in case of a huge volume of data. The combination of clusterings is performed on the results of different clustering algorithms applied to SPOT5 satellite images and shows the effectiveness of the proposed method.

1. Introduction

In recent years many different imaging sensors for Remote Sensing have appeared, delivering a huge amount of digital images. Experiments have shown that a small part of any satellite image may be captured in a vector of measures which expresses the main properties : the multi spectral or radiometric content, the textural properties, the structural properties, etc. Thus, this small image part is well represented as a point in a high dimensional space. We are interested in exploring this space in an unsupervised way. Different pattern recognition methods provide different interpretations of this space. For instance, partitional clustering methods look for the very dense parts of the space and aggregate the samples around them. Hierarchical approaches do not look for a partition but for a tree-like structure between clusters, which reflects different topological relationships [6]. We want to benefit from the diversity of these points of view to code different aspects of the space and to better reflect the complex information contained in the data.

Clearly, the different space representations are redundant but each of them can give some specific new information about data. Some approaches using several clusterings to make a final decision have been presented in [1, 2, 5, 8, 10, 12, 13, 14]. The main concern is how to combine them to obtain a final clustering solution. The contribution of this paper is to address this problem.

A review a some of the existing approaches is given in Section 2, the problem of combination is formulated in Section 3, then Section 4 describes the proposed algorithm and its improvements for real data application. Results on SPOT5 satellite images are presented in Section 5.

2. Related works

We give a short review of several combination methods and indicate some of their limits for our application. Some of the approaches for combining different clustering results were proposed in [5, 12]. Fred and Jain [5] use a co-association matrix to represent clusterings, a method that we will also adopt: each element of this square matrix has a mean value that corresponds to the frequency of co-occurrence of two elements in the same cluster. At the first step *K-means* is used as a clustering algorithm with a random initialization as well as a random number of clusters and it is run several times. A hierarchical clustering with a single-link method is then applied to the co-occurrence matrix. The final number of clusters is taken as the one that corresponds to the longest lifetime on the dendrogram of the hierarchical algorithm.

The method of Fred and Jain [5] is well adapted when the number of clusters is approximately known a priori. If the number of clusters is sequentially changed from 2 to the number of samples, the co-association matrix will change towards a near diagonal matrix with small values out of diagonal. So, the more clusters used to build the co-association matrix the more clusters result from the combination. The same kind of experiments but with another combination scheme is made by Topchy *et al.* in [14].

The problem of clustering combination is considered as a finite mixture model of clustering ensembles and it is solved according to the maximum likelihood criterion by the Expectation-Maximization algorithm.

Several combination methods need prior parameters. Strehl and Ghosh [12] use a mutual information-based objective function to combine clusterings. This procedure needs a predetermined number of clusters; its complexity is exponential. In [10] the authors set a priori parameters to combine clustering results. Ayad and Kamel [1] propose to combine clusterings generated by *K-means* algorithm with bootstrapping for different subsets of input data with the same number of clusters. The labels of clusterings are used to obtain a matrix of pairwise distances between clusters. A group-average hierarchical clustering algorithm is applied to group this matrix. An a priori fixed number of final clusters should be set. In [2], an agreement matrix is used to combine different clusterings. Assigning an element to a final cluster is reestimated to optimize an objective function subject to the constraint that clustered elements belong only to one final cluster. A fixed number of clusters is used for the final combination. Lange and Buhmann [7] optimize a probabilistic model of the co-association matrix. The EM-algorithm optimizes model parameters and needs $O(I^2)$ operations for each iteration, where I is the number of data samples. It makes difficult to apply this approach to a high volume of data. Many of the presented methods need to know a priori about data to combine clusterings or to set manually some parameters for a combination scheme. This provided us with the motivation to pose the problem of combination in a form which will not depend on any parameter and a priori knowledge. Our formulation of the problem is based on a co-association matrix. It allows us to process a huge volume of data as well as clusterings.

3. Problem statement

Let us consider the frequent case when we have no information on the final clustering that we want to derive from a set of initial clusterings. In order to determine the common clusters within each clustering, we examine which samples are associated. This is done by collecting the co-association matrix A^p. Let P be the number of initial clusterings. For each clustering $p = 1, ..., P$, A is a symmetric binary square matrix of size I (I equals the number of samples) where each single element A_{uv}, $(u, v = 1, .., I)$ is:

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

We may also describe the p^{th} clustering by a binary rectangular matrix B^p with I rows and J_p columns (J_p equals to the number of clusters in p^{th} clustering) so that:

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

where $u = 1, ..., I$, $j = 1, .., J_p$. B_p is called a partition matrix. We verify that:

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

where $'$ denotes the matrix transposition. For several clusterings J_p, we can compute the average matrix A as:

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

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 B^s a consensus clustering. Our goal is to obtain such a clustering B^s from the matrix A. We may compute the square $I \times I$ matrix D as

$$D = B^s B^{s\prime}. \tag{5}$$

Such a D would be a binary co-association matrix. In any clustering problems, with P different partitions, we may observe the matrix A, but the ideal binary partition matrix B^s is unknown as well as D.

We propose to formulate the problem of combination as looking for the best partition B^s from the knowledge of the co-association matrix A Eq. (4), which minimizes a square error:

$$E = \sum_{u=1}^{I}\sum_{v=1}^{I}\left(\sum_{r=1}^{I}(B^s_{ur}{B^s_{rv}}') - A_{uv}\right)^2 = \sum_{u=1}^{I}\sum_{v=1}^{I} D_{uv}(1 - 2A_{uv}) + \sum_{u=1}^{I}\sum_{v=1}^{I} A^2_{uv}, \tag{6}$$

$$\text{subject to } \; B^{s\prime}B^s = \mathbf{I}, \sum_{i}^{I} \mathbf{I}_{ii} = I, \; B^s_{uv} \in \{0, 1\},$$

where $\mathbf{I}$ is a diagonal $I \times I$ matrix with diagonal elements that correspond to cluster sizes. Proposed quadratic objective function Eq. (6) has a convex form for all possible consensus clusterings, contrary to a mutual information criterion as proposed in [5, 12].

4. Proposed solution

4.1 Combination algorithm

We consider the matrix A as the similarity matrix. To combine different clustering results and find B^s for the minimum of Eq. (6) we propose a single-link based agglomeration algorithm [6]. This algorithm has been experimentally shown as giving very good results when compared to other hierarchical algorithms like average-link, Ward, complete-link, etc., [5]. This proposition is based on the previous remark that the general term A_{uv} of the matrix A may be considered as the probability of 2 samples belonging to the same cluster. Of course we do not know to which cluster u and v belong and how many clusters exist, but it is reasonable to group in the same cluster elements of A that have the highest linkage probability. The single-link approach is also chosen because it forms clusters on a connectedness criterion [6]. The Least Square Error Combination algorithm (*LSEC*) that we propose to solve Eq. (6) may now be given.

***LSEC*-algorithm**

Step 1 Set B^s as the identity matrix, $i \leftarrow 1$ and compute the error $E^{(i)} \leftarrow \sum_{u=1}^{I}\sum_{v=1}^{I}\left(\sum_{r=1}^{I}(B^s_{ur}{B^s_{rv}}') - A_{uv}\right)^2$

Step 2 For all pairs (u, v) find indices of maximal probability of connectedness $(r, t) = max\{A_{uv} : u, v = 1, ..., I, u \neq v\}$

Step 3 If $A_{rt} = 0$, then B^s is the optimal partition, stop.

Step 4 Set $A_{rt} \leftarrow 0$, $B^h \leftarrow B^s$. Merge the r^{th} and t^{th} clusters by **4 a)** summing column k to r $B^h_{kr} \leftarrow \left(B^h_{kr} + B^h_{kt}\right)$ and **4 b)** setting to zero column k: $B^h_{kt} \leftarrow 0$, where $k = 1, ..., I$.

$$E^{(i+1)} \leftarrow \sum_{u=1}^{I}\sum_{v=1}^{I}\left(\sum_{r=1}^{I}(B^h_{ur}{B^h_{rv}}') - A_{uv}\right)^2$$

if $E^{(i+1)} \leq E^{(i)}$, then $i \leftarrow i + 1$, $B^s \leftarrow B^h$, $A \leftarrow A. * (1 - B^s B^{s\prime})$.
Go to **Step 2**

The expression $A \leftarrow A.*(1-B^s B^{s\prime})$ in **Step 4** is needed to delete connection between elements belonging to the same cluster and to keep the other connections unchanged, where '$.*$' is the pointwise product of matrices. The optimal number of clusters is found when the error E Eq. (6) has its minimum.

An experiment of a combination was carried out on synthetic data with $J_p = 30$ clusters of sizes from 1 to 30 samples. 20% of cluster labels were changed randomly with uniform noise. The matrix B^p Eq. (2) is constructed for each of $P = 40$ noisy clusterings. The matrix A Eq. (4) was estimated by B^p, $p = 1, ..., P$. After the combination of A by *LSEC*-algorithm all the clusters were found exactly. Matrix A is computed in $I(I-1)/2$ iterations. To combine clusters I iterations are needed where the error E is calculated in $I(I-1)/2$ iterations for each combination. The time complexity of such an algorithm is approximately:

$$O(I^2 + I^3). \tag{7}$$

4.2. Initialization

Our combination algorithm begins by a simple initialization of the matrix B^s as the identity matrix. A good initialization can accelerate the convergence of the algorithm. Let us consider a gradient like method which iteratively modifies B^s and minimizes the error E, Eq. (6). An optimization technique may be in a random selection of one sample q, and its allocation to a cluster j instead of its initial cluster j_0. Let B^{j_0} and B^j be partitions before and after the allocation of sample q respectively. The variation of the criterion E Eq. (6) is:

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

where $D^i = B^j B^{j\prime}$ and $D^{j_0} = B^{j_0} B^{j_0\prime}$ as in Eq. (5). 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 improves E. As the variation of the error E Eq. (8) depends only on the difference between D^j_{uv} and $D^{j_0}_{uv}$ it could be rewritten as:

$$\Delta E(q|j_0 \to j) = 2\sum_k(1-2A_{qk}) - 2\sum_l(1-2A_{ql}), \tag{9}$$

where index k runs for indexes of samples of cluster j without the index q, and l for indexes of samples of cluster j_0 without the index q. Let us look at the initial step when all samples are alone in their cluster: matrix B^s is the identity matrix. We move sample q to the cluster, which minimizes the error Eq. (9). In this case cluster j_0 has only one sample q and l is an empty set. Then the error Eq. (9) has the following form:

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

As each possible cluster j has only one sample, then j equals k. The minimization of the error Eq. (10) is equivalent to finding the maximum of A_{qk}, excepting diagonal elements of A. Using the nonpositiveness condition of the error variation Eq. (10), the necessary condition to examine points A_{qk} is:

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

The condition (11) means that two points could be combined if they are in the same cluster in more than half of the cases. This optimization procedure is equivalent to building nearest-neighbour subgraphs. It permits to avoid the storage of the square matrix A. It is very important when we process a huge amount of data. Points of each subgraph are assigned to the same cluster. Such clusters form the initialization matrix B^s for *LSEC*-algorithm which optimizes the criterion Eq. (6).

4.3. Improvements for real data

In the proposed algorithm we should compute matrix A at **Step 4**. It makes difficult to apply the algorithm for real applications such as images or large database clustering, because of the dimensional issue of matrix A. In image processing we want to cluster an image of size $n \times n$ on a pixel basis thus with n^2 samples, we have to build a matrix A of size $n^2 \times n^2$, i.e. with n^4 terms. It produces a huge volume of data for large n and can not be processed in a reasonable time for our experiments. However, we can find the solution to this problem in analyzing the error of combination Eq. (6). Instead of calculating the error at each step of the optimization procedure and comparing several errors, we may use the optimization error gradient, and follow a descending approach as an optimization strategy. The error gradient will very much reduce the computation time as well as the volume of stored and processed data.

We present the principle on the generic term of the iterative process. We want to find an error gradient ΔE after a combination of two clusters. Let k and l be indexes of samples of two clusters j_0 and j respectively where n_{j_0} and n_j are their numbers of samples. Let D^{j_0} be a binary co-association matrix with two clusters and D^j is a matrix after combination as in Eq. (5). In this case all elements of D^j equal 1. Let E^{j_0} be an error of two clusters and E^j be an error after their combination as in Eq. (6). We obtain the difference ΔE between errors E^j, E^{j_0} by substituting matrices D^{j_0} and D^j in the error Eq. (8):

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

We find a new condition for the subcluster combination from the property of the error gradient $\Delta E \leq 0$ Eq. (12) which simplifies calculations:

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

The interpretation of the property (13) is: two subclusters j_0 and j are combined if the sum of their connection probabilities is greater than a half of all possible connections of their points. We say that the normalized sum of their connections is greater than 0.5.

Now let see us the results presented in Section 4.2. The main advantage of the proposition is the good initialization of our algorithm by clusters of the nearest neighbour graphs. Let J^g be the number of these clusters. From J^g, we build a binary matrix B^g according to Eq. (2) and a matrix B^c as a concatenation of B^p. We derive A by Eq. (4) as:

$$A = \frac{1}{P}B^c B^{c\prime}. \tag{14}$$

We can find a matrix S of size J^gxJ^g as a sum of connections between all pairs of clusters J^g:

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

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 cluster k and l, respectively. Then the normalized sum of connections of two clusters k and l is the matrix $\overline{S}$ each element $\overline{S}_{kl}$ is expressed as:

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

where $0 \leq \overline{S} \leq 1$. The matrix $\overline{S}$ is a generalization of condition (13): if $\overline{S}_{kl} \geq 0.5$, two clusters k and l should be combined. The condition (13) indicates which two clusters should be combined to reduce the error E Eq. (6) for *LSEC*-algorithm. The element A_{rt} at **Step 2** indicates the order of clusters which should be grouped. It significantly reduces computations and allows it to be applied to large volumes of data. The bootstrapping is one of possible applications of the *LSEC*-algorithm. For the experiment, we take randomly 60% of samples with initial clustering labels and 40% as unclassified labels for which we set the same label. After 100 times of boosting the combination returns initial clustering. It could be one of the ways for a parallel clustering of huge amounts of data.

To compute J^g clusters of the nearest neighbour graph for the initialization of our algorithm as described in Section 4.2 we need $I(I-1)/2$ operations at maximum. The combination of these clusters as presented in Section 4.3 needs $J^g - 1$ operations, where $J^g \ll I$. The time complexity of optimized *LSEC*-algorithm is approximately:

$$O(I^2 + J^g). \tag{18}$$

Note, that our method only needs about $O(I^2)$ operations at maximum for the complete optimization comparing to the method in [7] which has near $O(I^2)$ operations at each step of optimization. Moreover, *LSEC*-algorithm can have a linear complexity if we take only nearest-neighbours ($e.g.$, in image processing applications).

5. Preliminary results

Remote sensed images of Earth Observation (EO) are used by the experts of various domains (ecology, agriculture, defence, etc.). Along with the image, the expert gets a description of an image in terms of sensor type, geographical coordinates, time of reception, spectral bands. This information gives a rough description of the image, but it characterizes the whole image and cannot give answers about the precise content of the image. However, such information which would describe the exact content in terms of imagery may facilitate image understanding, discover new information and improve the management of image databases. We are interested in applying different clustering algorithms to analyze satellite images. In this section we show only preliminary results of clustering combination. The goal of these experiments is to show that the combination of different points of view on the data derives common informative clusters. For the future we will use this combination as well as relationships between clusters to get a new information.

We do experiments on clusterings obtained on 6 different SPOT5 satellite images at a resolution of 5 meters per pixel. Each image has a size 1024x1024 pixels. They represent 6 cities: Copenhagen (Denmark), Istanbul (Turkey), Los Angeles (USA), La Paz (Mexico), Madrid (Spain), Paris (France). We assume that because of geography, culture and history each image has different surface textures. Samples of images are presented in Figure 1. We form a database

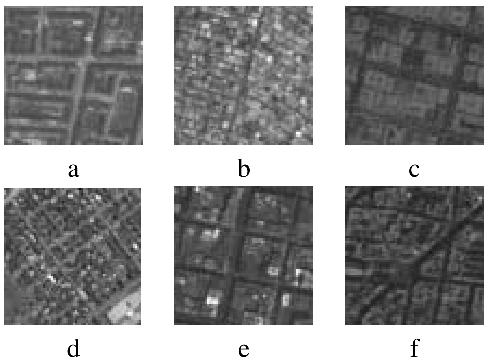

Figure 1. Samples of SPOT5 images: a - Copenhagen (Denmark), b - Istanbul (Turkey), c - Los Angeles (USA), d - La Paz (Mexique), e - Madrid (Spain), f - Paris (France). ©Copyright CNES

of samples by cutting each image in 400 samples, each of size 64x64 pixels. Samples overlap by 13 pixels in both directions. It produces a database composed of 2400 samples, 6 cities and 400 samples per city. From each sample several features have been extracted: statistics issued from Quadratic Mirror Filters filtering, statistics from Gabor filters and from Haralick co-occurrence matrix descriptors. 10 features were automatically selected from the 185 initial features using a Fisher selection [3]. Different unsupervised clustering algorithms were used to cluster a matrix data of size 2400x10: a classical *K-means* algorithm [6], Spectral *K-means* algorithm [9], Kernel *K-means* algorithm [11], Ward's hierarchical clustering algorithm [6] and *Expectation-Maximization* algorithm with a Gaussian mixture model implemented in AUTOCLASS [4]. To cluster data we set the fixed number of clusters to 6. We leave the determination of the optimal number of clusters for each algorithm out of the scope of this paper.

Clustering results are presented as confusion matrices in Tables 1 a-f. Each column of a table corresponds to a city in the same order that in Figure 1. Each line represents a cluster. As the number of clusters was set equal for every algorithm an estimation of the clustering's quality is the percentage of samples which were wrongly clustered. The largest number of samples in a cluster was set as the true clustered and all other samples in this cluster are set as misclassified. From the confusion matrices in Tables 1 a-f we can see that for certain classes different algorithms give different clustering solutions. All clusterings have redundant information but at the same time their intersections can generate new informative clusters. To analyze all intersections between clusters is a difficult task. The combination will allow us to derive a criterion to find stable clusters and stable representing samples of each cluster. *LSEC*-algorithm was used to combine 5 different clusterings. After initialization presented in Section 4.2 we obtain 63 subclusters. Proposed combination schema determines the optimal number of clusters equal to 6 in much less than 1 second. We see from Table 1 f that the combination generates appropriate common results with the performance as good as the best classification alone. Moreover, it provides clusters which reflect groups simultaneously proposed by different algorithms based on different criteria. The clustering error is better than such separate clusterings as in Table 1 d or in Table 1 e. Note, that the higher optimal number of clusters for each algorithm would decrease this error.

6. Conclusions

In this paper, we proposed an efficient optimization algorithm for the combination of optimal clusterings which avoids the use of any parameter, does not depend on initialisation, determines the number of clusters in an unsupervised way and significantly reduces redundant information. We showed the objective function and conditions for its optimization. Moreover, the method is able to work with very large set of samples, without facing problems of memory or time complexity.

The combination of different clusterings is able to improve unsupervised data mining which can produce new information about data. We did not consider the detection of the optimal number of clusters for each algorithm in this paper. To analyze data it is possible and even preferable to use different clustering algorithms. In such a way we can com-

Cities						
1	2	3	4	5	6	$\sum$
273	134	0	0	3	0	410
89	246	1	0	5	5	346
30	12	335	0	138	20	535
8	1	2	314	9	6	340
0	7	20	5	226	30	288
0	0	42	81	19	339	481
400	400	400	400	400	400	

a

Cities						
1	2	3	4	5	6	$\sum$
237	116	0	0	3	0	356
132	266	1	0	7	6	412
20	8	329	0	105	9	471
11	1	3	327	9	11	362
0	9	36	5	260	46	356
0	0	31	68	16	328	443
400	400	400	400	400	400	

b

Cities						
1	2	3	4	5	6	$\sum$
233	113	0	0	3	0	349
139	269	1	0	7	6	422
17	8	333	0	101	2	461
11	1	4	333	9	12	370
0	9	39	5	262	35	350
0	0	23	62	18	345	448
400	400	400	400	400	400	

c

Cities						
1	2	3	4	5	6	$\sum$
297	317	0	0	0	0	614
102	77	3	4	18	13	217
0	0	296	0	7	254	557
0	0	1	325	0	3	329
0	0	12	0	299	0	311
1	6	88	71	76	130	372
400	400	400	400	400	400	

d

Cities						
1	2	3	4	5	6	$\sum$
136	91	0	0	0	0	227
205	223	0	0	3	1	432
22	66	1	0	4	6	99
0	0	2	307	8	3	320
0	12	86	5	281	27	411
37	8	311	88	104	363	911
400	400	400	400	400	400	

e

Cities						
1	2	3	4	5	6	$\sum$
240	116	0	0	3	0	359
132	266	1	0	5	6	410
17	8	336	0	107	14	482
11	1	2	333	9	10	366
0	9	36	5	262	27	339
0	0	25	62	14	343	444
400	400	400	400	400	400	

f

Table 1. Confusion matrices and clustering errors for 6 classes. a - *K - means* **algorithm** 28%**, b -** *Spectral K - mean* **algorithm** 27%**, c -** *Kernel K - means* **algorithm** 26%**, d -** *EM - algorithm* 38%**, e -** *Ward's* **hierarchical clustering algorithm** 42%**, f - clustering combination by** *LSEC***-algorithm** 26%**. Note, that one city is often distributed in** 1 **or** 2 **classes only, reflecting a strong homogeneity in textures.**

pare and process different metrics which are not comparable in an original form. We note, that the number of clusters after the combination by *LSEC*-algorithm can differ from the number of clusters that is in each clustering.

This method can be used for many different applications of data mining tasks: 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. The combination can stabilize clustering result for an algorithm which depends on the choice of the set of initial parameters.

Acknowledgements: This study was done with the financial support of Centre National d'Etudes Spatiales (CNES-France).

References

[1] H. Ayad and M. S. Kamel. Cluster-based cumulative ensembles. In *Multiple Classifier Systems*, pages 236–245, 2005.

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

[3] M. Campedel, B. Luo, H. Maître, E. Moulines, M. Roux, and I. Kyrgyzov. Indexation des images satellitaires. détection et évaluation des caractéristiques de classification. Technical report, École Nationale Supérieure des Télécommunications, Département Traitement du Signal et des Images, 2004.

[4] P. Cheeseman and J. Stutz. Bayesian classification (AUTOCLASS): Theory and results. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, pages 153–180. AAAI Press/MIT Press, 1996.

[5] A. L. Fred and A. K. Jain. Combining multiple clusterings using evidence accumulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):835–850, 2005.

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

[7] T. Lange and J. M. Buhmann. Combining partitions by probabilistic label aggregation. In *KDD '05: Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 147–156, New York, NY, USA, 2005. ACM Press.

[8] T. Li, M. Ogihara, and S. Ma. On combining multiple clusterings. In *CIKM '04: Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 294–303, New York, NY, USA, 2004. ACM Press.

[9] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 849–856, Cambridge, MA, 2002. MIT Press.

[10] Y. Qian and C. Suen. Clustering combination method. *icpr*, 02:2732, 2000.

[11] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[12] A. Strehl and J. Ghosh. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.*, 3:583–617, 2003.

[13] A. Topchy, B. Minaei-Bidgoli, A. K. Jain, and W. F. Punch. Adaptive clustering ensembles. *icpr*, 01:272–275, 2004.

[14] A. P. Topchy, A. K. Jain, and W. F. Punch. A mixture model for clustering ensembles. In *SDM*, 2004.