Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Active constrained clustering with multiple cluster representatives

Shaohong Zhang, Hau-San Wong
Dept. of Computer Science
City University of Hong Kong
Hong Kong, China
shazhang@student.cityu.edu.hk

***Abstract*—Constrained clustering has recently become an active research topic. This type of clustering methods takes advantage of partial knowledge in the form of pairwise constraints, and acquires significant improvement beyond the traditional unsupervised clustering. However, most of the existing constrained clustering methods use constraints which are selected at random. Recently active constrained clustering algorithms utilizing active constraints have proved themselves to be more effective and efficient. In this paper, we propose an improved algorithm which introduces multiple representatives into constrained clustering to make further use of the active constraints. Experiments on several benchmark data sets and public image data sets demonstrate the advantages of our algorithm over the referenced competitors.**

***Index Terms*—Constrained clustering, active learning, image processing.**

I. Introduction

Clustering is one of the most important techniques to help people to understand the world with vast number of patterns (or data) but relatively scarce knowledge (or information). The objective of clustering is to group a set of data objects into a number of clusters such that intra-cluster distances are minimized and inter-cluster distances are maximized [1]. Clustering techniques are commonly used to discover knowledge or organize data in a lot of applications, which includes pattern recognition, machine learning, data mining, information retrieval, image segmentation and so on. Clustering techniques are also commonly used as a powerful tool to pre-process data in a number of applications [2], such as data indexing, data reduction, hypothesis generation and hypothesis testing. A great number of clustering algorithms are proposed and the topic is still under active development. According to [1], major clustering algorithms can be roughly classified into several categories including: (i) partitioning methods, which construct partitions of the data which correspond to clusters, such as Kmeans [3] or K-medoids clustering, including PAM [4], CLARA [4] and CLARANS [5]; (ii) hierarchical methods, which create a hierarchical decomposition of the given set of data points, such as linkage algorithms [6] [7] including single link, average link and complete link, ROCK [8], CURE [9], BIRCH [10] or CHAMELEON [11]; (iii) density-based methods, which continue growing the given clusters as long as the density (number of data points) in the neighborhood exceeds some threshold, such as DBSCAN [12], GDBSCAN [13], OPTICS [14] or DENCLUE [15]; (iv) grid-based methods, which quantize the data point space into a finite number of cells that form a grid structure; (v) model-based methods, which hypothesize a model for each of the clusters and find the best fit of the data to the given model.

Although there are currently hundreds of different clustering algorithms, not any single one can handle all problems due to the complicated variations of real-world data, including cluster shape, data size, data distribution, data noise, and specialized biases. Recently, one popular improvement is to introduce partial knowledge of label information into the traditional unsupervised clustering methods. This new type of clustering methods is usually referred to as semi-supervised clustering. Partial knowledge in semi-supervised clustering algorithms includes two types. The first is the true label information for a small number of data points, and the latter is the pairwise constraints between some data point pairs. The pairwise constraint usually contains two types, the Must-Link (ML) constraint and the Cannot-Link (CL) constraint [16]: for two data points x and y, a Must-Link (ML) constraint $ML(x, y)$ requires that these two data points must be assigned to the same class, while a Cannot-Link (CL) constraint $CL(x, y)$ requires them to be assigned to two different classes. In general, to decide whether two objects are in the same class or not is much easier than to get the class labels for these points. Therefore, semi-supervised clustering based on pairwise constraints becomes an active research topic in recent years, which is also referred to as constrained clustering. A lot of constrained clustering algorithms have recently been proposed to take advantage of these two kinds of pairwise constraints. For example, COPKmeans [16] tries to satisfy each constraint beyond considering distances to cluster centroids. Partial Constrained Kmeans (PCKmeans) [17] and Partial closure-based constrained Kmeans (PCCKmeans) [18] optimize some cost functions which take into consideration not only distances but also constraints. However, most of the existing constraint clustering use constraints chosen at random, i.e., by randomly selecting point pairs and then asking whether these point pairs come from the same class. This random method is in most cases neither effective nor efficient. Intuitively, the Cannot-Link constraints between point pairs which are far apart and the Must-Link constraints between nearby point pairs usually provide no additional contributions since these can be learned in the unsupervised clustering algorithms. In view of this problem, some researchers [17] [19] focus on the active constraints, which are usually se-

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

lected using an active mechanism. In this paper, we extend the PCCKmeans algorithm to a new active constraint-based clustering algorithm by introducing multiple representatives into constrained clustering with the active constraints which are selected using the Explore and Consolidate (EC) approach proposed by [17]. To our best knowledge, this work is the first effort to investigate multiple representatives in constrained clustering. Experimental results on several benchmark data sets and two public image data sets demonstrate the advantages of our algorithm.

II. Multiple representatives based image clustering with active constraints

In this section, we will first briefly introduce the Explore and Consolidate approach (EC) approach and two partial constrained Kmeans (PCKmeans [17] and PCCKmeans [18]), and then we will propose our new algorithm in detail.

A. the Explore and Consolidate approach (EC)

For a particular data set with k classes, the EC approach is used to find a disjoint k closures (a closure is a point set belonging to the same class) based on querying constraints between selected candidate points. EC includes two phases: (i) Explore and (ii) Consolidate. In the explore phase, points are selected using the farthest-first traversal algorithm until at least one point for each class have been found. After the explore phase, points are selected at random to be added to the existed k closures, which is referred to as the Consolidate phase in [17]. The interested reader is referred to [17] for further information.

B. Partial Constrained Kmeans (PCKmeans) and Partial closure-based constrained Kmeans (PCCKmeans)

Given a data set X, a set of ML constraints M, a set of CL constraints C, the corresponding penalty weights $w_{ij}(\vec{w_{ij}})$ for violating ML(CL) constraints and the number k of clusters, PCKmeans aims to find a disjoint k partitioning $\{X_h\}_{h=1}^{k}$ (each with its centroid μ_h) so as to minimize the following cost function

$$\begin{aligned} J_{pckm} &= \frac{1}{2}\sum_{h=1}^{k}\sum_{x_i\in X_h}\|x_i-\mu_h\|^2 \\ &+ \sum_{(x_i,x_j)\in M} w_{ij}\delta(L(x_i)\neq(L(x_j)) \\ &+ \sum_{(x_i,x_j)\in C} \vec{w}_{ij}\delta(L(x_i)=(L(x_j)) \end{aligned} \tag{1}$$

with the indicator function

$$\delta(true)=1, \delta(false)=0. \tag{2}$$

and $L(x_i)$ denoting the estimated cluster label for point x_i.

Similar to Kmeans, PCKmeans uses the traditional search scheme to get a greedy solution. Specifically, it first assigns points to clusters according to Eq. (1) and then calculates new cluster centroids as the mean of points in each cluster in each iteration, i.e.,

$$\mu_h = \frac{\sum_{x_i\in X_h} x_i}{|X_h|} \tag{3}$$

where $|X_h|$ is the cluster X_h's cardinality.

The PCCKmeans algorithm [18] introduces the closure property into the partial constrained Kmeans (PCKmeans) [17]. PCCKmeans firstly deduces a closure set, $\{c_i\}_{i=1}^{N_c}$ with the closure size $\{N_i\}_{i=1}^{N_c}$, from the Must-Link (ML) constraints. Note that, in PCCKmeans, the singular points are also regarded as a special kind of closures in which there is only a single point member (i.e., the size is 1). The Cannot-Link constraints between points are inherited by the closures to form a new Cannot-Link set C_c. PCCKmeans aims to find a k disjoint partition $\{X_h\}_{h=1}^{k}$ (with their centroids $\{\mu_h\}_{h=1}^{k}$) so as to minimize the following cost function

$$\begin{aligned} J_{pcckm} &= \sum_{h=1}^{k}\sum_{c_i\in X_h} N_i * \|m_{c_i}-\mu_h\|^2 + \\ &+ \sum_{(c_a,c_b)\in C_c} N_p * N_q * \delta(L(c_a)=L(c_b)) \end{aligned} \tag{4}$$

where m_{c_i} is the mean (centroid) of closure c_i, $L(c_a)$ is the estimated cluster label for closure c_a.

C. Active multiple representatives based partial constrained Kmeans (AMPCC)

Most of the constrained clustering algorithms, such as COPKmeans, PCKmeans and PCCKmeans, are assumed to be used with the non-active constraints. However, since active constraints are more powerful and effective, it is also meaningful to consider algorithms working under the active constraints [17] [19]. In this paper, we take further advantage of the active constraints by introducing multiple representatives into constrained clustering. We use the simple type of representatives in partitioning clustering algorithms, i.e., centroids, in AMPCC. More specifically, we use several subcentroids to represent a cluster in AMPCC. With the selected closures from the EC approach, we propose to use these closures to find *multiple* initial representatives for each cluster in our new algorithm AMPCC. Similar heuristics of multiple representatives are employed in several unsupervised clustering algorithms. For example, CURE uses multiple representatives to capture the shape of each cluster [9]. Chameleon constructs a k-nearest neighbor graph and merges the partitions of this graph to form final clusters [11]. However, these representatives are only derived based on some distances such that the representatives cannot be guaranteed to be correct for all of the clusters. In our algorithm, the multiple representatives are acquired from the constraints such that these representatives are definitely correct for each cluster. It is notable that the initialization of the clusters is important for the whole clustering. Our algorithm benefits from the fact that we take advantage of the active constraints in the initialization of the multiple representatives and the constrained clustering process.

Given a data set $X = \{x_i\}_{i=1}^N$, cluster number k, and a closure set $\{c_a\}_{a=1}^k$ with the size $\{N_a\}_{a=1}^k$ acquired from the EC approach, AMPCC finds initial subcentroids from the points in the closures. Note that, for the reason of fairness, the subcentroid number k_s for each cluster is set to the same number. Since the closure sizes are not all equal, it is difficult to determine the value of k_s. We use the traditional method to assign the value of k_s

$$k_s = \lceil \sqrt{\min(N_a)} \rceil, a = 1, 2, \cdots, k \tag{5}$$

where $\min(N_a)$ is to select the minimum value from the size lists for the closures (i. e. $N_a, a = 1, 2, \cdots, k$) and $\lceil Z \rceil$ is to round the value of Z to the nearest integer which is greater than itself . Therefore, our task is first to find $(k * k_s)$ subclusters with these initial subcentroids and then to merge the subclusters to k resulting clusters. Since the initial subcentroids are acquired from the constraints, we assign these subclusters with the initial subcentroids from a particular closure to a corresponding final cluster. Formally, denoted by $L_s(x)$ the subcluster label of point x (the possible value of $L_s(x)$ ranges from 1 to $k * k_s$) and $L(x)$ the final clustering label of point x, a fixed mapping between these two types of labels is imposed on all the points

$$L(x) = \lceil (L_s(x)/k_s) \rceil \tag{6}$$

Thus in AMPCC, we proceed with clustering by minimizing the following cost function

$$\begin{aligned} J_{ampcc} = & \sum_{h=1}^{k_s} \sum_{x_i \in X_h} (\|x_i - \mu_h\|^2) \\ & + \sum_a N_a \sum_{x_i \in c_a, x_j \in c_a} \delta(L(x_i) \neq L(x_j)) \\ & + \sum_{x_i \in c_a, x_j \in c_b, a \neq b} N_a N_b \delta(L(x_i) = L(x_j)) \end{aligned} \tag{7}$$

using the closure sizes as the penalty factors for constraint violation.

The subcentroid re-estimation is performed to compute the mean of the points as indicated in Eq. (3). The whole algorithm is summarized in Figure 1.

Algorithm 4: AMPCC
INPUT:
data set X,
cluster number k;
k closures $\{c_i\}_{i=1}^k$ from the EC approach, each for one cluster;
maximum iteration T;
OUTPUT:
clustered label result L of X.
METHOD:
1. calculate the common subcluster number k_s for all closures using (5);
2. perform kmeans clustering to acquire k_s subclusters for each cluster;
3. initialize clusters centroids $\{\mu_i\}_{i=1}^{k_s}$;
4. **Repeat**
5. assign subcluster label L_s to the points according to (7);
6. update subcentroids as the mean of points belonging to them;
7. **Until** L_s converges or maximum iteration T reaches
8. assign cluster label L to X according to (6) with L_s.
9. return L.

Fig. 1. The AMPCC algorithm

III. Experiments

A. Data sets

We perform experiments on two types of public data sets. All of the data sets used in the experiments are summarized in Table I, in which N represents the number of points of a data set, D represents the number of dimensions, K represents the number of classes, and "class distribution" represents the number of points in each class respectively.

The first three data sets are obtained from the well-known UCI machine learning repository[1], including Iris, Ionosphere and Dermatology. We perform the min-max normalization on all these three data sets in each dimension. For example, denote by min_v/max_v the minimum/maximum value for a particular dimension A respectively, the normalized value for v (of dimension A) is

$$v' = \frac{v - min_v}{max_v - min_v} \tag{8}$$

The first two Principal Component Analysis (PCA) [20] dimensions for these normalized three data sets are plotted in Figure 6 respectively.

We also test our algorithm in two image data sets: (i) Caltech5: a subset from the public Caltech101 data set [21] (The data set is available at the web site[2]), which including 5 classes (with numbers in parenthesis representing the number of images in each class): minaret (76), pagoda (47), pizza (53), schooner (63) and trilobite (86). Representative images are shown in Figure 4; (ii) SIMPLicity10: the SIMPLlcity data set contains 10 classes and each class has 100 images (The data set is available at the web site[3]). Representative images are shown in Figure 5. We extract four different kinds of features for these two image data sets, including Grid Color Moment, Edge, Gabor Wavelet Texture, and Local Binary Pattern (LBP) following [22]. To perform a suitable combination of the different features, we normalize each feature, say f_{ij} for the jth feature of the ith image, with their mean value μ_j and variance σ_j as follows

$$f_{ij}^{'} = \frac{f_{ij} - \mu_j}{\sigma_j} \tag{9}$$

where

$$\mu_j = \frac{\sum_i f_{ij}}{N}, \sigma_j = \sqrt{\sum_j (f_{ij} - \mu_j)^2} \tag{10}$$

and N is the number of images. After normalization, we use Principal Component Analysis (PCA) [20] to reduce the feature dimensionality from the original number 297 to 50 so as to reduce the computational complexity. The similarity matrices of these two data sets are shown in Figure 2 and Figure 3 respectively.

[1] http://archive.ics.uci.edu/ml/

[2] http://www.vision.caltech.edu/Image_Datasets/Caltech101/

[3] http://wang.ist.psu.edu/docs/related.shtml

TABLE I
SUMMARY OF THE DATA SETS USED IN THE EXPERIMENTS.

Data set name	Category	N	D	K	class distribution
Ionosphere	UCI	351	34	2	225, 126
Iris	UCI	150	4	3	50, 50, 50
Dermatology	UCI	366	34	6	61, 112, 72, 52, 49, 20
Caltech5	Image	325	50	5	76, 47, 53, 63, 86
SIMPLIcity10	Image	1000	50	10	100, 100, 100, 100, 100, 100, 100, 100, 100, 100

Fig. 4. examples from the Caltech5 data set

Fig. 5. examples from the SIMPLIcity10 data set

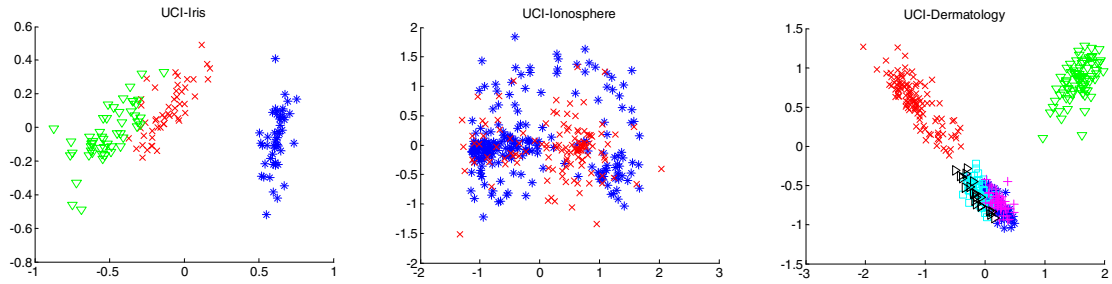

Fig. 6. Plot of the two main PCA dimensions of the three UCI data sets: (a)Iris, (b)Ionosphere, (c)Dermatology.

B. Evaluation Methodology

We use Normalized Mutual Information (NMI) [23] to evaluate the performance. NMI provides a measure of the extent to which two clusterings share statistical information. NMI is defined for two clustering labels X and Y as [23]

$$NMI(X,Y) = \frac{I(X,Y)}{\sqrt{H(X)H(Y)}} \tag{11}$$

where $I(X,Y)$ is the mutual information between X and Y, and $H(X)$ is the Shannon entropy of X. To provide better clarity, we use the following notations: the clustering X has k^X classes and there are N_i^X points in its ith class. For two clusterings with the class distributions $\{N_i^X\}_{i=1}^{k^X}$ and $\{N_i^Y\}_{i=1}^{k^Y}$ respectively, these entropy measurements can be calculated as

$$H(X) = -\sum_{i=1}^{k^X} \frac{N_i^X}{N^X} \log \frac{N_i^X}{N^X} \tag{12}$$

and

$$I(X,Y) = \sum_{i=1}^{k^X}\sum_{j=1}^{k^Y} \frac{N_i^X N_j^Y}{N^X N^Y} \log \frac{N_i^X N_j^Y}{N^X N^Y} \tag{13}$$

where

$$N^X = \sum_{i=1}^{k^X} N_i^X, N^Y = \sum_{i=1}^{k^Y} N_i^Y \tag{14}$$

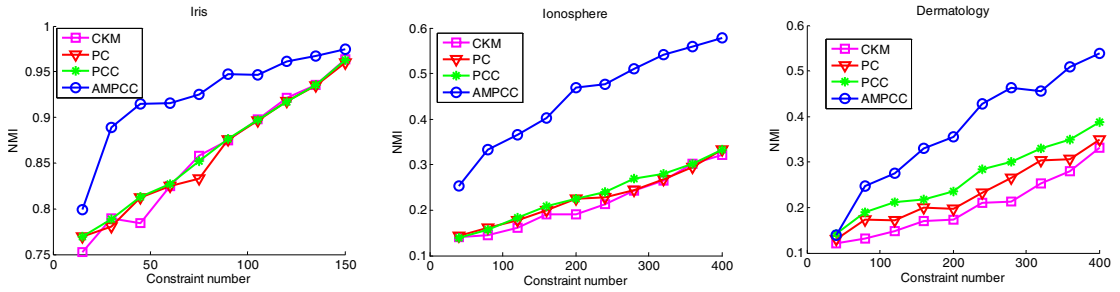

Fig. 7. clustering result of the three UCI data sets: (a)Iris, (b)Ionosphere, (c)Dermatology.

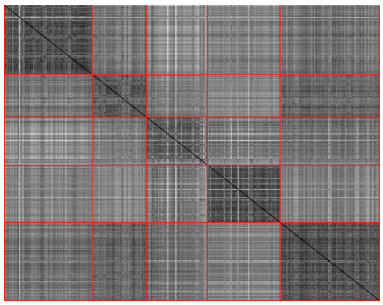

Fig. 2. Similarity matrix of the Caltech5 data set

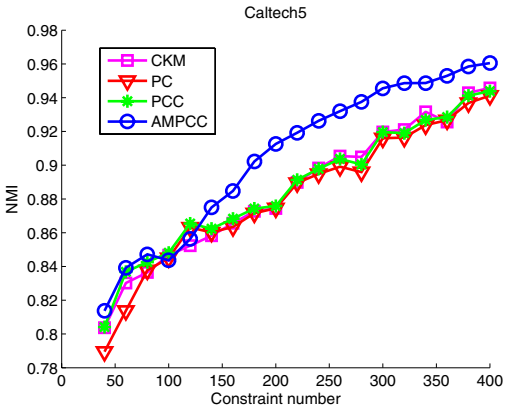

Fig. 8. clustering result of the Caltech5 data set

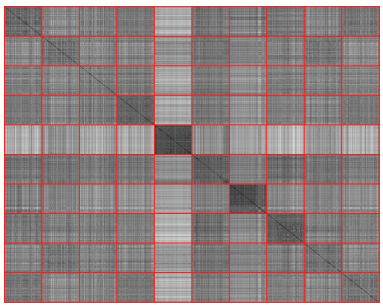

Fig. 3. Similarity matrix of the SIMPLIcity10 data set

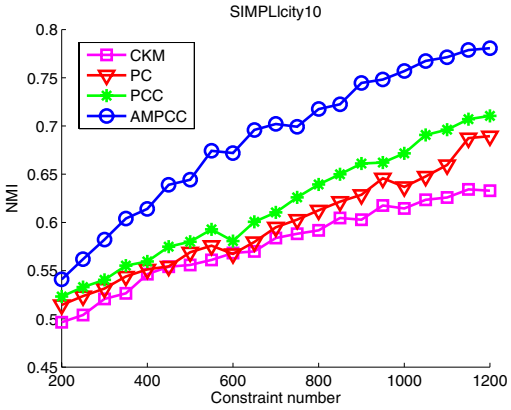

Fig. 9. clustering result of the SIMPLIcity10 data set

We compare our AMPCC algorithm with two well-known constrained clustering algorithms, COPKmeans (COP), PCK-means (PC) and its base algorithm PCCKmeans (PCC). For fair comparison, all the comparisons are made using the active constraints selected by the EC approach. In all the experiments, we run 20 trials for each data set and the mean results are reported.

C. Numerical Results

We first observe the distributions of the data sets themselves. From Figure 6, we can see that the Iris is the easiest data set while the Ionosphere data set and the Dermatology data set are the more difficult. For the image data sets in Figure 2 and Figure 3, we can see that the SIMPLIcity10 image data

set is less well defined than the Caltech5 image data set.

NMI results for the UCI data sets are shown in Figure 7, and those for the two image data set are shown in Figure 8 and Figure 9 respectively. For all the data sets, AMPCC is significantly better than the previous approaches, including its base algorithm PCCKmeans. This is because the centroid-based clustering algorithms usually assume that data within each cluster are approximately Gaussian distributed, while in general the practical features are too complicated for only Gaussian distributions to characterize. In AMPCC, for each cluster, the multiple subcentroids with the help of the active constraints have better ability to reduce the above problem. This observation is more evident for the data sets with fewer classes, e.g., Iris (with three classes) and Ionosphere (with two classes) in Figure 7. It is also interesting that in most cases the performance for AMPCC improves to a greater extent than those of the previous approaches when the number of active constraints increases.

Another interesting observation is that for those data sets with relatively larger sizes, AMPCC have similar performance with PCCKmeans using only a relatively small number of constraints (e.g., the Dematology data set with 40 constraints in Figure 7 and the Caltech5 image data set with constraints ranging from 40 to 120 in Figure 8). This is because in these cases, we have fewer chances to get enough points in each closure so as to obtain multiple centroids in each cluster. AMPCC performs approximately the same as PCCKmeans when the subcentroid number for each cluster is set to one. While the constraint number increases, the difference between these two algorithms becomes significant.

IV. Conclusion and Future work

In this paper, we investigate the multiple-cluster-representative based constrained clustering algorithm, AMPCC, with the active constraints selected using the Explore and Consolidate (EC) approach proposed by [17]. Our major contribution is to introduce the multiple cluster representatives into constrained clustering. Experimental results on several benchmark data sets and two public image data sets demonstrate the advantages of our algorithm.

In the future, we shall expand our current set of cluster representatives to other cluster representatives. It is also interesting to include different types of cluster representatives for the same clusters.

Acknowledgments

The work described in this paper was partially supported by grants from the Research Grants Council of Hong Kong Special Administrative Region, China [Project No. CityU 121607], and grants from the City University of Hong Kong [Project No. 7002141 and 7002374].

References

[1] J. W. Han and M. Kamber, *Data Mining: Concepts and Technique*, Elsevier Inc, 2006.

[2] Sergios Theodoridis and Konstantinos Koutroumbas, *Pattern Recognition, Third Edition*, Academic Press, 2006.

[3] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, 1967, vol. 1, pp. 281–297.

[4] L. Kaufman and P. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*, John Wiley and Sons, New York, NY., 1990.

[5] Raymond T. Ng and Jiawei Han, "Efficient and effective clustering methods for spatial data mining," in *VLDB*, 1994, pp. 144–155.

[6] F. Murtagh, "A survey of recent advances in hierarchical clustering algorithms," *Computer Journal*, vol. 26, no. 4, pp. 354–359., 1983.

[7] C. Olson, "Parallel algorithms for hierarchical clustering," *Parallel Computing*, vol. 21, pp. 1313–1325, 1995.

[8] S. Guha, R. Rastogi, and K. Shim, "ROCK: A robust clustering algorithm for categorical attributes," in *Proceedings of the 15th ICDE*, 1999, pp. 512–521.

[9] S. Guha, R. Rastogi, and K. Shim, "CURE: An efficient clustering algorithm for large databases," in *Proceedings of the ACM SIGMOD Conference*, 1998, pp. 73–84.

[10] T. Hang, R. Ramakrishnan, and M Livny, "BIRCH: an efficient data clustering method for very large databases," in *Proceedings of the ACM SIGMOD Conference*, 1996, pp. 103–114.

[11] George Karypis, Eui-Hong Han, and Vipin Kumar, "CHAMELEON: Hierarchical clustering using dynamic modeling," *IEEE Computer*, vol. 32, no. 8, pp. 68–75, 1999.

[12] M. Ester, H-P. Kriegel, J. Sander, and X Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the 2nd ACM SIGKDD*, 1996, pp. 226–231.

[13] Jörg Sander, Martin Ester, Hans-Peter Kriegel, and Xiaowei Xu, "Density-based clustering in spatial databases: The algorithm gdbscan and its applications," *Data Min. Knowl. Discov.*, vol. 2, no. 2, pp. 169–194, 1998.

[14] M. Ankerst, M. Breunig, H.-P. Kriegel, and J. Sander, "OPTICS: Ordering points to identify clustering structure," in *Proceedings of the ACM SIGMOD Conference*, 1999, pp. 49–60.

[15] Alexander Hinneburg and Daniel A. Keim, "An efficient approach to clustering in large multimedia databases with noise," in *Proceedings of the ACM SIGMOD Conference*, 1998, pp. 58–65.

[16] K.Wagstaff, C. Cardie, S. Rogers, and S. Schroed, "Constrained k-means clustering with background knowledge," in *Proc. of 18th Intl Conf. on Machine Learning*, 2001.

[17] S. Basu, M. Bilenko, and R. J. Mooney, "Active semi-supervision for pairwise constrained clustering," in *Conf of 4th SIAM Data Mining*, 2004.

[18] Shaohong Zhang and Hau-San Wong, "Partial closure-based constrained clustering with order ranking," in *Proceedings of the 2008 IEEE International Conference on Pattern Recognition*, Tampa, Florida, USA, 2008.

[19] Nizar Grira., Michel Crucianu, and Nozha Boujemaa, "Active semi-supervised fuzzy clustering," *Pattern Recognition*, vol. 5, no. 41, pp. 1834–1844, 2008.

[20] I.T. Jollife, *Principal Component Analysis*, Springer-Verlag, 1986.

[21] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories," in *IEEE. CVPR 2004, Workshop on Generative-Model Based Vision*, 2004.

[22] Jianke Zhu, Steven C.H. Hoi, Michael R. Lyu, and Shuicheng Yan, "Near-duplicate keyframe retrieval by nonrigid image matching," in *MM '08: Proceeding of the 16th ACM International Conference on Multimedia*, New York, NY, USA, 2008, pp. 41–50, ACM.

[23] Alexander Strehl and Joydeep Ghosh, "Cluster ensembles — a knowledge reuse framework for combining multiple partitions," *Journal of Machine Learning Research*, vol. 3, pp. 583–617, 2002.