Unsupervised and semi-supervised clustering for large image database indexing and retrieval

Lai Hien Phuong*†, Muriel Visani*, Alain Boucher† and Jean-Marc Ogier*
*L3I, Université de La Rochelle, 17042 La Rochelle cedex 1, France
Email: hien_phuong.lai@univ-lr.fr,
muriel.visani@univ-lr.fr, jean-marc.ogier@univ-lr.fr
†IFI, MSI team; IRD, UMI 209 UMMISCO; Vietnam National University, 42 Ta Quang Buu, Hanoi, Vietnam
Email: alain.boucher@auf.org

***Abstract*—In recent years, the expansion of acquisition devices such as digital cameras, the evolution of storage and transmission techniques of multimedia documents and the developement of tablet computers facilitate the growing of many large image databases as well as the interactions with the users. This increases the need for efficient and robust methods for finding information in these huge masses of data, including feature extraction methods and feature space structuring methods. The feature extraction methods aim to extract feature descriptors for each image. The feature space structuring methods organize indexed images in order to facilitate, accelerate and improve the results of further retrieval. Clustering is one kind of feature space structuring. Clustering may organize the dataset into groups of similar objects without prior knowledge (unsupervised clustering) or with a limited amount of prior knowledge (semi-supervised clustering). In this article, we present both formal and experimental comparisons of different unsupervised clustering methods for structuring large image databases. We use different image databases of increasing sizes (Wang, PascalVoc2006, Caltech101, Corel30k) to study the scalability of the different approaches. Moreover, a summary of semi-supervised clustering methods is presented and an interactive semi-supervised clustering model using the HMRF-kmeans is experimented on the Wang image database in order to analyse the improvement of the clustering results when user feedbacks are provided.**

I. Introduction

The traditional content-based image retrieval relies in general on two phases. The first phase is to extract the feature vectors of all the images in the database. The second phase is to compare the feature vector of the query image to that of all the other images in the database for finding the nearest images. With the development of many large image databases, the exhaustive search is not generally compatible. Feature space structuring methods (clustering, classification) are therefore necessary for organizing feature vectors of all images in order to facilitate and accelerate further retrieval.

Clustering aims to split a collection of data into groups (clusters) so that similar objects belong to the same group and dissimilar objects are in different groups. Because the feature vectors only capture low level information such as color, shape or texture of image (global descriptor) or of a part of an image (local descriptor), there is a semantic gap between high-level semantic concepts expressed by the user and these low-level features. The clustering results are therefore generally different from the intent of the user. Our final work aims involving the user into the clustering phase so that the user could interact with the system in order to improve the clustering results (the user may split or group some clusters, add new images, etc.). With this aim, we are looking for clustering methods which can be incrementally built in order to facilitate the insertion or deletion of images. The clustering methods should also produce a hierarchical cluster structure where the initial clusters may be easily merged or split. It can be noted that the incrementality is also very important in the context of very large image databases, when the whole dataset cannot be stored in the main memory. Another very important point is the computational complexity of the clustering algorithm, especially in an interactive context where the user is involved.

In the case of large image database indexing, we may be interested in traditional clustering (unsupervised) or semi-supervised clustering. While no information about the ground truth is provided in the case of unsupervised clustering, a limited amount of knowledge is available in the case of semi-supervised clustering. The provided knowledge may consist in class labels (for some objects) or pairwise constraints (must-link or cannot-link) between objects.

Some general surveys of unsupervised clustering techniques have been proposed in the literature [1], [2]. Jain *et al.* [1] presents an overview of different clustering methods and gives some important applications of clustering algorithms such as image segmentation or object recognition, but they did not present any experimental comparison of these methods. A well-researched survey of clustering methods is presented in [2], including analysis of different clustering methods and some experimental results, but the experiments are not specific to image analysis. There are three main contributions in this paper. First, we analyze the advantages and drawbacks of different unsupervised clustering methods in a context of huge masses of data where incrementality and hierarchical structuring are needed. Second, we experimentally compare four of these methods (global k-means [3], AHC [4], SR-tree [5] and BIRCH [6]) with different real image databases of increasing sizes (Wang, PascalVoc2006, Caltech101, Corel30k) (the number of images going from 1000 to 30000) to study the scalability of different approaches when the size of the database is increased. Third, we present some semi-supervised clustering methods and propose a preliminary experiment of an

interactive semi-supervised clustering model using the HMRF-kmeans (Hidden Markov Random Fields kmeans) clustering [31] on the Wang image database in order to analyse the improvement of the clustering process when user feedbacks are provided.

This paper is structured as follows. Section II presents both formal and experimental comparisons of some unsupervised clustering methods. Different semi-supervised clustering methods are described in section III. A preliminary experiment of an interactive semi-supervised clustering model is proposed in section IV. Section V presents some conclusions and further work.

II. Unsupervised clustering methods comparisons

Unsupervised clustering methods are divided into several types:

- Partitioning methods (k-means [7], k-medoids [8], PAM [9], CLARA [9], CLARANS [10], ISODATA [11], etc.) partition the dataset based on the proximities of the images in the feature space. These methods give in general a "flat" (*i.e.* non hierarchical) organization of clusters.
- Hierarchical methods (AGNES [9], DIANA [9], AHC [4], R-tree family [5], SS-tree [5], SR-tree [5], BIRCH [6], CURE [12], ROCK [13], etc.) organize the points in a hierarchical structure of clusters.
- Grid-based methods (STING [14], WaveCluster [15], CLICK [16], etc.) partition a priori the space into cells without considering the distribution of the data and then group neighbouring cells to create clusters. The cells may be organized in a hierarchical structure or not.
- Density-based methods (EM [17], DBSCAN [18], DENCLUE [19], OPTICS [20], etc.) aim to partition a set of points based on their local densities. These methods give a "flat" organization of clusters.
- Neural network-based methods (LVQ [21], SOM [21], ART [22], etc.) aim to group similar objects using the network and represent them by a single unit (neuron).

A. Formal comparison

As stated in section I, in our context, we need the clustering methods producing a hierarchical cluster structure. Among all five types of unsupervised clustering, the hierarchical methods always produce a hierarchical structure. We thus compare formally in Table I different hierarchical clustering methods (AHC, BIRCH, CURE, R-tree, SS-tree, SR-tree) towards some of the most popular methods of other types: k-means (partitioning methods), STING (grid-based methods), EM (density-based methods) and SOM (neural network-based methods). Different criteria (complexity, appropriateness to large databases, incrementality, hierarchical structure, data order dependence, sensitivity to outliers and parameter dependence) are used for the comparison.

K-means is not incremental, it does not produce any hierarchical structure. K-means is independent of the processing order of the data and does not depend on any parameter. Its computational and storage complexities can be considered as linear to the number of objects, it is thus suitable to large databases. The hierarchical methods (in italics) organize data in a hierarchical structure. Therefore, by considering the structure at different levels, we can obtain different numbers of clusters, which is useful in the context where users are involved. AHC is not incremental and it is not suitable to large databases because its computational and storage complexities are very high (at least quadratic to the number of elements). BIRCH, R-tree, SS-tree and SR-tree are by nature incremental because they are built by adding incrementally records. They are also adapted to large databases because of their relatively low computational complexity. CURE realizes the hierarchical clustering using only a random subset containing N_{sample} points of the database, the other points being associated to the closest cluster. Its computational complexity is thus relatively low and CURE is adapted to large databases. It is incremental but the results depend much on the random selection of the samples and the records which are not in this random selection have to be reassigned whenever the number of clusters k is changed. CURE is thus not suitable to the context where users are involved. STING, the grid-based method, divides the feature space into rectangular cells and organizes them according to a hierarchical structure. With a linear computational complexity, it is adapted to large databases. It is also incremental. However, as STING is used for spatial data and its attribute-dependent parameters have to be calculated for each attribute, it is not suitable to high dimensional data such as feature image space. Moreover, when the space is almost empty, hierarchical methods perform better than grid-methods. The EM density-based method is suitable to large databases because of its low computational complexity and is able to detect outliers. But it is very dependent on the parameters, it does not produce any hierarchical structure and is not incremental. SOM groups similar objects using a neural network which output layer contains neurons representing the clusters. SOM depends on initialization values and on the rules of influence of a neuron on its neighbors. It is incremental as the weight vectors of the output neurons can be updated when new data arrive. SOM is also adapted to large database, but it does not produce any hierarchical structure. We can conclude from this analysis that the methods BIRCH, R-tree, SS-tree and SR-tree are the most suitable to our context.

B. Experimental comparison

In this section, we present an experimental comparison of the partitioning method global k-means [3] with three hierarchical methods (AHC [4], SR-tree [5] and BIRCH [6]). Global k-means is a variant of the well known and widely used k-means method. The advantage of the global k-means is that we can automatically select the number of clusters k by stopping the algorithm at the value of k providing acceptable results. The other methods provide hierarchical clusters. AHC is chosen because it is the most popular method in the hierarchical family and there exists an incremental version of this method. Among four methods BIRCH, R-tree, SS-

TABLE I

FORMAL COMPARISON OF DIFFERENT CLUSTERING METHODS BASED ON DIFFERENT CRITERIA. METHODS IN GREY ARE CHOSEN FOR EXPERIMENTAL COMPARISON. FOR COMPLEXITY ANALYSIS, WE USE THE FOLLOWING NOTATIONS: N-NUMBER OF OBJECTS IN THE DATASET, k-NUMBER OF CLUSTERS, l-NUMBER OF ITERATIONS, N_{sample}-NUMBER OF SAMPLES CHOSEN, m-NUMBER OF TRAINING ITERATIONS, k'-NUMBER OF NEURONS IN THE OUTPUT LAYER.

Methods	Complexity	Appropriateness to large database	Incrementality	Hierarchical structure	Data order dependence	Sensitivity to outliers	Parameter Dependence
k-means [7] (partitioning)	$O(Nkl)$ (time) $O(N+k)$ (space)	Yes	No	No	No	Sensitive	No
AHC [4] (hierarchical)	$O(N^2logN)$ *(time)* $O(N^2)$ *(space)*	*No*	*Have incremental version*	*Yes*	*No*	*Sensitive*	*No*
BIRCH [6] (hierarchical)	$O(N)$ *(time)*	*Yes*	*Yes*	*Yes*	*Yes*	*Enable outliers detection*	*Yes*
CURE [12] (hierarchical)	$O(N^2_{sample}logN_{sample})$ *(time)*	*Yes*	*Able to add new points*	*Yes*	*No*	*Less sensitive*	*No*
R-tree, SS-tree, SR-tree [5] (hierarchical)	$O(NlogN)$ *(time)*	*Yes*	*Yes*	*Yes*	*Yes*	*Sensitive*	*Yes*
STING [14] (grid-based)	$O(N)$ (time)	Yes	Yes	Yes	No	Enable outliers detection	No
EM [17] (density-based)	$O(Nk^2l)$ (time)	Yes	No	No	No	Enable outliers detection	Yes
SOM [21] (neural network-based)	$O(k'Nm)$ (time)	Yes	Yes	No	Yes	Sensitive	Yes

tree, SR-tree that are most suitable to our context, we choose BIRCH and SR-tree because SR-tree combines the advantages of R-tree and SS-tree methods.

We compare the four selected clustering methods using different image databases of increasing size (Wang[1] (1000 images of 10 classes), PascalVoc2006[2] (5304 images of 10 classes), Caltech101[3] (9143 images of 101 classes) and Corel30k (31695 images of 320 classes)). Towards feature descriptors, we implement rgSIFT [23], a color SIFT descriptor that is widely used nowadays for its high performance. We use the color SIFT descriptor code of Koen van de Sande[4]. The "Bag of words" approach is chosen to group local features into a single vector representing the frequency of occurrence of the visual words in the dictionary [24]. The number of visual words in the dictionary (also called dictionary size) is fixed to 200. Both internal (Silhouette-Width (SW) [25]) and external measures (Rand Index [26]) are used in order to analyze the clustering results. While internal measures are low-level measures which are essentially numerical and unsupervised, external measures are high-level measures which give a supervised (semantic) evaluation based on the comparison between the clusters produced by the algorithm and the ground truth.

Figure 1 shows the result of the different clustering methods on the different image databases of increasing sizes. The results show that SR-tree gives the worst results on the Wang image database, it is not used anymore on larger databases (PascalVoc2006, Caltech101, Corel30k). The AHC method is not used on the Corel30k image database because of the lack of RAM memory. In fact, the AHC clustering requires a large amount of memory when processing more than 10000 elements, while the Corel30k contains more than 30000 images. We can see that the internal and external measures do not evaluate the same aspects and give very different results. The external measures are closer to the user's attempts. The results show that, according to internal measures, the best method varies from each database while BIRCH is always the best method regardless of the size of the database according to external measures (which are more suitable to the context where users are involved). Moreover, in comparison to global k-means and AHC, BIRCH is much faster, especially in the

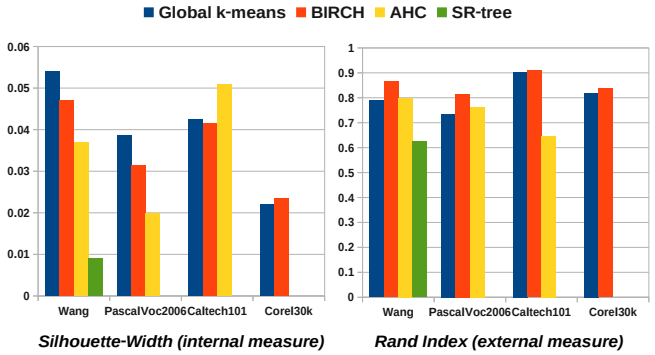

Fig. 1. Comparison of different unsupervised clustering methods (Global k-means, SR-tree, BIRCH, AHC) on different image databases (Wang, PascalVoc2006, Caltech101, Corel30k) using the local feature descriptor rgSIFT with a dictionary of size 200. Both internal measure (Silhouette-Width) and external measure (Rand Index) are used. The higher are these measures, the best are the results.

[1]http://wang.ist.psu.edu/docs/related/
[2]http://pascallin.ecs.soton.ac.uk/challenges/VOC/
[3]http://www.vision.caltech.edu/Image_Datasets/Caltech101/
[4]http://staff.science.uva.nl/~ksande/research/colordescriptors/

case of the Caltech101 and Corel30k image databases (*e.g* the execution time of BIRCH in the case of the Corel30k is about 400 times faster than that of the global k-means).

III. Semi-supervised clustering methods

In semi-supervised clustering, some prior knowledge is available, either in the form of class labels (for some objects) or in the form of pairwise constraints between some objects. Pairwise constraints specify whether two objects should be in the same cluster (must-link constraint) or in different clusters (cannot-link constraint). This prior knowledge is used to guide the clustering process.

Some semi-supervised clustering methods using prior knowledge in the form of labeled objects have been proposed in the literature: seeded-kmeans [28], constrainted-kmeans [28], etc. Seeded-kmeans and constrained k-means are based on the k-means algorithm. Prior knowledge of these two methods is a small subset of the input database, called *seed set*, containing user-specified labeled objects of k different clusters. Rather than initializing randomly the clustering, these two methods initialize their k cluster centers using different partitions of the *seed set*. The second step of the seeded-kmeans is to apply the k-means algorithm on the whole database without considering the prior labels of the objects in the *seed set*. In constrast, the constrained-kmeans applies the k-means algorithm while keeping the label of user-specified objects unchanged. An interactive cluster-level semi-supervised clustering was proposed in [29]. In this model, knowledge is not provided a priori, it is progressively provided as assignment feedbacks and cluster description feedbacks of users after each interactive iteration. Using assignment feedback, the user moves an object from one of the current clusters to another. Using cluster description feedback, the user modifies the feature vector of any current cluster, for example, by increasing the weights of some important words (note that this method is implemented for document analysis). The algorithm learns from all feedbacks provided in earlier stages to re-cluster the dataset in order to minimize the sum of distance between points and corresponding cluster centers while minimizing the violation of constraints corresponding to feedbacks.

Some semi-supervised clustering methods that use prior knowledge in the form of constraints between objects are COP-kmeans (constrained k-means) [30], HMRF-kmeans (Hidden Markov Random Fields Kmeans) [31], etc. In COP-kmeans, each point is assigned to the closest cluster while respecting the constraints; the clustering fails if no solution respecting the constraints is found. In HMRF-kmeans, constraint violation is allowed with a violation cost (penalty). The violation cost of a pairwise constraint may be either a constant or a function of the distance between the two points specified in the pairwise constraint. In order to ensure the respect of the most difficult constraints, higher penalties are assigned to violations of must-link constraints between points that are distant. With the same idea, higher penalties are assigned to violations of cannot-link constraints between points which are close in the feature space. HMRF-kmeans initializes the k cluster centers based on the user-specified constraints and unlabeled points, as described in [31]. After the initialization step, an iterative algorithm is applied to minimize the objective function (which is the sum of distances between points and corresponding centers with the penalties of violated constraints). The iterative algorithm consists in three steps:

- E-step: Re-assign each data point to the cluster which minimizes its contribution to the objective function.
- M-step (A): Re-estimate the cluster centers to minimize the objective function.
- M-step (B): If the distance between points are estimated by a parameterized distortion measure, the parameters of the distortion measure are subsequently updated to reduce the objective function.

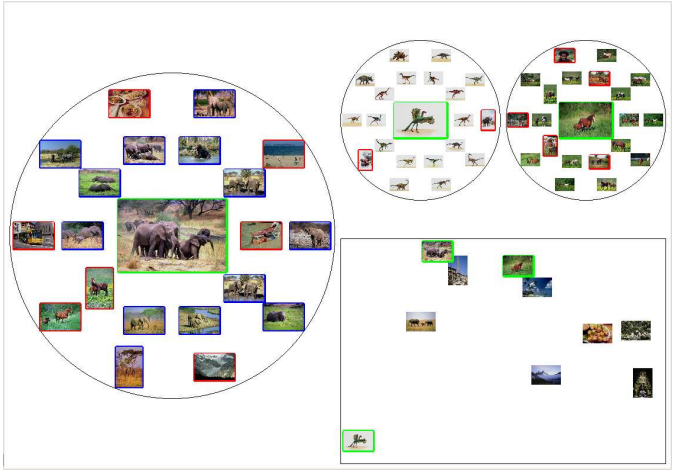

Fig. 2. 2D interactive interface representing the results of the Wang image database. The rectangle at the bottom right corner represents the principal plane consisting of the two first principal axis (obtained by PCA) of the representative images of all clusters. Each circle represents the details of a particular cluster selected by the user.

IV. Interactive semi-supervised clustering experimentation

In this section, we present some experimental results of an interactive semi-supervised clustering model on the Wang image database. The initial clustering is realized without any prior knowledge, using k-means. We implement an interactive interface that allows the user to view the clustering results and to provide feedbacks to the systems. Using Principal Component Analysis (PCA), all the representative images (one for each cluster) are presented in the principal plane (the rectangle at the bottom right corner of Figure 2, the principal plane consists of the two principal asis associated with the highest eigenvalues). User can view the details of some clusters by clicking the corresponding representative images. In our experiments, we use the internal measure Silhouette-Width (SW) [25] to estimate the quality of each image in a cluster. The higher is the SW value of an image in a cluster, the more compatible is this image for this cluster. In Figure 2, each cluster selected by the user is represented

by a circle: the image at the center of the circle is the most representative image (image with the highest SW value) of this cluster; the 10 most representative images (images with the highest SW values) are located near the center and the 10 least representative images (images with the smallest SW values) are located near the border of a cluster. User can specify positive feedbacks and negative feedbacks (respectively images with blue and red border in Figure 2) for each cluster. User can also change the cluster assignment of a given image. When an image is changed from a cluster A to a cluster B, it is considered as a negative feedback for cluster A and a positive feedback for cluster B. While only positive images of a cluster are used to derive must-link constraints, both positive and negative images are needed for deriving cannot-link constraints. After receiving feedbacks from the user, the HMRF-kmeans is applied to re-cluster the whole dataset using pairwise constraints derived from feedbacks accumulated from all earlier stages. The interactive process is repeated until the clustering result satisfy the user. Note that the distorsion measure used in our first experimentation is the Euclidian distance because of its simplicity and its popularity in the image domain.

1) Experimental protocol: In order to automatically realize the interactive tests, we implement an agent later called "user agent" that simulates the behaviors of the user when interacting with the system (assuming that the agent knows all the ground truth which contains the class label of each image). At each interactive iteration, clustering results are returned to the user agent by the system; the agent simulates the behaviors of the user to give feedbacks to the system. The system then uses these feedbacks to update the clustering. Note that the clustering results returned to the user agent are the most representative images (one for each cluster) and their positions in the principal plane. When the agent user views a cluster, the 10 most and 10 least representative images of this cluster are displayed.

For simulating the user's behaviors, we proposed some simple rules:

- At each iteration, the user agent chooses to view a fixed number of c clusters.
- There are two strategies for choosing clusters by the user agent: randomly choose c clusters, or choose iteratively two closest clusters until having c clusters.
- The user agent determines the image class (in the ground truth) corresponding to each cluster by the most represented class among the 21 shown images. The number of images in this class must be greater than a threshold $MinImages$. If it is not the case, this cluster can be considered as a noise cluster.
- When there are several clusters (among chosen clusters) that correspond to a same class, the user agent chooses the cluster in which the images of this class are the most numerous (among the 21 shown images of the cluster) as the principal cluster of this class. The classes of the other clusters are redefined as usual, but neutralizing the images from this class.

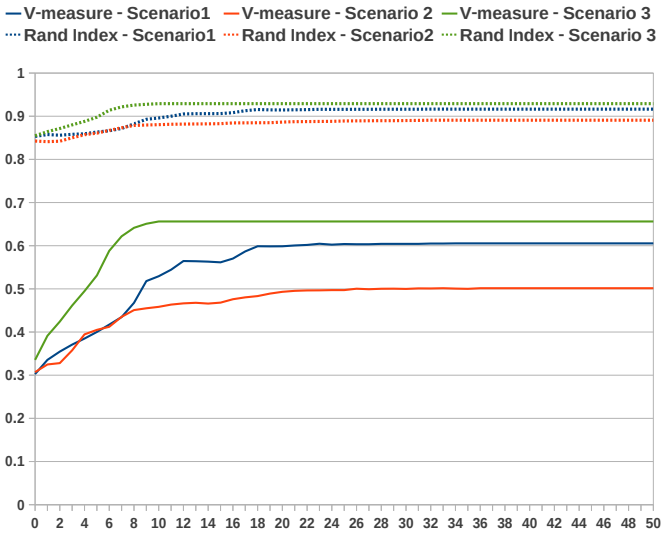

Fig. 3. Results of the automatic test of interactive semi-supervised clustering on the Wang image database using rgSIFT. Three scenarios and two external measures (V-measure, Rand Index) are used. The horizontal axis specifies interaction iterations (iteration 0 means the initial k-means without prior knowledge).

- In each chosen cluster, all images such that the result of the algorithm corresponds to the ground truth are positive samples of this cluster, while the others are negative samples of this clusters. All negative samples are moved to the cluster corresponding to their true class in the ground truth.

We propose three test scenarios for experiments on the Wang image database. Note that the number of clusters k in the clustering is equal to the number of classes (10) in the ground truth. We set the threshold $MinImages = 5$ for all three scenarios. In scenarios 1 and 2, we use $c = 5$ clusters for interacting, while in scenario 3, we use all the clusters ($c = 10$). In scenario 1, clusters are randomly chosen (strategy 1) for interacting, while we iteratively choose the closest clusters (strategy 2) in scenario 2.

2) Experimental results and discussions: Figure 3 presents the results of the three previous scenarios on the Wang image database using two external measures (Rand Index [26] and V-measure [27]). The external measures compare the clustering results with the ground truth that is compatible to estimate the quality of the interactive clustering after receiving feedbacks from the user. The local feature descriptor rgSIFT with a dictionary of size 200 is used for these tests. We can see that for all these three scenarios, the clustering results are improved after each interactive iteration, in which the system re-clusters the dataset following the feedbacks accumulated from the previous iterations. However, after some iterations, the clustering results converge. This may be due to the fact that no new knowledge is provided to the system because the 21 images shown to the user remain unchanged. Another strategy consisting in showing only the images that were not previously presented to the user might be interesting.

Moreover, we can see that the clustering results converge more quichly when the number of chosen clusters at each iterative iteration is high (scenario 3 converges more quickly than scenarios 1 and 2). Performing automatic tests on larger databases (PascalVoc2006, Caltech101, Corel30k) is a part of our future work.

V. Conclusion

There are three contributions of this paper. Firstly, this paper compares formally different unsupervised clustering methods in the context of large image databases where incrementality and hierarchical structuring are needed. We can conclude from this analysis that the methods R-tree, SS-tree, SR-tree and BIRCH are most suitable to our context because their computational complexities are not high, that makes them adapted to large databases. Moreover, these methods are by nature incremental, so that they are promising to be used in the context where the user is involved.

Secondly, we compare experimentally different unsupervised clustering methods using different image databases of increasing size. In comparison to the AHC, SR-tree and global k-means clustering methods, BIRCH is more efficient in the context of large image databases.

Thirdly, we propose in this paper an interactive model, using the semi-supervised clustering method HMRF-kmeans, in which the knowledge is accumulated from the feedbacks of the user at every interactive iterations. The results of the three automatic test scenarios, using an user agent for simulating the user's behaviors, show an improvement of the clustering results with the accumulation of the user feedbacks in the clustering process.

Our future work aims to replace the k-means method by the BIRCH clustering method into the interactive semi-supervised clustering model in order to improve the clustering results of this method.

Acknowledgment

Grateful acknowledgement is made for financial support by the Poitou-Charentes Region (France).

References

[1] A. K. Jain, M. N. Murty, P. J. Flynn, *Data clustering: A review*. ACM Computing Surveys, 31:264-323, 1999.

[2] R. Xu and D. I. I. Wunsch, *Survey of clustering algorithms*. IEEE Transactions on Neural Networks, 16(3):645-678, 2005.

[3] A. Likas, N. Vlassis, J. Verbeek, *The global k-means clustering algorithm*. Pattern Recognition, 36(2):451-461, 2003.

[4] G.N. Lance, W.T. Williams, *A general theory of classification sorting strategies. II. Clustering systems*. Computer journal, pp. 271-277, 1967.

[5] N. Katayama, S. Satoh, *The SR-tree: An index structure for High-Dimentional Nearest Neighbor Queries*. In Proc. of the ACM SIGMOD International Conference on Management of Data, Tucson, Arizon, pp. 369-380, 1997.

[6] T. Zhang, R. Ramakrishnan, M. Livny, *BIRCH: An efficient data clustering method for very large databases*. SIGMOD Rec. 25, 2:103-114, 1996.

[7] J. McQueen, *Some methods for classification and analysis of multivariate observations*. In Proc. of 5th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281-297, 1967.

[8] S.A. Berrani, *Recherche approximative de plus proches voisins avec contrôle probabiliste de la précision; application à la recherche dimages par le contenu*, PHD thesis, 210 pages, 2004.

[9] L. Kaufman, P.J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley, Newyork, 368 pages, 1990.

[10] R. T. Ng and J. Han, *CLARANS: A Method for Clustering Objects for Spatial Data Mining*. IEEE Transaction on Knowledge and Data Engineering, 14(5):1003-1016, 2002.

[11] G. Ball, D. Hall, *A clustering technique for summarizing multivariate data*. Behavior Science, 12(2):153-155, 1967.

[12] S. Guha, R. Rastogi, K. Shim, *CURE: An Efficient Clustering Algorithms for Large Databases*. In Proc. of the ACM SIGMOD International Conference on Management of Data, Seattle, WA, pp. 73-84, 1998.

[13] S. Guha, R. Rastogi, K. Shim, *ROCK: A Robust Clustering Algorithm for Categorical Attributes*. In Proc. of the 15th IEEE International Conference on Data Engineering (ICDE), pp. 512-521, 1999.

[14] W. Wang, J. Yang, R. Muntz, *STING: A Statistical Information Grid Approach to Spatial Data Mining*. In Proc. of the 23th VLDB, Athens, Greece, pp. 186-195, 1997.

[15] G. Sheikholeslami, S. Chatterjee, A. Zhang, *WaveCluster: A Multi-Resolution Clustering Approach for Very Large Spatial Databases*. In Proc. of the 24th VLDB, New York, NY, USA, pp. 428-439, 1998.

[16] R. Agrawal, J. Gehrke, D. Gunopulos, P. Raghavan, *Automatic subspace clustering of high dimensional data for data mining applications*. In Proc. of the ACM SIGMOD International Conference on Management of data, New York, NY, USA, pp. 94-105, 1998.

[17] G. Mclachlan, T. Krishnan, *The EM algorithm and extensions*. Wiley, New York, NY, 304 pages, 1997.

[18] M. Ester, H-P. Kriegel, J. Sander, X. Xu, *A density-based algorithm for discovering clusters in large spatial databases with noise*. In Proc. of the 2nd International Conference on Knowledge Discovery and Data Mining, pp. 226-231, 1996.

[19] A. Hinneburg, D.A. Keim, *A general approach to clustering in large databases with noise*. Knowledge and Information Systems, 5(4):387-415, 2003.

[20] M. Ankerst, M. M. Breunig, H.P. Kriegel, J.Sande, *OPTICS: ordering points to identify the clustering structure*. In Proc. of the 1999 ACM SIGMOD International Conference on Management of Data, pp. 49-60, 1999.

[21] M. Koskela, *Interactive image retrieval using Self-Organizing Maps*. PhD thesis, Helsinki University of Technology, Dissertations in Computer and Information Science, Report D1, Espoo, Finland, 2003.

[22] G. Carpenter, S. Grossberg, *ART3: Hierarchical search using chemical transmitters in self-organizing pattern recognition architectures*. Neural Networks 3, pp. 129-152, 1990.

[23] K.E.A. van de Sande, T. Gevers, C.G.M. Snoek, *Evaluation of color descriptors for object and scene recognition*. IEEE Proc. of the Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Anchorage, Alaska, 2008.

[24] J. Sivic, A. Zisserman, *Video Google: A text retrieval approach to object matching in videos*. In Proc. of IEEE International Conference on Computer Vision (ICCV), Nice, France, pp. 1470-1477, 2003.

[25] P. J. Rousseeuw, *Silhouettes: a graphical aid to the interpretation and validation of cluster analysis*. Journal of Computational & Applied Mathematics, 20, pp. 53-65, 1987.

[26] W. M. Rand, *Objective criteria for the evaluation of clustering methods*. Journal of the American Statistical Association, 66(336):846-850, 1971.

[27] A. Rosenberg, J. Hirschberg, *V-measure: A conditional entropy-based external cluster evaluation measure*. In Joint Conference on Empirical Methods in Natural Language Processing and Computational Language Learning, Prague, pp. 410-420, 2007.

[28] S. Basu, A. Banerjee, R. J. Mooney, *Semi-supervised clustering by seeding*. Proc. of 19th International Conference on Machine Learning (ICML-2002), pp. 19-26, 2002.

[29] A. Dubey, I. Bhattacharya, S. Godbole, *A cluster-level semi-supervision model for interactive clustering*. Machine Learning and Knowledge Discovery in Databases, volume 6321 of Lecture Notes in Computer Science, Springer Berlin/ Heidelberg, pp. 409-424, 2010.

[30] K. Wagstaff, C. Cardie, S. Rogers, S. Schroedl, *Constrained k-means clustering with background knowledge*. Proc. of 18th ICML, pp. 577-584, 2001.

[31] S. Basu, M. Bilenko, A. Banerjee, R. J. Mooney, *Probabilistic semi-supervised clustering with constraints*. In O. Chapelle, B. Scholkopf, and A. Zien, editors, Semi-Supervised Learning. MIT Press, Cambridge, MA, 2006.