EasyChair Preprint

№ 532

Improved Architectural Redesign of MTree Clusterer in the Context of Image Segmentation

Marius Andrei Ciurez and Marian Cristian Mihaescu

September 27, 2018

Improved Architectural Redesign of MTree Clusterer in the Context of Image Segmentation

Marius Andrei Ciurez, Marian Cristian Mihaescu

University of Craiova, Craiova, Romania
mariusandrei.ciurez@gmail.com;mihaescu@software.ucv.ro

Abstract. Image segmentation by clustering represents a classical use-case of unsupervised learning. A key aspect of this problem is that instances that are being clusters may have various types and thus requesting specific algorithms that implement particular distance functions and quality metrics. This paper presents an improved version of MTree clusterer that has been tested in the context of image segmentation in the same setup as a new recently k-MS algorithm. The redesigned MTree algorithms allows many levers for setup so that many configurations are available depending on the particularities of the tackled problem. The experimental results are promising especially as compared with the ones from previous MTree version and also as compared with classical clustering algorithms or newly developed k-MS algorithm. Further improvements in terms of available algorithms for configuration and algorithmic efficiency of integrations may lead the way to a general purpose clusterer that may be used for processing various data types.

Keywords: clustering, MTree, image segmentation.

1 Introduction

Clustering algorithms have found many application domains where unsupervised learning provides efficient solutions to tackled problems. Among the most well known application domains there are medical image processing (i.e., pattern recognition and image segmentation) [1, 2], general and natural language processing knowledge discovery [3, 4, 11], navigation of robots [5, 6] and in many other contexts.

In the area of unsupervised learning there are several general classes of clustering algorithms (i.e., flat, hierarchical and density based) that all share two common problems: finding the optimal number of clusters and quickly and efficiently finding the correct clusters taking into consideration specific distance measures appropriate appropriate for the objects (i.e., pixels, points, persons, books, etc.) that are being grouped.

The objective of this work is to present an improved version of the MTree clustering algorithm [7] that is currently implemented as a Weka package [8, 9]. The improved version has been tested in a comparative benchmark with k-MS

morphological reconstruction clustering algorithm [10] as well as classical algorithms such as simple k-means, Cobweb, Farther First and Canopy.

The proposed approach tackles the practical problem of recognizing shapes as described in [10] by improving MTree clustering algorithm in terms of dataset preprocessing for finding optimal number of clusters and adjusting the business logic of the clusterer in terms of division policy and distance metric between instances. As compared with the initial results obtained by MTree clusterer reported in [10] we conclude that current version provides significantly better results than initial version and in several aspects challenges the clustering algorithms used in benchmarking process. The progress of MTree clusterer from the initial version consists in several improvements from algorithmic and implementation perspectives. The experimental results are validated by classical clustering quality metrics as in [10].

The paper is organized as follows. In Section 2, we perform a literature review with regards to finding optimal K (i.e., the number of clusters), clustering algorithms in Weka, division policies in clustering and validation by clustering quality metrics. Section 3 describes the proposed approach with a detailed presentation of each module from the clustering data analysis process with focus on algorithmic challenges.We also perform a complexity analysis of the newly obtained algorithm compared with the older one and with the other clustering algorithms used in the comparative analysis. In Section 4, we present experimental results that compare the quality and time performance of MTree implementation with other clustering algorithms. Finally, Section 5 contains the conclusions of this work, summarizes the key approaches of the improved version of the MTree algorithm and discusses potential improvements and applications.

2 Related Work

Data clustering is represented by classical area of unsupervised machine learning that come in many flavours and have found their way in image clustering or segmentation [12]. From this perspective, a wide range of variations we proposed in the literature.

Dhanachandra et. al. in [12] use subtractive clustering along classical K-means algorithm in order to preprocess the data for optimal centroid initialization. The experimental results were obtained on medical images representing infected blood cells with malaria and on classical images used for segmentation obtaining better results than k-means taking into account RMSE and PSNR metrics.

A more elaborate approach for image clustering was proposed by Chang et. al. in [13]. The propose a Deep Adaptive Clustering (DAC) approach that reduces to a classification problem in which similarity is determined by cosine distance and learned labeled features tend to be one-hot vectors obtaining good results on popular datasets like MNIST, CIFAR-10 and STL-10.

Retrieval of similar images from an image database (CBIR – Content Based Image Retrieval) represents a challenging task that has been addressed in [14] and [KK]. The first approach uses as features color and texture and employs K-means and

hierarchical clustering for finding the the most similar images. The second approach uses color, texture and shape as features and K-means as business logic for building four different groups of images: dinosaurs, flowers, busses and elephants. The obtained experimental results are promising in terms of good precision and recall values.

A more complex context occurs when the image source is unknown or when the ground truth for the training dataset is also unknown [15, 16]. In this situation, optimal K represents a critical issue as well as using an efficient distance function such that usage of a particular loss function provides good experimental results. These approaches propose as solution an workaround hierarchical clustering and clustering ensembles based graph partitioning methods, such as Cluster-based Similarity Partitioning Algorithm (CSPA), Hyper Graph Partitioning Algorithm (HGPA), and Meta CLustering Algorithm (MCLA).

Another critical aspect of unsupervised learning is represented by the optimal number of clusters that reside in the dataset. Unfortunately, scenarios in which the value of K is known occur in only a subset of practical scenarios. In general, image processing applications do not have a value of K that is known beforehand. This may occur when dealing with data streams [17] or with very high-dimensional datasets [18]. In general, the most suitable approach reduces to automatic determination of K that may be based on dynamic clustering [19] or joint tracking segmentation [20].

Finally, the whole clustering process needs validation, and this may be accomplished by many quality metrics for a wide range of algorithms [24]. Depending on the structure of the dataset various clustering quality frameworks [22, 23] have been proposed. The key issue that always arises regards choosing the proper similarity and quality metrics [23].

3 Proposed Approach

The proposed approach follows a classical data analysis pipeline that is appropriate for unsupervised learning and in presented in Figure 1. The input is represented by one image that is preprocessed in order to load the pixels and build an *.arff* file suitable for processing by Weka clustering algorithm implementations.

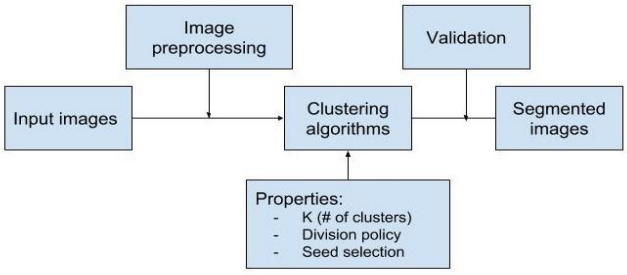

Fig. 1. Block diagram of the clustering analysis benchmark

The clustering analysis benchmark uses several classical clustering implementations such as k-means, Cobweb, Farthest First and Canopy along our own improved version of MTree, the k-MS algorithm and other utility algorithms such as xMeans, voteK, Hierarchical Clustering, EM or Cascade k-Means.

Running the clustering algorithms within the benchmark is managed by several key properties. One regards all the clustering algorithms and is represented by the number of clusters which are searched in the input image. The other settings are the division policy, seed selection mechanism and number of seeds and these apply only to MTree clusterer. All other clusterers use the same k as MTree along with other default settings. This approach makes possible further comparative analysis of various configurations of our MTree clusterer with clusterers that are already implemented in Weka and with k-MS algorithm reported in [10]. Another key aspects regard the number of seeds that are taken into consideration and the order in which data points are provided as input. These settings are provided as levers for MTree configurations and may influence the quality of the clustering results. As general rules, a minimum number of seeds needs to be taken into consideration as in many other clustering algorithms such that global optima is not missed due to a local optima. As for the order in which instances are provided to the MTree clusterer a random choosing approach represents a baseline scenario.

Finally, the clustering analysis benchmark returns a set of segmented images along with their corresponding validation metrics. For current approach we use two validation techniques: SSE and visual analytics.

The key component of the clustering analysis benchmark is represented by the clustering algorithms module and especially by the settings that accompany the MTree cluster and represent the core improvements in terms its capabilities and efficiency.

Figure 2 presents the algorithmic infrastructure of the clustering analysis benchmark with emphasis on the list of clustering algorithms and main options in terms of possible settings for used algorithms in general and for MTree clusterer in particular.

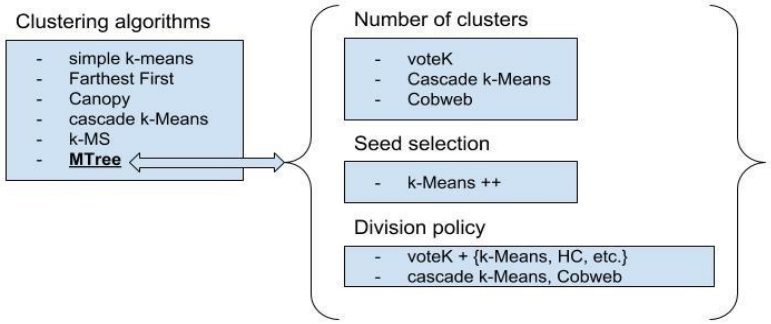

Fig. 2. Infrastructure of the clustering analysis benchmark

Finally, we describe the key improvements of MTree implementation as compared with previous one presented in [7]. The frist improvement regards the logic of split method that is performed for a full node. In this regard, there are two issues that were addresses: one regards the number of clusters and one regards the division policy. In the improved MTree the number of clusters may be set before running, but we may also leave this parameter to be determined at runtime by specifying a particular algorithm for determining the optimal k in the input dataset.

According with the value of k (i.e., specified or not specified) the splitting procedure uses an appropriate division policy. Thus, if the value of k is known, than the division policy is performed by an algorithm which require a value for k as input (i.e., k-Means, Farthest First). On the contrary, if k is not known, the division polity is performed by an algorithm which does not need a value for k, such that x-Means, Cascade k-means, EM or Cobweb.

A final improvement in MTree regards the seed selection, as a general issue in clustering data. The current approach uses random seed selection and selection based on k-means++ algorithm.

All algorithmic choices were made such that they are available in Weka and can be integrated in the business logic of the MTree and in the infrastructure of the clustering analysis benchmark.

4 Experimental Results

All the processing is performed on the image from [10] which reports good results for the proposed k-MS algorithm and poor results for MTree which justify current improvements.

The input image is preprocessed such that an *arff* file with two features is obtained. As in [10] the features are represented by the numeric values representing the cartesian coordinates of 9163 points. These input points are given as input to all configurations of MTree parametrized by various methods within split procedure.

Figure 3 presents a comparative result of the five MTree configurations versus five classical algorithms. Performed experiments use MTree configurations that integrate the voteK algorithm for getting the optimal number of clusters along with Cobweb (MT_vK_CW), Farther First (MT_vK_FF), Canopy (MT_vK_C), Hierarchical Clustering (MT_vK_HC) and Cascade simple K-Means (MT_vK_cSKM) algorithms within the split procedure.

All MTree configurations provide computed SSE values as well as classical simple k-Means. Therefore, all the obtained results from Figure 3 have as optimality criteria the minimum value for SSE and for providing a sound comparative analysis the themberof clusters was set to eight. The minimum SSE criteria and K equals to eight were chosen in order to have similar context with experimental results from [10].

The other algorithms do not provide values for SSE because of two reasons: either this functionality is not implemented in Weka (i.e., Farthest First, Canopy) or the algorithm itself - by its inner logic - is not suited for computing SSE values due to

lack of notion of centroid (i.e., k-MS, Cobweb). This is the reason why visual analytics is employed as a second evaluation technique.

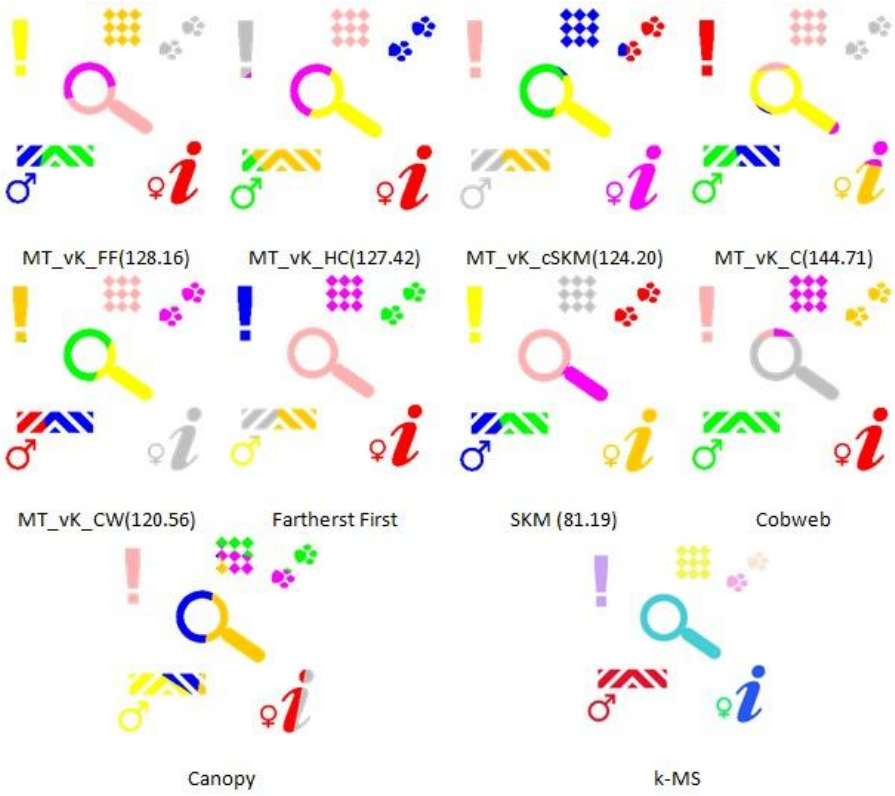

Fig. 3. Comparison between MTree results and other classical clustering algorithms

Therefore, visual (i.e., manual) analytics was used to evaluate the suboptimal clustering distributions, that is distributions that have larger SSE value although provide a better clustering. Figure 4 presents the visual analytics results for the five MTree configurations that were used in the clustering analysis benchmark.

Initial results show slight arguable improvements in two configurations (i.e., MT_vK_HC, MT_vK_C and MT_vK_cSKM) and better improvements in three configurations (i.e., MT_vK_FF, MT_vK_CW).

Two key settings for the experiments regard the number of seeds and the order in which data points are provided as input. Presented results were obtained after runs on 100 seeds as this is the usual default value in such situations. The data points were streamed to the MTree clusterer in random order. As we are dealing with images, experiments show a degradation of accuracy that is poor clustering results in terms of SSE values and correctly segmented images when data points are given into a particular (i.e., row-wise or column-wise) order.

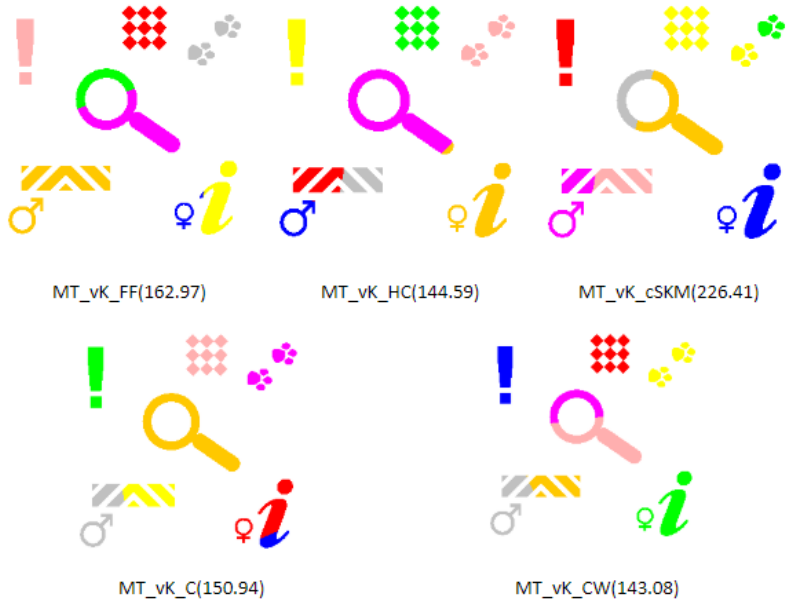

Fig. 4. Visual analytics results of MTree configurations

The distributions obtained by all MTree configurations are much better than the result reported in [10]. Still, the MT_vK_FF and MT_vK_CW configurations that use the SSE metric are arguable better than classical algorithms. The other three configurations, MT_vK_HC, MT_vK_cSKM and MT_vK_C are much better that their corresponding classical algorithms but do not outperform k-MS. The advantage of MTree algorithm resides in the speed by which it clusters new images once a clusterer has been trained.

5 Conclusions

Current study tackles the problem of image clustering. It provides an improved version of the MTree algorithm that is used for image segmentation in the same context as previously discussed in [10]. Improvements of MTree take into account the algorithmic approach that is is based on the split method in which the number of clusters, the seed selection and the division policy are key ingredients which have been parameterized such that various configurations may be obtained.

We performed experiments in various configurations and presented the ones that use the same setup as in [10] for a reproducible and comparative analysis. The improved MTree package along with voteK method for choosing optimal K are open-source and available in MTree Clusterer package [9].

Current results of all MTree configurations that were taken into consideration are highly improved as compared with initial one used in [10] and challenge classical clustering algorithms and k-MS. An advantage of the MTree clusterer is the feasibility

for customization such that it may process other data types (i.e., educational data) compared with k-MS that may work only for images.

Further improvements should take into account other clustering quality metrics and distances that may be better suited for this particular problem or for similar problems. Having access to SSE values for other clustering algorithms implemented in Weka and which have centroids and distances may provide a better objective comparative analysis. Observing that visual analytics may obtain slightly better distributions opens the way the need to take into consideration other relevant aspects that may automatically provide optimal solutions.

References

1. Silva, L. F., Santos, A. A. S., Bravo, R. S., Silva, A. C., Muchaluat-Saade, D. C., & Conci, A. (2016). Hybrid analysis for indicating patients with breast cancer using temperature time series. Computer methods and programs in biomedicine, 130, 142-153.
2. Goswami, S., & Bhaiya, L. K. P. (2013, April). Brain tumour detection using unsupervised learning based neural network. In Communication Systems and Network Technologies (CSNT), 2013 International Conference on (pp. 573-577). IEEE.
3. Gan, G., Ma, C., & Wu, J. (2007). Data clustering: theory, algorithms, and applications (Vol. 20). Siam.
4. Miñarro-Giménez, J. A., Kreuzthaler, M., & Schulz, S. (2015). Knowledge Extraction from MEDLINE by Combining Clustering with Natural Language Processing. In AMIA Annual Symposium Proceedings (Vol. 2015, p. 915). American Medical Informatics Association.
5. Di Caro, G. A., Ducatelle, F., & Gambardella, L. (2012, June). A fully distributed communication-based approach for spatial clustering in robotic swarms. In Proceedings of the 2nd Autonomous Robots and Multirobot Systems Workshop (ARMS), affiliated with the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)(Valencia, Spain, June 5) (pp. 153-171).
6. Gauci, M., Chen, J., Li, W., Dodd, T. J., & Gross, R. (2014, May). Clustering objects with robots that do not compute. In Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems (pp. 421-428). International Foundation for Autonomous Agents and Multiagent Systems.
7. Mihaescu, M. C., & Burdescu, D. D. (2012). Using m tree data structure as unsupervised classification method. Informatica, 36(2).
8. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: an update. ACM SIGKDD explorations newsletter, 11(1), 10-18.
9. MTree Clusterer, http://weka.sourceforge.net/packageMetaData/MTreeClusterer/index.html
10. Rodrigues, É. O., Torok, L., Liatsis, P., Viterbo, J., & Conci, A. (2017). k-MS: A novel clustering algorithm based on morphological reconstruction. Pattern Recognition, 66, 392-403.
11. Traian Rebedea, Costin-Gabriel Chiru, Ştefan Trăuşan-Matu, News Web Portal based on Natural Language Processing, http://rochi.utcluj.ro/rrioc/en/rochi2008.html

12. Dhanachandra, N., Manglem, K., & Chanu, Y. J. (2015). Image segmentation using K-means clustering algorithm and subtractive clustering algorithm. Procedia Computer Science, 54(2015), 764-771.
13. Chang, J., Wang, L., Meng, G., Xiang, S., & Pan, C. (2017, October). Deep Adaptive Image Clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 5879-5887).
14. Maheshwari, M., Silakari, S., & Motwani, M. (2009, July). Image clustering using color and texture. In Computational Intelligence, Communication Systems and Networks, 2009. CICSYN'09. First International Conference on (pp. 403-408). IEEE.
15. Caldelli, R., Amerini, I., Picchioni, F., & Innocenti, M. (2010, December). Fast image clustering of unknown source images. In Information Forensics and Security (WIFS), 2010 IEEE International Workshop on (pp. 1-5). IEEE.
16. Dash, A., Chatterjee, S., Prasad, T., & Bhattacharyya, M. (2016). Image clustering without ground truth. arXiv preprint arXiv:1610.07758.
17. Guha, S., & Mishra, N. (2016). Clustering data streams. In Data Stream Management (pp. 169-187). Springer, Berlin, Heidelberg.
18. Esmin, A. A., Coelho, R. A., & Matwin, S. (2015). A review on particle swarm optimization algorithm and its variants to clustering high-dimensional data. Artificial Intelligence Review, 44(1), 23-45.
19. Ozturk, C., Hancer, E., & Karaboga, D. (2015). Dynamic clustering with improved binary artificial bee colony algorithm. Applied Soft Computing, 28, 69-80.
20. Milan, A., Leal-Taixé, L., Schindler, K., & Reid, I. (2015, June). Joint tracking and segmentation of multiple targets. In Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on (pp. 5397-5406). IEEE.
21. Sleit, A., Qatawneh, M., Al-Sharief, M., Al-Jabaly, R., & Karajeh, O. (2011). Image Clustering using Color, Texture and Shape Features. KSII Transactions on Internet & Information Systems, 5(1).
22. Castellanos, A., Cigarrán, J., & García-Serrano, A. (2017). Formal concept analysis for topic detection: a clustering quality experimental analysis. Information Systems, 66, 24-42.
23. dos Santos, T. R., & Zárate, L. E. (2015). Categorical data clustering: What similarity measure to recommend?. Expert Systems with Applications, 42(3), 1247-1260.
24. Fahad, A., Alshatri, N., Tari, Z., Alamri, A., Khalil, I., Zomaya, A. Y., ... & Bouras, A. (2014). A survey of clustering algorithms for big data: Taxonomy and empirical analysis. IEEE transactions on emerging topics in computing, 2(3), 267-279.