A Split-Merge Framework for Evolutionary Clustering*

Veselka Boeva[1], Milena Angelova[2], Vishnu Manasa Devagiri[1], and Elena Tsiporkova[3]

[1] Blekinge Institute of Technology, Karlskrona, Sweden {veselka.boeva, vishnu.manasa.devagiri}@bth.se
[2] Technical University of Sofia, Plovdiv, Bulgaria mangelova@tu-plovdiv.bg
[3] EluciDATA Lab, Sirris, Brussels, Belgium elena.tsiporkova@sirris.be

Abstract. In this abstract, we discuss an ongoing work on a split-merge framework for evolutionary clustering. The proposed clustering technique is designed to be robust to concept drift scenarios by providing the flexibility to compute clusters on a new portion of data and to update the existing clustering solution by the computed new one. The split-merge framework models two clustering solutions as a bipartite graph. We have initially evaluated and compared the discussed evolutionary clustering technique with another bipartite correlation clustering algorithm (PivotBiCluster) on two different case studies.

Keywords: Evolutionary clustering · Unsupervised learning · Concept drift · Split-merge framework

1 Motivation and state of the art

In many applied fields such as document retrieval, customer profiling, text mining etc. new data is continuously generated at a rapid rate. As the data increases we need to re-group existing data and also accommodate new data points in the existing data categories. However, the existing original categories can become outdated caused by changing characteristics of the newly arrived data as a result of different external factors. This outdating of models is in fact, a concept drift and requires that the clustering techniques used, can deal with such a concept drift and enable reliable and scalable model update.

Incremental clustering solutions are maintained by accommodating newly arriving data points in the existing solution by either adding them to an existing cluster or placing it as a new singleton while two existing clusters are merged into one [3]. Clustering techniques such as incremental clustering or one-pass stream clustering [7] are designed to address the scalability issues of the clustering task, but these algorithms are not robust to concept drift phenomenon as the clustering solution is built on the entire data stream. This implies that

* This work is part of the research project "Scalable resource efficient systems for big data analytics" funded by the Knowledge Foundation (grant: 20140032) in Sweden.

changes in the characteristic of newly arriving data are not given the desired higher importance while building the clustering solution.

An interesting dynamic clustering algorithm which is equipped with dynamic split-and-merge operations and which is also dedicated to incremental clustering of data streams is proposed by Lughofer in [6]. In [5] similarly to the approach of Lughofer a set of splitting and merging actions are defined, where optional splitting and merging actions are only triggered during the iterative process when the conditions are met. Wang et al. also propose a split-merge-evolve algorithm for clustering data into k number of clusters [10]. This algorithm has the ability to optimize the clustering result in scenarios where new data samples may be added in to existing clusters. However, a k cluster output is always provided by the algorithm, i.e. it is also not sensitive to the evolution of data.

In [2], we have proposed a new split-merge evolutionary clustering algorithm which is robust to concept drift scenarios. The algorithm is designed to update the existing clustering solutions based on the data characteristics of newly arriving data by either splitting or merging existing clusters. The discussed clustering scenario is different from the one treated by incremental clustering. Our algorithm computes clusters on a new portion of data collected over a defined time period and then updates the existing clustering solution by the computed new clusters. The idea for the proposed clustering technique is inspired by the work of Xiang et al. [9]. Similarly to their approach we have designed a split-merge framework which models two clustering solutions as a bipartite graph. In order to study and evaluate the performance of the proposed split-merge technique it is compared with other two state of the art algorithms: PivotBiCluster [1] and Dynamic split-and-merge [6].

PivotBiCluster algorithm, introduced in [1], is also a bipartite correlation clustering algorithm similarly to our split-merge evolutionary clustering algorithm [2]. The PivotBiCluster algorithm starts by randomly selecting a node from the left side of the bipartite graph and forming a cluster with all its adjacent nodes from the right. Then all the remaining nodes on the left side are iterated to check if a node needs to be merged with the cluster or be a singleton, or finally left without any change for the next iteration. Evidently, in the final clustering some clusters are obtained by merging clusters from both side of the graph. However, existing clusters cannot be split by the PivotBiCluster algorithm even if the corresponding correlations with clusters from the newly extracted data elements reveal that these clusters are not homogeneous.

The dynamic split-and-merge algorithm of Lughofer can be used as an extension to any existing incremental and evolutionary clustering algorithm provided it stores details regarding cluster centers, spread, elements of a cluster [6]. Once the newly arriving data points are assigned to existing clusters by applying some incremental clustering algorithm, all the modified clusters are then examined in order to identify whether they need to be split or merged. Although, the dynamic split-and-merge algorithm addresses the clustering dynamics, it is not very sensitive to concept drift phenomenon, because it assigns the newly arriving data points to the existing clusters in an incremental way and then improves the

clustering solution by either splitting or merging the modified clusters. In comparison our split-merge clustering technique provides the flexibility to compute clusters on a new portion of data collected over a defined time period and to update the existing clustering solution by the computed new one [2]. Such an updating clustering should better reflect the current characteristics of the data by being able to examine clusters occurring in the considered time period and eventually capture interesting trends in the area.

It can also be observed that some of the characteristics of evolutionary clustering models such as ability to handle volume, velocity, variety can be related to the ideas implemented in stream reasoning models [4].

2 A split-merge framework for evolutionary clustering

In [2], we propose a split-merge framework that can be used to adjust the existing clustering solution to newly arrived data. The proposed framework models two clusterings (the existing and the newly constructed one) as a bipartite graph which is decomposed into connected components (bi-cliques). Each component is further analysed and if it is necessary it is decomposed into subcomponents. The subcomponents are then taken into consideration in producing the final clustering solution. For example, if an existing cluster is *overclustered*, i.e. it intersects two or more clusters in the new clustering, it is split between those. If several existing clusters intersect the same new cluster, i.e. they are *underclustered*, they are merged with that cluster.

Let us formally describe the proposed split-merge evolutionary clustering technique. The input bipartite graph is $G = (C, C', E)$, where C and C' are sets of clusters and E is a subset of $C \times C'$ that represents correlations between the nodes of the two sets. The algorithm consists of two main steps:

1. At the first step, all bi-cliques of G are found. Then we treat three different scenarios: (i) If a bi-clique is an unreachable node it is made a singleton in the final clustering solution; (ii) If a bi-clique connects a node from the left side of G with several nodes from C' the elements of this node are split among the corresponding nodes from C'; (iii) In the opposite case, i.e., when we have a bi-clique that connects a node from the right side of G with several nodes from left those nodes have to be merged with that node (cluster).
2. At the second step, the remained bi-cliques are decomposed into split/merge subcomponents. Each bi-clique, which is a bipartite graph, is transformed into a tripartite graph constructed by two (split and merge) bipartite graphs. Suppose $G_i = (C_i, C'_i, E_i)$ is the considered bi-clique. Then the corresponding tripartite graph is built by the following two bipartite graphs: $G_{iL} = (C_i, E_i, E_{iL})$ and $G_{iR} = (E_i, C'_i, E_{iR})$, where C_i, C'_i and E_i are ones from G_i, E_{iL} is a subset of $C_i \times E_i$ that represents correlations between the nodes of C_i and E_i, and E_{iR} is a subset of $E_i \times C'_i$ representing correlations between the nodes of E_i and C'_i. First all overclustered nodes of G_{iL} are split and new temporary clusters are formed as a result. Then we perform the corresponding merging for all underclustered nodes in G_{iR}.

3 Conclusions and future work

In this work, we have discussed an ongoing study for implementing and evaluating of a split-merge framework for evolutionary clustering. The proposed clustering technique has been initially evaluated and compared to PivotBiCluster algorithm [2]. The two algorithms have been tested and demonstrated in two different case studies: expertise retrieval and patient profiling in healthcare. Our algorithm has shown better performance than the PivotBiCluster in most of the studied experimental scenarios.

Our future plans are to pursue further comparison and evaluation of the proposed clustering technique with other existing dynamic clustering algorithms on richer data sets and in new case studies from different application domains. The latter is motivated by the work of Luxburg et al. [8], where they argue that the clusterings should not be treated as domain-independent mathematical problems, i.e. it is desirable to be evaluated in more than one end-user domain. We currently design and implement experimental scenarios for further comparison of our clustering technique with the dynamic split-and-merge algorithm of Lughofer [6].

In a long-term perspective, we are interested in building upon the proposed split-merge algorithm and develop measures for monitoring clusters evolution and mining changes. This might be treated as time-series forecasting problem where we need to forecast the changes in the clustering solution that might occur. Other interesting future direction is to use the proposed split-merge framework for developing a continual and shared learning technique that enable to learn from multiple data sources by continual updating and evolving of the model.

References

1. Alion, N. et al.: Improved approximation algorithms for bipartite correlation clustering. In 19th European Symposium on Algorithms, ESA 2011 pp. 25–36.
2. Boeva, V., Angelova, M., Tsiporkova, E.: A Split-Merge Evolutionary Clustering Algorithm. In Proc. of ICAART 2019, pp. 337–346.
3. Charikar, M. et al.: Incremental clustering and dynamic information retrieval. In 29th Annual ACM Symposium on Theory of Computing, STOC 1997 pp. 626–635.
4. DellAglio, D., Valle, E.D., van Harmelen, F., Bernstein, A.: Stream reasoning: A survey and outlook. Data Science 2017. vol. 1, pp. 59–83.
5. Fa, R. and Nandi, A. K.: Smart: Novel self splitting-merging clustering algorithm, In Proc. of European Signal Processing Conference, 2012.
6. Lughofer, E.: A dynamic split-and-merge approach for evolving cluster models. Evolving Systems 2012. vol. 3, pp. 135–151.
7. O'Callaghan, L. et al.: Streaming-Data Algorithms for High-Quality Clustering, In Proc. of IEEE International Conference on Data Engineering 2001, pp. 685–694.
8. von Luxburg, U. et al.: Clustering: Science or art? In JMLR: Workshop and Conference Proceedings, vol.27, 2012, pp. 65–79.
9. Xiang, Q. et al.: A Split-Merge Framework for Comparing Clusterings. In ICML 2012 pp. 1055–1062.
10. Wang, M. et al.: A Novel Split-Merge-Evolve k Clustering Algorithm. In Proc. of IEEE 4th Int. Conference on Big Data Computing Service and Applications, 2018.