A Split-Merge Evolutionary Clustering Algorithm

Veselka Boeva[1], Milena Angelova[2] and Elena Tsiporkova[3]
[1]*Computer Science and Engineering Dept., Blekinge Institute of Technology, Karlskrona, Sweden*
[2]*Computer Systems and Technologies Dept., Technical University of Sofia, Plovdiv, Bulgaria*
[3]*The Collective Center for the Belgian Technological Industry, Brussels, Belgium*

Keywords: Data Mining, Evolutionary Clustering, Bipartite Clustering, PubMed Data, Unsupervised Learning.

Abstract: In this article we propose a bipartite correlation clustering technique that can be used to adapt the existing clustering solution to a clustering of newly collected data elements. The proposed technique is supposed to provide 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. 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. For example, some clusters will be updated by merging with ones from newly constructed clustering while others will be transformed by splitting their elements among several new clusters. The proposed clustering algorithm, entitled *Split-Merge Evolutionary Clustering*, is evaluated and compared to another bipartite correlation clustering technique (PivotBiCluster) on two different case studies: expertise retrieval and patient profiling in healthcare.

1 INTRODUCTION

In this work, we are interested in developing evolutionary clustering techniques that are suited for applications affected by concept drift phenomena. In many practical applications such as, expertise (or document) retrieval systems the information available in the system database is periodically updated by collecting (extracting) new data. The available data elements, e.g., experts in a given domain, are usually partitioned into a number of disjoint subject categories. It is becoming impractical to re-cluster this large volume of available information. Profiling of users with wearable applications with the purpose to provide personalized recommendations is another example. As more users get involved one needs to re-cluster the initial clusters and also assign new incoming users to the existing clusters. In the context of profiling of machines (industrial assets) for the purpose of condition monitoring the existing original clusters can become outdated caused by aging of the machines and degradation of performance due to influence of changing external factors. This outdating of models is in fact a concept drift and requires that the clustering techniques, used for deriving the original machine profiles, can deal with such a concept drift and enable reliable and scalable model update.

Incremental clustering methods process one data element at a time and maintain a good solution by either adding each new element to an existing cluster or placing it in a new singleton cluster while two existing clusters are merged into one (Charikar et al., 1997). Incremental algorithms also bear a resemblance to one-pass stream clustering algorithms (O'Callaghan et al., 2001). Although, one-pass stream clustering methods address the scalability issues of the clustering problem, they are not sensitive to the evolution of the data, because they assume that the clusters are to be computed over the entire data stream.

The clustering scenario discussed herein is different from the one treated by incremental clustering methods. Namely, the evolutionary clustering techniques considered in this work are supposed to provide 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. 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. We propose and study two different clustering algorithms to be suited for the discussed scenario: *PivotBiCluster* (Ailon et al., 2011) and *Split-Merge Evolutionary Clustering*. Both algo-

Boeva, V., Angelova, M. and Tsiporkova, E.
A Split-Merge Evolutionary Clustering Algorithm.
DOI: 10.5220/0007573103370346
In *Proceedings of the 11th International Conference on Agents and Artificial Intelligence (ICAART 2019)*, pages 337-346
ISBN: 978-989-758-350-6

rithms are bipartite correlation clustering algorithms that do not need prior knowledge about the optimal number of clusters in order to produce a good clustering solution. In the final clustering generated by the PivotBiCluster algorithm some clusters are obtained by merging clusters from both side of the graph, i.e. some of existing clusters will be updated by some of the computed new ones. However, existing clusters cannot be split by the PivotBiCluster algorithm even the corresponding correlations with clusters from the newly extracted data elements reveal that these clusters are not homogeneous. This has motivated us to develop a new *Split-Merge Evolutionary Clustering* algorithm that overcomes this disadvantage. Namely, our algorithm is able to analyze the correlations between two clustering solutions and based on the discovered patterns it treats the existing clusters in different ways. Thus some clusters will be updated by merging with ones from newly constructed clustering while others will be transformed by splitting their elements among several new clusters.

The rest of the paper is organized as follows. Section 2 reviews related works. Section 3 briefly discusses the PivotBiCluster algorithm and describes the proposed Split-Merge Evolutionary Clustering technique. Section 4 introduces the two case studies used to compare and evaluate the two algorithms. Section 5 presents the evaluation of the proposed evolutionary clustering algorithm in expertise retrieval and patient profiling in healthcare domains and discusses the obtained results. Section 6 is devoted to conclusions and future work.

2 RELATED WORK

The model of incremental algorithms for data clustering is motivated by practical applications where the demand sequence is unknown in advance and a hierarchical clustering is required. Incremental clustering methods process one data element at a time and maintain a good solution by either adding each new element to an existing cluster or placing it in a new singleton cluster while two existing clusters are merged into one (Charikar et al., 1997).

To qualify the type of cluster structure present in data, Balcan introduced the notion of clusterability (Balcan et al., 2008). It requires that every element be closer to data in its own cluster than to other points. In addition, Balcan showed that the clusterings that adhere to this requirement are readily detected offline by classical batch algorithms. On the other hand, it was proven by Ackerman (Ackerman and Dasgupta, 2014) that no incremental method can discover these partitions. Thus, batch algorithms are significantly stronger than incremental methods in their ability to detect cluster structure.

Incremental algorithms also bear a resemblance to one-pass clustering algorithms for data stream problems (O'Callaghan et al., 2001). Such algorithms need to maintain a substantial amount of information so that important details are not lost. For example, the algorithm in (O'Callaghan et al., 2001) is implemented as a continuous version of k-means algorithm which continues to maintain a number of cluster centers which change or merge as necessary throughout the execution of the algorithm. In addition, Lughofer proposes a dynamic clustering algorithm which is equipped with dynamic split-and-merge operations and which is also dedicated to incremental clustering of data streams (Lughofer, 2012). In (Fa and Nandi, 2012) similarly to the approach of Lughofer a set of splitting and merging action conditions are defined, where optional splitting and merging actions are only triggered during the iterative process when the conditions are met. Although, one-pass stream clustering methods address the scalability issues of the clustering problem, they are not sensitive to the evolution of the data because they assume that the clusters are to be computed over the entire data stream.

The clustering scenario discussed herein is different from the one treated by incremental clustering methods. Namely, the evolutionary clustering technique proposed in this work is supposed to provide 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.

Gionis et al. proposed an approach to clustering that is based on the concept of aggregation (Gionis et al., 2007). They are interested in a problem in which a number of different clusterings are given on some data set of elements. The objective is to produce a single clustering of the elements that agrees as much as possible with the given clusterings. Clustering aggregation provides a framework for dealing with a variety of clustering problems. For instance, it can handle categorical or heterogeneous data by producing a clustering on each available attribute and then aggregating the produced clusterings into a single result. Another possibility is to combine the results of several clustering algorithms applied on the same dataset etc. Clustering aggregation can be thought as a more general model of multi-view clustering proposed in (Bickel and Scheffer, 2004). The multi-view approach considers clustering problems in which the available attributes can be split into two independent subsets. A clustering is produced on each subset and then the two clusterings are combined into a single result. Consensus clustering algorithms deal with similar problems to those treated by clustering aggregation techniques. Namely, such algorithms try to reconcile clustering information about the same data set coming from different sources (Boeva et al., 2014) or from different runs of the same algorithm (Goder and Filkov, 2008). The both clustering techniques are not suited for our scenario, since they are used to integrate a number of clustering results generated on one and the same data set.

An interesting split-merge-evolve algorithm for clustering data into k number of clusters is proposed by Wang et al. (Wang et al., 2018). The algorithm randomly divides data into k clusters initially, then repeatedly splits bad clusters and merges closest clusters to evolve the final clustering result. 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 not sensitive to the evolution of the data, as well.

The idea for the proposed *Split-Merge Evolutionary Clustering* algorithm is inspired by the work of Xiang et al. (Xiang et al., 2012). They have proposed a split-merge framework that can be tailored to different applications. The framework models two clusterings as a bipartite graph which is decomposed into connected components, and each component is further decomposed into subcomponents. Pairs of related subcomponents are then taken into consideration in designing a clustering similarity measure within the framework.

3 METHODS

3.1 Description of the Framework

Let us formalize the cluster updating problem we are interested in. We assume that X is the available set of data points and each data point is represented by a vector of attributes (features). In addition, the data points are partitioned into k groups, i.e. $C = \{C_1, C_2, \dots, C_k\}$ is an existing clustering solution of X and each C_i $(i = 1, 2, \dots, k)$ can be considered as a disjoint cluster. In addition, a new set X' of recently extracted data elements (samples) is created, i.e. $X \cap X'$ is an empty set. Each data point in X' is again represented by a list of attributes and $C' = \{C'_1, C'_2, \dots, C'_{k'}\}$ is a clustering solution of X'. The objective is to produce a single clustering of $X \cup X'$ by combining C and C' in such a way that the obtained clustering realistically reflects the current distribution in the domain under interest.

3.2 Pivot Bi-Clustering Algorithm

Two existing clustering techniques are suitable for the considered context: correlation clustering (Bansal et al., 2004) and bipartite correlation clustering (Ailon et al., 2011). The latter algorithm seems to be better aligned to our clustering scenario. In Bipartite Correlation Clustering (BCC) a bipartite graph is given as input, and a set of disjoint clusters covering the graph nodes is output. Clusters may contain nodes from either side of the graph, but they may possibly contain nodes from only one side. A cluster is thought as a bi-clique connecting all the objects from its left and right counterparts. Consequently, a final clustering is a union of bi-cliques covering the input node set. We compare our Split-Merge correlation clustering algorithm described in the following section with *PivotBi-Cluster* realization of the BCC algorithm (Ailon et al., 2011). The PivotBiCluster algorithm is implemented according to the original description given in (Ailon et al., 2011).

Notice that in our considerations the input graph nodes of the PivotBiCluster algorithm are clusters and in the final clustering some clusters are obtained by merging clusters (nodes) from both sides of the graph, i.e. some of the existing clusters will be updated by some of the computed new ones. However, existing clusters cannot be split by the BCC algorithm even the corresponding correlations with clusters from the newly extracted data elements reveal that these clusters are not homogeneous.

3.3 Split-Merge Evolutionary Clustering Algorithm

In this paper, we propose an evolutionary clustering algorithm that overcomes the above mentioned disadvantage of BCC algorithm. Namely, our algorithm is able to analyze the correlations between two clustering solutions C and C' and based on the discovered patterns it treats the existing clusters (C) in different ways. Thus, some clusters will be updated by merging with ones from newly constructed clustering (C') while others will be transformed by splitting their elements among several new clusters. One can find some similarity between our idea and an interactive clustering model proposed in (Awasthi et al., 2017). In this model, the algorithm starts with some initial clustering of data and the user may request a certain cluster to be split if it is *overclustered* (intersects two or more clusters in the target clustering). The user may also request to merge two given clusters if they are *underclustered* (both intersect the same target cluster).

As it was already mentioned in Section 2 our evolutionary clustering algorithm is inspired by a split-merge framework proposed by Xiang et al. in (Xiang et al., 2012). By modeling the intrinsic relation between two clusterings as a bipartite graph, they have designed a split-merge framework that can be used to obtain similarity measures to compare clusterings on different data sets. The problem addressed in this article is different from the one considered by Xiang et al. (Xiang et al., 2012). Namely, we concern with the development of split-merge framework that can be used to adjust the existing clustering solution to newly arrived data. Our framework also models two clusterings (the existing and the newly constructed one) as a bipartite graph which is decomposed into connected components (bi-cliques) (see Fig. 1 (a), (b) and (c)). Each component is further analysed and if it is necessary it is decomposed into subcomponents (see Fig. 1 (c) and (d)). The subcomponents are then taken into consideration in producing the final clustering solution. For example, if an existing cluster is *overclustered* (Fig. 1 (b)), 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* (Fig. 1 (a)), they are merged with that cluster.

Let us formally describe the proposed Split-Merge Evolutionary Clustering algorithm. The input bipartite graph is $G = (C, C', E)$, where C and C' are sets of clusters of left and right nodes and E is a subset of $C \times C'$ that represents correlations between the nodes of two sets. The three main steps of the algorithm are as follows:

1. Initially, all unreachable nodes from either side of G are found. These are singleton clusters (outliers) in our final clustering solution. We remove these nodes from the graph.
2. At the second step, all bi-cliques of G are found and considered. If a bi-clique connects a node from the left side (C) of G with several nodes from C' the elements of this node have to be split among the corresponding nodes from C' (see Fig. 1 (b)). In the opposite case, i.e., when we have a bi-clique that connects a node from the right side (C') of G with several nodes from left those nodes have to be merged with that node (cluster) (see Fig. 1 (a)). All clustered nodes are removed from the graph.
3. At the final 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 (see Fig. 1 (c) and (d)). For example, $c_i \in C_i$ will be correlated with all pairs $(c_j, c'_k) \in E_i$ such that $c_i \equiv c_j$, and $c'_i \in C'_i$ will be correlated with all pairs $(c_j, c'_k) \in E_i$ such that $c'_i \equiv c'_k$. 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}. For example, in Fig. 1 (d), cluster C_2 will first be split among clusters C'_1, C'_2 and C'_3, i.e. three new clusters, denoted by (C_2, C'_1), (C_2, C'_2) and (C_2, C'_3), will be obtained. Then at the third step of the algorithm clusters (C_2, C'_1) and (C_3, C'_1) will be merged together.

The pseudocode of the proposed *Split-Merge Evolutionary Clustering* algorithm is given in Algorithm 1. In addition, the algorithm is illustrated with an example in Fig. 2. The clustering solution generated by the Split-Merge Clustering is compared to one produced by the PivotBiCluster. It is interesting to notice that the two algorithms will produce very different clustering solutions on the same input graph. For example, the Split-Merge Clustering will generate a 4-cluster solution while one obtained by the PivotBiCluster will have only 2 clusters. The latter number is quite low taking into account the number of clusters in the two input clusterings. Moreover, as it was mentioned in the previous section the PivotBiCluster algorithm cannot produce a clustering solution in which existing clusters are split among new clusters.

4 CASE STUDIES

Luxburg et al. (von Luxburg et al., 2012) argue that clustering should not be treated as an application-independent mathematical problem, but should always be studied in the context of its end-use. Motivated by this study we have illustrated and initially evaluated the two studied clustering algorithms in two different case studies. We have compared the performance of the algorithms in expertise retrieval domain by applying them on data extracted from PubMed repository. In addition, a case study in profiling patients in healthcare domain has been conducted.

Algorithm 1 : Split-Merge Evolutionary Clustering Algorithm.

```
function SPLIT-MERGE($G = (C, C', E)$)
    for all nodes $c \in C \cup C'$ do (*First step*)
        if $c$ is an unreachable node then
            Turn $c$ into a singleton and remove it from $G$
        end if
    end for
    for all nodes $c \in C \cup C'$ do (*Second step*)
        Choose $c_1$ uniformly at random from $C$
        if $c_1$ is the only node from $C$ that takes part in a bi-clique connecting it with one or several nodes from $C'$ then
            Split $c_1$ among the corresponding nodes from $C'$
        end if
    end for
    for all nodes $c \in C \cup C'$ do
        Choose $c'_1$ uniformly at random from $C'$
        if $c'_1$ is the only node from $C'$ that takes part in a bi-clique connecting it with one or several nodes from $C$ then
            Merge $c'_1$ with the corresponding nodes from $C$
        end if
    end for
    for all nodes $c \in C$ do (*Third step*)
        Choose $c_1$ uniformly at random from $C$
        Split $c_1$ among its adjacent nodes from $C'$ and form new temporary clusters
    end for
    for all nodes $c' \in C'$ do
        Choose $c'_1$ uniformly at random from $C'$
        Merge $c'_1$ with its adjacent nodes from the built set of temporary clusters
        Remove the clustered nodes from $G$
    end for
    return all connected components (bi-cliques) as clusters of $X \cup X'$
end function
```

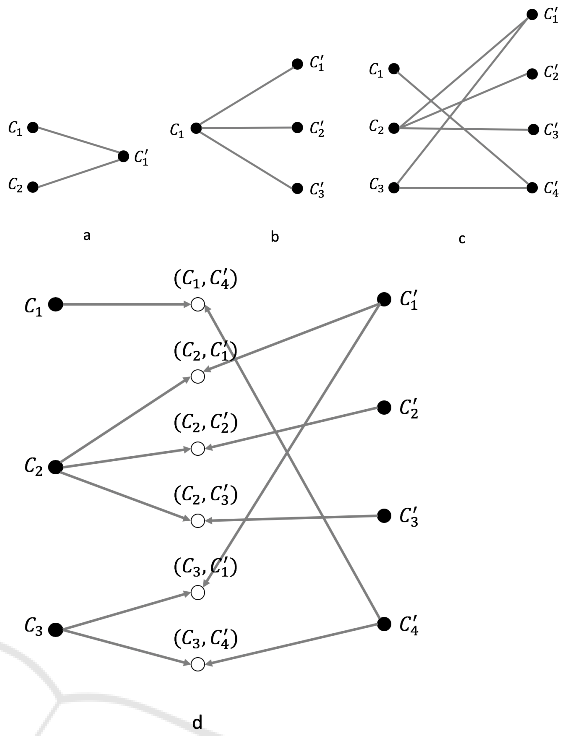

Figure 1: Split-Merge Framework: a) a bi-clique that contains underclustered nodes (C_1 and C_2 intersect C'_1); b) a bi-clique that contains an overclustered node (C_1 intersects C'_1, C'_2 and C'_3); c) a bi-clique that has to be decomposed into subcomponents d) a tripartite graph obtained by decomposing the bi-clique depicted in (c) into split (left) and merge (right) subcomponents.

4.1 Expertise Retrieval Domain

Currently, organizations search for new employees not only relying on their internal information sources, but they also use data available on the Internet to locate the required experts. Thus the need for services that enable finding experts grows especially with the expansion of virtual organizations. People are more often working together by forming task-specific teams across geographic boundaries. The formation and sustainability of such virtual organizations greatly depends on their ability to quickly trace those people who have the required expertise. In response to this, research on identifying experts from on-line data sources (Abramowicz et al., 2011), (Balog and de Rijke, 2007), (Bozzon et al., 2013), (Hristoskova et al., 2013), (Stankovic et al., 2011), (Singh et al., 2013), (Tsiporkova and Tourwé, 2011),(Boeva et al., 2016), (Boeva et al., 2018), (Lin et al., 2017) has been gradually gaining interest in the recent years.

4.1.1 Case Description

Let us suppose that an expertise recommender system for finding biomedical experts based on on-line data is under development. The system builds and maintains a big repository of biomedical experts by extracting the information about experts' peer-reviewed articles that are published and indexed in PubMed. The experts stored in such big data repositories are usually partitioned into a number of subject categories in order to facilitate the further search and identification of experts with the appropriate skills and knowledge. In addition, the system database is periodically updated by extracting new data. It is becoming impractical to re-cluster this large volume of available information. Therefore, the objective is to update the existing expert partitioning by the clustering produced on the newly extracted experts.

4.1.2 Data Sets

The data needed for our task is extracted from PubMed, which is one of the largest repositories of peer-reviewed biomedical articles published worldwide. Medical Subject Headings (MeSH) is a controlled vocabulary developed by the US National Library of Medicine for indexing research publications, articles and books. Using the MeSH terms associated

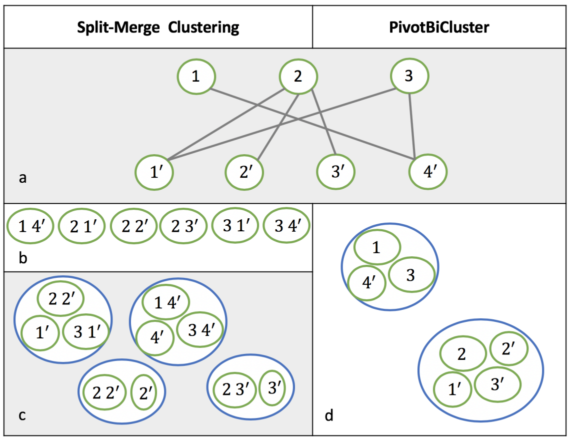

Figure 2: Clustering solutions generated by Split-Merge Clustering (left) and PivotBiCluster (right), respectively: a) the input bipartite graph; b) temporary clusters formed by Split-Merge Clustering after splitting overclustered nodes from the left (upper) set ($\{1,2,3\}$) of the graph among corresponding nodes from the right (below) set ($\{1',2',3',4'\}$); c) the final clustering solution produced by Split-Merge Clustering, d) the final clustering solution produced by PivotBiCluster.[1]

with peer-reviewed articles published by the above considered researchers and indexed in the PubMed, we extract such authors and construct their expert profiles. An expert profile is defined by a list of MeSH terms used in the PubMed articles of the author in question to describe her/his expertise areas.

We have extracted a set of 4343 authors from the PubMed repository. After resolving the problem with ambiguity[2] the set is reduced to one containing only 3753 different researchers. Then each author is also represented by a list of all different MeSH headings used to describe the major topics of her/his PubMed articles.

In addition to the above set of 3753 biomedical researchers we have used a set of 102 researchers who have taken part in a scientific conference devoted to integrative biology[3]. These researchers have been grouped into 8 clusters with respect to the conference sessions. They are considered as relevant experts, thus, used as the ground truth to benchmark the results of the studied clustering algorithms.

4.2 Patient Profiling in Healthcare Domain

The volumes of current patient data as well as their complexity make clinical decision making more challenging than ever for physicians and other care givers. Decision Support Systems (DSS) can be used to process data and form recommendations and/or predictions to assist such decision makers (Belle et al., 2013). Data mining techniques can be applied to identify pattern or rules about various quality problems. For example, profiling together patients who share similar clinical conditions can facilitate the diagnosis and initial treatment of individuals having similar illness predisposition.

The ability of machine learning and data mining tools to identify significant features from complex data sets detects their importance. A variety of such techniques have already been proposed in healthcare domain (Cheng et al., 2013), (Aishwarya and Anto, 2014), (Golino et al., 2014), (Menasalvas et al., 2018).

4.2.1 Case Description

Let us suppose a decision support system that can be used to study and associate the patient anthropometric measurements with the person increased risk for cardiovascular disease, e.g., hypertension, is under development. The core of the system is based on clustering techniques which provide groupings of profiles of individuals with similar anthropometric features, e.g., such as body mass index (BMI), waist (WC) and hip circumference (HC), and waist hip ratio (WHR). The classification of groups of patients who share properties in common might provide useful information for the diagnosis and initial management of risk for hypertension or other cardiovascular disease. For example, the patients who share the same profile should probably have similar predisposition and should be provided similar healthcare recommendations. In addition, the system must be able to update and improved the produced anthropometric categories by the clusters generated on newly arrived patients anthropometric measurements.

4.2.2 Data Sets

The dataset used in this case study is publicly available and published in (Golino et al., 2014). The data contains 400 undergraduate students aged between 16 and 63 years old, where a 56.3% are women. The following features describe the data: age, obesity, BMI, WC, HC, WHR, Systolic Blood Pressure (SBP), Diastolic Blood Pressure (DBP), *preh* for women and

[1] A cluster is represented by a circle or an ellipse. An ellipse with two cluster labels inside, e.g., $2\ 1'$, means that some elements from the first cluster (2) are added to the second cluster ($1'$).

[2] This problem refers to the fact that multiple profiles may represent one and the same person and therefore must be merged into a single generalized expert profile.

[3] Integrative Biology 2017: 5th International Conference on Integrative Biology (London, UK, June 19-21, 2017).

hyper for men, where the *preh* and *hyper* are classification labels that show what kind of blood pressure the individual has (e.g., regular or hyper). According to the results published in (Li et al., 2016) people can be grouped into six clusters depending on their blood pressure. Based on this the individuals in our test data set have been grouped into 6 clusters. This grouping is considered as the ground truth to benchmark the results generated by the two studied clustering algorithms.

5 EXPERIMENTS AND DISCUSSION

5.1 Metrics

The data mining literature provides a range of different cluster validation measures, which are broadly divided into two major categories: external and internal (Jain and Dubes, 1988). External validation measures have the benefit of providing an independent assessment of clustering quality, since they validate a clustering result by comparing it to a given external standard. However, an external standard is rarely available. Internal validation techniques, on the other hand, avoid the need for using such additional knowledge, but have the alternative problem to base their validation on the same information used to derive the clusters themselves.

In this work, we have implemented three different validation measures for estimating the quality of clusters, produced by the two studied clustering algorithms. Since we have a benchmark clustering of the set of 102 biomedical researchers, described in the foregoing section, we have used the F-measure as an external validation metric (Larsen and Aone, 1999). The F-measure is the harmonic mean of the precision and recall values for each cluster. For a perfect clustering the maximum value of the F-measure is 1. In addition, Silhouette Index (SI) has been applied as an internal measure to assess compactness and separation properties of the generated clustering solutions (Rousseeuw, 1987). The values of Silhouette Index vary from -1 to 1.

In addition to the above two metrics, we have used Jaccard index (Jaccard similarity coefficient) (Jaccard, 1912) to evaluate the stability of a clustering method. The Jaccard Index ranges from 0 to 1, where a higher value indicates a higher similarity between clustering solutions. Jaccard Index has been used to measure the similarity between the generated clustering solutions and the benchmark partitioning of the data in the second case study.

5.2 Implementation and Availability

We used the Entrez Programming Utilities (E-utilities) to download all the publications associated with the extracted biomedical researchers (Sayers, 2010). The E-utilities are the public API to the NBCI Entrez system and allow access to all Entrez databases including PubMed, PMC, Gene, Nuccore and Protein. For calculation of semantic similarities between MeSH headings, we use MeSHSim which is implemented in an R package. It also supports querying the hierarchy information of a MeSH heading and information of a given document including title, abstraction and MeSH headings (Zhou and Shui, 2015). The two studied clustering algorithms (Split-Merge Clustering and PivotBiCluster) are implemented in Python. The cluster validation measures (see Section 5.1) used to validate the clustering solutions generated in our experiments are implemented in scikit-learn library. Scikit-learn is a Python library for data mining and data analysis. Supplementary information is available at GitLab[4].

5.3 Case Study 1

Initially, we use the first built data set that contains 3753 PubMed expert profiles of biomedical researchers. Each expert profile is a vector of subject keywords describing the expert's competence. The researchers of this set are randomly separated in two sets. The one set contains 2407 experts grouped into 122 clusters by using *k*-means and the other one has 1346 experts separated into 112 clusters again by applying *k*-means. The number of clusters is determined by clustering each set applying *k*-means for different *k* and evaluating the obtained solutions by SI. The two clustering algorithms are then executed twice to integrate the clustering solutions of these two data sets. The clustering solution produced by the PivotBiCluster has 95 clusters while the proposed Split-Merge Clustering algorithm has generated a solution with 104 clusters. The generated clustering solutions are evaluated by SI and the average scores are -0.158 (PivotBiCluster) and 0.058 (Split-Merge Clustering), respectively. Evidently, the Split-Merge Clustering algorithm outperforms PivotBiCluster on this data set. We believe this is due to the fact that it adjusts better to data by being able not only to merge those clusters

[4] https://gitlab.com/machine_learning_vm/clustering_techniques

that are underclustered but also to split those that are overclustered.

Next, the benchmark set of 102 different expert profiles is used to generate 10 test data sets couples. Each test couple separates the researchers randomly in two sets. The one set (containing 70 experts) of each couple presents the available set of experts, and another one (32 experts) is the set of newly extracted experts. In that way, 10 test clustering couples are created.

We have studied two different experiment scenarios. In the first scenario the experts in each test set are grouped into clusters of experts with similar expertise based on the conference session information, i.e. each set is partitioned into 8 clusters. In the second scenario for each data sets the optimal number of clusters is determined by clustering the set applying k-means for different k and evaluating the obtained solutions by SI. In this way, two different experiments have been conducted on 10 test data set couples. In both experiments, the PivotBiCluster algorithm is executed 10 times (i.e., 100 executions in total for each experiment) to integrate the corresponding clusterings. In comparison to the PivotBiCluster, the Split-Merge Clustering is conducted only once on each test couple, since it does not start by a random cluster selection. Namely, it initially identifies those bi-cliques that have to be split and merged, respectively, i.e. the clustering result is not dependent on the algorithm initialization.

The obtained results for SI and F-measure are shown in Table 1 and Table 2, respectively. Observe that the PivotBiCluster outperforms the Split-Merge Clustering algorithm only in one case. Namely, it has generated a higher F-measure value than the Split-Merge Clustering algorithm in the first experiment. It is interesting to notice that in the second experiment (see Table 2) the SI scores are not only higher in comparison to the ones generated in the first experiment, but they are also positive. Evidently, using the optimal number of clusters significantly improves the quality of the generated clustering solutions with respect to compactness and separation properties. However, the corresponding F-measure scores are lower than the ones generated in the first experiment.

The above results support the mentioned above arguments of Luxburg et al. (von Luxburg et al., 2012) that the cluster evaluation methods can produce contradictory results and often do not serve their purpose. The main point of the authors is that clustering algorithms cannot be evaluated in a problem independent way, i.e. the known cluster validation measures cannot be used to evaluate the usefulness of the clustering. It is still not clear how we can measure the usefulness of a newly developed clustering algorithm. Certainly, the proposed Split-Merge Clustering algorithm needs further evaluation and validation in case studies from different application domains. Thus in the next section we present an additional evaluation of the two studied algorithms in a case study from healthcare domain.

Table 1: Experiment 1: Average F-measure and SI values generated on the clustering solutions of the 10 test data set couples.

	Experiment 1	
Metrics	***PivotBiCluster***	***Split-Merge Clust.***
F-measure	0.618	0.582
SI	-0.145	-0.129

Table 2: Experiment 2: Average F-measure and SI values generated on the clustering solutions of the 10 test data set couples.

	Experiment 2	
Metrics	***PivotBiCluster***	***Split-Merge Clust.***
F-measure	0.321	0.331
SI	0.137	0.157

5.4 Case Study 2

In this case study, we have used the data set explained in Section 4.2.2. This set consists of 400 individual profiles and it is used to generate 10 test data set couples by randomly separating the individuals in two sets. The one set (280 patients) of each couple presents the available set of individual profiles, and another one (120 individuals) is the set of newly collected patients' profiles. In that way similar to the first case study we have created 10 test data set couples. Notice that each patient profile is a vector of the patient's anthropometric features (BMI, WC, HC, WHR), and the patient's Systolic Blood Pressure (SBP) and Diastolic Blood Pressure (DBP). The patients' profiles of each set have been grouped in 6 clusters according to (Li et al., 2016), see Section 4.2.2. Namely, the individuals have been grouped in six clusters depending on their blood pressure. The obtained clusters are presented by their centroids.

Analogously to the first case study, the PivotBiCluster has been executed ten times for each test data set couple (i.e., 100 executions in total). The algorithm considers clusters in random order and generates a different clustering solution for each execution. As a result, the average value over these ten executions has been calculated. The above randomness is not presented in the Split-Merge Clustering algorithm. Therefore, it has been executed only once over each test data set couple. Both algorithms are explained in more detail in Section 3.

The generated clustering solutions are again evaluated by SI and F-measure. The obtained average SI scores are -0.013 (PivotBiCluster) and -0.170 (Split-Merge Clustering), respectively. Evidently, the PivotBiCluster outperforms the Split-Merge Clustering algorithm with respect to this evaluation criteria. The results obtained by F-measure also support the better performance of PivotBiCluster (0.71 against 0.46 for Split-Merge Clustering) on this data set. However, it is interesting to observe that the number of clusters of the clustering solutions generated by the PivotBiCluster on the test data set couples varies from 1 to 5 while in the case of the Split-Merge Clustering the individuals are grouped into 5 or 6 clusters. Notice that the benchmark clustering (Section 4.2.2) has 6 clusters.

The above results have motivated us to use Jaccard Index for an additional comparison of the two studied algorithms. Namely, we have applied the Jaccard Index to measure the similarity between the generated clustering solutions and the benchmark clustering. The corresponding values are 0.081 (PivotBiCluster) and 0.291 (Split-Merge Clustering), i.e., the Split-Merge Clustering algorithm has generated a higher average Jaccard score than the PivotBiCluster.

In addition, we have evaluated the two clustering algorithms with respect to the purity of the generated clustering solutions. For this purpose we consider how the two main classes (regular and hyper blood pressure) are distributed among the clusters. The score obtained for the benchmark clustering is 0.16. The values generated for the two studied algorithms are 0.025 (PivotBiCluster) and 0.17 (Split-Merge Clustering), respectively. Evidently, the Split-Merge Clustering performs better than PivotBiCluster with respect to this criteria and manages to preserve a level of purity closed to one of the benchmark clustering.

6 CONCLUSION AND FUTURE WORK

In this work, we have proposed a novel evolutionary clustering technique, entitled Split-Merge Evolutionary Clustering, that can be used to adapt the existing clustering solution to a clustering of newly collected data elements. The proposed technique has been compared to PivotBiCluster, an existing clustering algorithm that is also suitable for concept drift scenarios. The two algorithms have been evaluated and demonstrated in two different case studies. The Split-Merge Clustering algorithm has shown better performance than the PivotBiCluster in most of the studied experimental scenarios.

For future work, we aim to pursue further comparison and evaluation of the two clustering algorithms in different application domains and on richer data sets.

REFERENCES

Abramowicz, W., Bukowska, E., Dzikowski, J., Filipowska, A., and Kaczmarek, M. (2011). Semantically enabled experts finding system - ontologies, reasoning approach and web interface design. In *ADBIS 2011, Research Communications, Proc. II of the 15th East-European Conference on Advances in Databases and Information Systems, September 20-23, Vienna, Austria*, pages 157–166.

Ackerman, M. and Dasgupta, S. (2014). Incremental clustering: The case for extra clusters. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'14, pages 307–315.

Ailon, N., Avigdor-Elgrabli, N., Liberty, E., and van Zuylen, A. (2011). Improved approximation algorithms for bipartite correlation clustering. In *Algorithms - ESA 2011 - 19th Annual European Symposium, Saarbrücken, Germany, September 5-9, 2011. Proceedings*, pages 25–36.

Aishwarya, A. and Anto, S. (2014). A medical decision support system based on genetic algorithm and least square support vector machine for diabetes disease diagnosis. *International Journal of Engineering Sciences & Research Technology*, 3(4):4042–4046.

Awasthi, P., Balcan, M. F., and Voevodski, K. (2017). Local algorithms for interactive clustering. *Journal of Machine Learning Research*, 18(3):1–35.

Balcan, M.-F., Blum, A., and Vempala, S. (2008). A discriminative framework for clustering via similarity functions. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC '08, pages 671–680.

Balog, K. and de Rijke, M. (2007). Finding similar experts. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 821–822.

Bansal, N., Blum, A., and Chawla, S. (2004). Correlation clustering. *Machine Learning*, 56(1-3):89–113.

Belle, A., Kon, M. A., and Najarian, K. (2013). Biomedical informatics for computer-aided decision support systems: A survey. *The Scientific World Journal*, pages 1–8.

Bickel, S. and Scheffer, T. (2004). Multi-view clustering. In *Proceedings of the Fourth IEEE International Conference on Data Mining*, ICDM '04, pages 19–26.

Boeva, V., Angelova, M., Boneva, L., and Tsiporkova, E. (2016). Identifying a group of subject experts using formal concept analysis. In *8th IEEE International Conference on Intelligent Systems, IS 2016, Sofia, Bulgaria, September 4-6*, IS IEEE, pages 464–469.

Boeva, V., Angelova, M., Lavesson, N., Rosander, O., and Tsiporkova, E. (2018). Evolutionary clustering techniques for expertise mining scenarios. In *Proceedings of the 10th International Conference on Agents and Artificial Intelligence, ICAART, Volume 2, Funchal, Madeira, Portugal, January 16-18*, pages 523–530.

Boeva, V., Tsiporkova, E., and Kostadinova, E. (2014). *Analysis of Multiple DNA Microarray Datasets*, pages 223–234. Springer Berlin Heidelberg.

Bozzon, A., Brambilla, M., Ceri, S., Silvestri, M., and Vesci, G. (2013). Choosing the right crowd: Expert finding in social networks. In *Proceedings of the 16th International Conference on Extending Database Technology*, EDBT '13, pages 637–648. ACM.

Charikar, M., Chekuri, C., Feder, T., and Motwani, R. (1997). Incremental clustering and dynamic information retrieval. In *Proc. of the 29th Annual ACM Symposium on Theory of Computing*, STOC '97, pages 626–635.

Cheng, C. W., Chanani, N., Venugopalan, J., Maher, K., and Wang, M. D. (2013). An icu clinical decision support system using association rule mining. *Translational Engineering in Health and Medicine, IEEE*, 1(2):8–17.

Fa, R. and Nandi, A. K. (2012). Smart: Novel self splitting-merging clustering algorithm. In *European Signal Processing Conference, Bucharest, Romania, August, 27-32*. IEEE.

Gionis, A., Mannila, H., and Tsaparas, P. (2007). Clustering aggregation. *ACM Transaction of Knowledge Discovery Data*, 1(1).

Goder, A. and Filkov, V. (2008). Consensus clustering algorithms: Comparison and refinement. In *ALENEX*, pages 109–234.

Golino, H. F., de Brito Amaral, L. S., Duarte, S. F. P., and et al. (2014). Predicting increased blood pressure using machine learning. *Journal of Obesity*, 2014.

Hristoskova, A., Tsiporkova, E., Tourwé, T., Buelens, S., Putman, M., and Turck, F. D. (2013). A graph-based disambiguation approach for construction of an expert repository from public online sources. In *ICAART 2013 - Proceedings of the 5th International Conference on Agents and Artificial Intelligence, Volume 2, Barcelona, Spain, 15-18 February*, pages 24–33.

Jaccard, P. (1912). The distribution of flora in the alpine zone. *New Phytologist*, 11:37–50.

Jain, K. A. and Dubes, C. R. (1988). *Algorithms for Clustering Data*. Prentice-Hall, Inc.

Larsen, B. and Aone, C. (1999). Fast and effective text mining using linear-time document clustering. In *Proc. of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD'99, pages 16–22. ACM.

Li, Y., Feng, X., Zhang, M., Zhou, M., Wang, N., and Wangb, L. (2016). Clustering of cardiovascular behavioral risk factors and blood pressure among people diagnosed with hypertension: a nationally representative survey in china. *Sci Rep.*, 6:27627.

Lin, S., Hong, W., Wang, D., and Li, T. (2017). A survey on expert finding techniques. *Journal of Intelligent Information Systems*, 49:255–279.

Lughofer, E. (2012). A dynamic split-and-merge approach for evolving cluster models. *Evolving Systems*, 3:135–151.

Menasalvas, E. R., Tuñas, M. J., Bermejo, G., Gonzalo, C. M., Rodríguez-González, A., Zanin, M., Pedro, C. G. D., Méndez, M., Zaretskaia, O., Rey, J., Parejo, C., Bermudez, L. J. C., and Provencio, M. (2018). Profiling lung cancer patients using electronic health records. *Journal of Medical Systems*, 42:1–10.

O'Callaghan, L., Mishra, N., Meyerson, A., Guha, S., and Motwani, R. (2001). Streaming-data algorithms for high-quality clustering. In *Proceedings of IEEE International Conference on Data Engineering*, pages 685–694.

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65.

Sayers, E. (2010). *A General Introduction to the E-utilities. In: Entrez Programming Utilities Help [Internet]. Bethesda (MD): National Center for Biotechnology Information (US).*

Singh, H. S., Singh, R., Malhotra, A., and Kaur, M. (2013). Developing a biomedical expert finding system using medical subject headings. In *Healthcare informatics research*, 19(4): 243–249.

Stankovic, M., Jovanovic, J., and Laublet, P. (2011). Linked data metrics for flexible expert search on the open web. In *Proceedings of the 8th Extended Semantic Web Conference on The Semantic Web: Research and Applications - Volume Part I*, ESWC'11, pages 108–123. Springer-Verlag.

Tsiporkova, E. and Tourwé, T. (2011). Tool support for technology scouting using online sources. In *Advances in Conceptual Modeling. Recent Developments and New Directions*, pages 371–376. Springer Berlin Heidelberg.

von Luxburg, U., Williamson, R. C., and Guyon, I. (2012). Clustering: Science or art? In *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, volume 27 of *Proceedings of Machine Learning Research*, pages 65–79.

Wang, M., Huang, V., and Bosneag, A.-M. C. (2018). A novel split-merge-evolve k clustering algorithm. In *IEEE 4th International Conference on Big Data Computing Service and Applications (BigDataService), Bamberg, Germany, March 26-29*.

Xiang, Q., Mao, Q., Chai, K. M. A., Chieu, H. L., Tsang, I. W., and Zhao, Z. (2012). A split-merge framework for comparing clusterings. In *ICML*, pages 1055-1062.

Zhou, J. and Shui, Y. (2015). *MeSHSim: MeSH(Medical Subject Headings) Semantic Similarity Measures*. R package version 1.4.0.