Cluster Ensembles for Big Data Mining Problems

Milton Pividori, Georgina Stegmayer, and Diego Milone

sinc(i) - Research Institute for Signals, Systems and Computational Intelligence
Universidad Nacional del Litoral - Facultad de Ingeniería y Ciencias Hídricas

Mining big data involves several problems and new challenges [8], in addition to the huge volume of information. One the one hand, these data generally come from autonomous and decentralized sources, thus its dimensionality is heterogeneous and diverse, and generally involves privacy issues. On the other hand, algorithms for mining data such as clustering methods, have particular characteristics that make them useful for different types of data mining problems. Due to the huge amount of information, the task of choosing a single clustering approach becomes even more difficult. For instance, k-means, a very popular algorithm, always assumes spherical clusters in data; hierarchical approaches can be used when there is interest in finding this type of structure; expectation-maximization iteratively adjusts the parameters of a statistical model to fit the observed data. Moreover, all these methods work properly only with relatively small data sets. Large-volume data often make their application unfeasible, not to mention if data come from autonomous sources that are constantly growing and evolving.

In the last years, a new clustering approach has emerged, called *consensus clustering* or *cluster ensembles* [2, 6]. Instead of running a single algorithm, this approach produces, at first, a set of data partitions (*ensemble*) by employing different clustering techniques on the same original data set. Then, this ensemble is processed by a *consensus function*, which produces a single *consensus partition* that outperforms individual solutions in the input ensemble. This approach has been successfully employed for distributed data mining [6], what makes it very interesting and applicable in the big data context. Although many techniques have been proposed for large data sets [1, 7], most of them mainly focus on making individual components more efficient, instead of improving the whole consensus approach for the case of big data.

In this work, we propose *big cluster ensembles*, a consensus clustering scheme for big data scenarios, which is depicted in Figure 1. At the bottom, a huge amount of data sources is shown, where heterogeneous and complex features are available. These sources can be very dynamic, they can even become obsolete and new ones might be available to explore. In this scenario, the end user is interested in discovering which are the different hidden data structures in all that enormous amount of data. Due to its constant evolution and growth, final consensus solutions (shown at the top of the figure) evolve as well, thus reflecting the nature of data. The first component of this scheme are *data samplers*, denoted by θ_i. They read from data sources and sample objects and their features. In this way, data size is reduced not only by taking smaller and tractable object samples, but also by sampling their features, or even by employing a di-

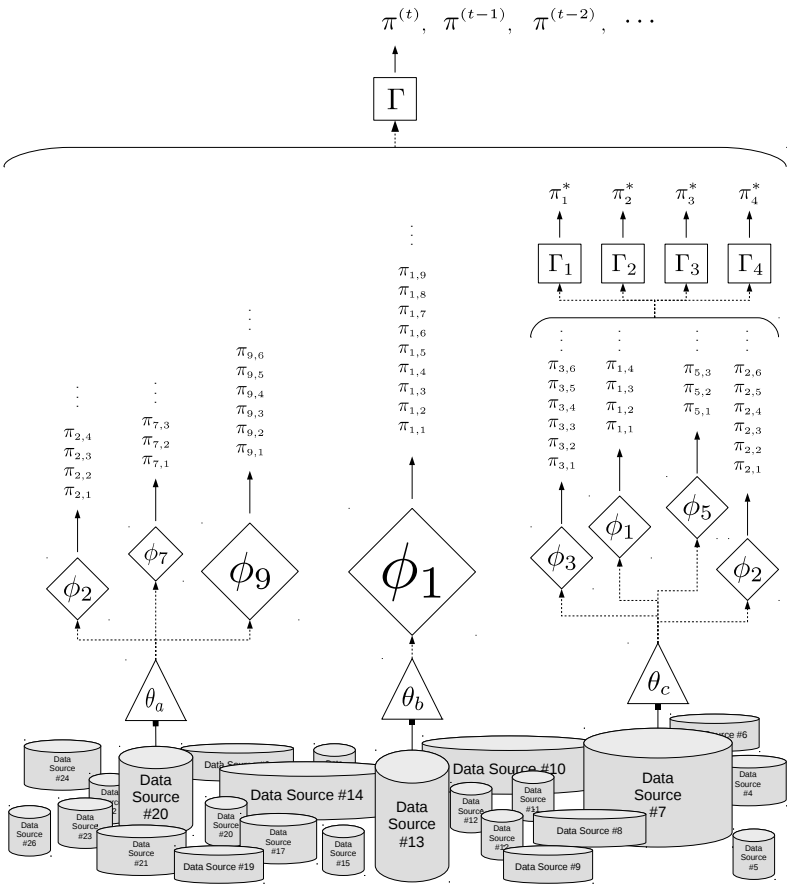

Fig. 1. Big cluster ensembles scheme.

mensionality reduction method. Once a sample is available, *clusterers* ϕ_i obtain a data partition $\pi_{i,j}$ from it. A clusterer refers to a clustering algorithm with a fixed set of parameters (for example, a k-means instance with $k = 7$, or a self-organizing map with size 30x30). They can be run several times by randomly varying their initial state, thus producing many data partitions and increasing diversity among ensemble members [4]. The configuration of a clusterer can be adjusted to fit the computational resources available in the node/instance where it will be run, what favors an horizontal scaling model [5]. This characteristic is shown in the figure with different sizes for each ϕ_i.

The set of partitions produced by clusterers is called ensemble, and it is the input of the consensus function (denoted in the figure as Γ). In this proposal, the consensus function is employed at two stages. It is used at the middle of the whole process, where *representative partitions* π_i^* are obtained: as ensembles produced by clusterers could become extremely large and redundant, these high-quality intermediate partitions summarize base solutions produced by clusterers.

The second stage where consensus is obtained is at the end of the process, where it derives a final consensus partition from the input ensemble, which unveils the underlying structure of data (top of the figure).

The scheme proposed here has several advantages. Note that full access to data is only required by clusterers, whereas consensus functions only need partitions, that is to say, they only need to know which cluster a data object belongs to. This, in addition to naturally allow distributed computing, also solves privacy issues, as clusterers can be run privately, and only partition labels need to be shared to the consensus function. Besides, heterogeneity does not need to be tackled locally by each clusterer, thus avoiding the traditional merging issues that arise when a single standard clustering algorithm is used [3]. Finally, in contrast to state-of-the-art techniques that try to optimize the clusterers or the consensus functions, our proposal improves the architectural aspect as a whole, not each individual component. This particular characteristic broaden the available choices at each stage, what improves flexibility. For example, our proposal allows us to employ any consensus function, not only a specialized one as proposed in other works [1].

Nowadays, there are many fields where big data has arrived, like biology (complete genome), agriculture (high-precision systems), astronomy, health and cybersecurity (intrusion detection). In this work, the big cluster ensemble approach was presented, which can be employed in a big data context to understand the hidden structure of large volumes of data. Many studies have demonstrated that cluster ensembles provide better performance than traditional methods, and the approach presented here offers more flexibility for data with high volume, velocity and variety.

1. Hore, P., Hall, L.O., Goldgof, D.B.: A scalable framework for cluster ensembles. Pattern Recognition 42(5), 676–688 (2009)
2. Iam-On, N., Boongoen, T., Garrett, S., Price, C.: A link-based approach to the cluster ensemble problem. Pattern Analysis and Machine Intelligence, IEEE Transactions on 33(12), 2396–2409 (2011)
3. Pividori, M., Stegmayer, G., Carrari, F., Milone, D.H.: Consensus clustering from heterogeneous measures of s. lycopersicum. In: 4to. Congreso Argentino de Bioinformática y Biología Computacional (4CAB2C) (2013), http://fich.unl.edu.ar/sinc/sinc-publications/2013/PSCM13, in press.
4. Pividori, M., Stegmayer, G., Milone, D.H.: A method to improve the analysis of cluster ensembles. Revista Iberoamericana de Inteligencia Artificial 17(53), 46–56 (2014), http://fich.unl.edu.ar/sinc/sinc-publications/2014/PSM14, iSSN: 1137-3601
5. Singh, D., Reddy, C.K.: A survey on platforms for big data analytics. Journal of Big Data 2(8) (2014)
6. Strehl, A., Ghosh, J., Cardie, C.: Cluster ensembles - a knowledge reuse framework for combining multiple partitions. Journal of Machine Learning Research 3, 583–617 (2002)
7. Traganitis, P., Slavakis, K., Giannakis, G.: Clustering high-dimensional data via random sampling and consensus. In: Signal and Information Processing (GlobalSIP), 2014 IEEE Global Conference on. pp. 307–311 (2014)
8. Wu, X., Zhu, X., Wu, G.Q., Ding, W.: Data mining with big data. Knowledge and Data Engineering, IEEE Transactions on 26(1), 97–107 (2014)