ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

Federated Learning approach for Spectral Clustering

Elena Hernández-Pereira and Oscar Fontenla-Romero and
Bertha Guijarro-Berdiñas and Beatriz Pérez-Sánchez*

CITIC and Facultad de Informática, Universidade da Coruña
Campus de Elviña, A Coruña, Spain

Abstract. Spectral clustering is a clustering paradigm that has been shown to be more effective in finding clusters with non-convex shapes than some traditional algorithms such as k-means. However, this algorithm is not directly applicable when the data is naturally distributed in different locations, as it happens in many Internet of Things scenarios. In this work, we propose a distributed spectral clustering to create a cooperative federated model to deal with those cases in which the data is distributed in different sites and with data privacy concerns. We demonstrate that sharing a minimal amount of information allows this distributed version of the spectral clustering to achieve good behavior for clustering several synthetic data sets.

1 Introduction

Clustering is one of the most widely used techniques for exploratory data analysis with wide ranging applications. Clustering algorithms identify groups of objects (clusters) taking into account some similarities inherent among them. Different types of clustering methods have been proposed in the literature and they are classified mainly in three categories: hierarchical, density and partitioning techniques. Hierarchical clustering arranges objects in a hierarchy with a treelike structure based on the distance or similarity between them. Clusters are formed by iteratively dividing the patterns using a top-down (agglomerative) or a bottom up (divisive) approach. Density methods try to identify clusters as dense regions in the data space, separated by regions of lower point density. Finally, in the case of partition clustering methods, data are assigned into a concrete number of clusters, without any hierarchical structure, thanks to the optimization of some criterion function. K-means is one of the simplest and most used partition techniques [1].

Among all the methods, Spectral Clustering (SC) [2, 3] has become one of the most popular techniques. This method constructs a similarity graph from the raw data and then tries to find groups of data connected by high-weight edges (clusters) separated by low-weight edges from the other groups on the

*This work has been supported by grant *Machine Learning on the Edge (Ayudas Fundación BBVA a Equipos de Investigación Científica 2019)*, also by the National Plan for Scientific and Technical R&I of the Spanish Government (Grant PID2019-109238GB-C2), and by the Xunta de Galicia (Grant ED431C 2018/34) with the European Union ERDF funds. CITIC is partially funded by "Consellería de Cultura, Educación e Universidades from Xunta de Galicia" (Grant ED431G 2019/01).

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

graph. This technique often outperforms traditional clustering algorithms since it does not make strong assumptions on the form of the clusters. However, spectral clustering is not suitable to deal with large data sets due to the high computational cost and memory usage needed to calculate the similarity graph between all the data points [4]. Few proposals have been presented to alleviate this problem, the most prominent is the one by Chen et al. [5], which presents a parallel calculation method based on the use of a sparse similarity matrix. This is a very beneficial proposal to scale up the clustering process using several computing nodes, but it assumes that any data can be accessed from any node in the parallel system. This can be a problem if, for data privacy reasons, it is not desirable to share data between nodes.

In this work, we propose a Distributed Spectral Clustering (DSC) to create a cooperative federated model to deal with those cases in which the data is naturally distributed in different locations (devices), as it usually happens in many scenarios, for example, of Internet of Things (IoT). This approach allows privacy issues to be handled with a minimum amount of private data to be released. This can be also used to speed up the computation of a centralized spectral clustering by introducing execution parallelism among multiple cores.

2 Proposed method

Let's assume that the data set is distributed among n nodes or devices, as it could happen in the IoT scenario shown in Figure 1, where the spectral clustering should be performed. The goal of this work is to develop a learning algorithm that is able to obtain a single collaborative clustering model in this distributed scenario without having to send all the local data from the nodes to a centralized location. This would significantly reduce the network traffic and the need to share, potentially private. Besides, as mentioned, applying a centralized SC method on a large amount of data might not be feasible.

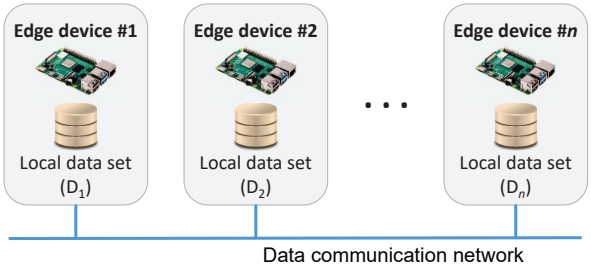

Fig. 1: Network of IoT devices containing local data sets

Spectral clustering method transforms the clustering problem into the spectral decomposition of Laplacian matrix (similarity matrix). First, it constructs this matrix from the data set, obtains the first k-eigenvectors to create a new data feature space, and then use k-means to cluster data in the eigenvector space.

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

The clustering results are mapped back to the original space. To perform a distributed spectral clustering the locally obtained centroids for the eigenvectors of each node would be shared, to later perform a global grouping using the local centroids. However, this has the problem that there is an intrinsic sign indeterminacy in the matrix decomposition obtained by singular value decomposition (SVD) or eigenvalue decomposition (EVD). Decompositions are unique, except that two decompositions can be obtained so that one is a reflection of the other. Although this has no relevance from the mathematical point of view, from a practical point of view it adds an indeterminacy in the signs calculated in the eigenvectors of the local spectral clustering [6]. This makes this naive distributed approach not appropriate because data from different nodes that should be in the same cluster can have eigenvectors with opposite signs.

In order to provide a better distributed approach, our proposed method consists of the following phases:

1. First, an arbitrarily chosen node, performs a spectral clustering using only its local data. As a result, from the calculated eigenvectors (V) the centroids are obtained by means of a k-means model. Later, this node selects one data at random, and sends it and its eigenvector, to the rest of the nodes.

2. The other nodes receive this information and perform the eigenvalue decomposition of their local data but adding also the data point sent by the arbitrary node. Once the eigenvectors have been calculated, it is checked if for this data point the sign of each component of its local eigenvector coincides with those of the eigenvector also sent by the arbitrary node. If in some component the sign is opposite, then the sign of that component is changed for all the local eigenvectors of that node. The objective of this process is to avoid the issue of the ambiguity of the sign.

3. After making the necessary sign changes, the eigenvectors are grouped at each node using k-means.

4. Finally, in order to obtain a set of global centroids, all nodes will send their local centroids to the arbitrary node to perform a final grouping using k-means again. The global centroids define a model similar to the one that would have been learned centrally from the global data set.

5. The final global centroids are shared with all nodes so that they can perform the local classification of each new data point that reaches them.

This process is described in detail in Algorithm 1.

3 Experimental results

To check the performance of the proposed Distributed Spectral Clustering (DSC) algorithm, its results were compared with those of the centralized SC and the naive DSC (DSC without the sign swapping stage).

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

Algorithm 1: Distributed spectral clustering

input : Data points split in n nodes: $\{D_1, D_2, \ldots, D_n\}$ and the number of clusters (k)
output: Centroids of the distributed spectral clustering

Select an arbitrary node to start the process, let's assume node 1 for simplicity.
In node 1, compute $V_1, centroids_1 = SpectralClustering(D_1)$
From node 1 send to the other nodes a randomly selected data point $\mathbf{d} \in D_1$ and its eigenvector $\mathbf{v}_1^d \in V_1$
for $i \leftarrow 2$ **to** n **do**
  $D_i = D_i \cup \mathbf{d}$
  Compute $V_i = eigenvectors(D_i)$
  Get from V_i the eigenvector ($\mathbf{v}_i^d$) associated to $\mathbf{d}$
  for $c \leftarrow 1$ **to** k **do**
    if $sign(\mathbf{v}_i^d[c]) \neq sign(\mathbf{v}_1^d[c])$ **then**
      $sign\_flip(\mathbf{v}[c]),\ \forall \mathbf{v} \in V_i$
    end
  end
  $centroids_i = k - means(V_i \backslash \{\mathbf{v}_i^d\}, k)$
  Send $centroids_i$ to node 1
end
In node 1, compute $centroids = k - means(\{centroids_1, \ldots, centroids_n\}, k)$
Return $centroids$ as the global centroids

3.1 Data sets and metrics

To validate our proposal, several 2D synthetic data sets were used. These are shown in Figure 2 and their characteristics are presented in Table 1. The performance of the method was evaluated by comparing labels obtained by it with the true labels provided by ground truth. The metric used for the evaluation was clustering accuracy (Acc) that computes the percentage of total hits. Let T_i and L_i be, respectively, the ground truth labels and the labels obtained by the method. Then, accuracy is defined as [7]:

$$Acc(T, L) = \frac{\sum_{i=1}^{D} \delta(T_i, L_i)}{D}$$

where D is total number of data and $\delta(x, y)$ is a Kronecker function.

Data set	#samples per cluster	# clusters
$Moons_2$	5000	2
$Blobs$	5000	3
$Circles$	5000	3
$Moons_4$	4000	4

Table 1: Data sets characteristics

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

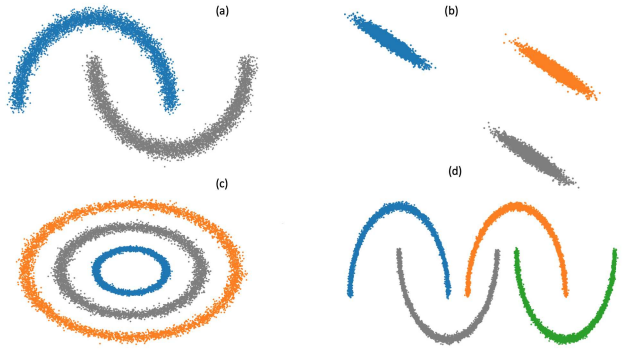

Fig. 2: Synthetic data sets used: (a) $Moons_2$, (b) $Blobs$, (c) $Circles$, (d) $Moons_4$

3.2 Discussion

Clustering experiments were performed on a CPU 3.2GHz with 32GB RAM using 10 cores, each corresponding to one processing node. To obtain more reliable results, each experiment was repeated 20 times. Average accuracy results are summarized in Table 2. The column labeled as *Naive DSC* shows the results obtained by the DSC version without the swapping stage. As expected, naive DSC does not provide good results as it missclassifies in all cases around half of the points. Conversely, the proposed method, using the swapping stage, obtain better results. As can be seen, in some cases ($Moons_2$ and $Blobs$) it carries out always the clustering task with no errors, achieving the same performance as the centralized method (100%). For the other data sets, $Circles$ and $Moons_4$, some data points were missclassified in some of the 20 experiments despite of the swapping stage.

Data set	Centralized SC	Naive DSC	DSC
$Moons_2$	100.0 ± 0.0	57.60 ± 7.57	100.0 ± 0.0
$Blobs$	100.0 ± 0.0	56.09 ± 8.41	100.0 ± 0.0
$Circles$	100.0 ± 0.0	50.32 ± 8.41	85.02 ± 9.75
$Moons_4$	100.0 ± 0.0	48.84 ± 6.32	92.02 ± 10.26

Table 2: Mean Acc ± std. dev. (%) for each data set

Table 3 compares the execution times used by the centralized and distributed version to process all data sets. As can be seen, DSC achieves a considerable reduction in execution time.

4 Conclusions

Spectral clustering has been shown to be more effective in finding clusters than some traditional algorithms such as k-means. However, it suffers from scalability

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

Data set	Centralized SC	DSC
$Moons_2$	3,399.85	84.05
$Blobs$	12,287.42	228.65
$Circles$	11,130.97	234.32
$Moons_4$	14,450.38	277.84

Table 3: Mean CPU time (s) needed to clustering each data set

problems in terms of memory use and computational times. In this work, a distributed approach is proposed to improve speedup with large data sets. Besides, in a federated learning environment, it is required that each node to perform the clustering task over its own data and to share only the results, thus maintaining privacy. However, this solution is damaged by the indeterminacy of the eigenvectors sign which causes wrong class assignment. To overcome this problem, we have proposed a solution in which from one node, only one data point and its eigenvector are shared, so the rest of the nodes can determine the correct sign of eigenvectors. With this approach, clustering results are drastically improved, coming very close to or even matching the centralized version of the algorithm, at the cost of a much lower execution time demand. All this makes DSC algorithm a very competitive solution when a centralized solution is not possible, either because data cannot be shared for privacy reasons, or because computing times are not acceptable when handling large volumes of data. As future work, we plan to design new strategies regarding the information that is shared with the reference node used by the rest of the nodes to update the knowledge learned locally, in order to achieve a better performance, as well as to carry carry out a more complete experimental study to evaluate the method on data sets with a greater number of variables and clusters.

References

[1] A. Vijayaraghavan, A. Dutta, and A. Wang. Clustering stable instances of euclidean k-means. In *NIPS*, 2017.

[2] W. E. Donath and A. J. Hoffman. Lower bounds for the partitioning of graphs. *IBM Journal of Research and Development*, 17(5):420–425, 1973.

[3] H. Jia, S. Ding, X Xu, and R. Nie. The lastest research progress on spectral clustering. *Neural Computing and Applications*, 24:1477–1486, 2014.

[4] C. Fowlkes, S. Belongie, F. Chung, and J. Malik. Spectral grouping using the Nyström method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):214–225, 2004.

[5] Wen-Yen Chen, Yangqiu Song, Hongjie Bai, Chih-Jen Lin, and Edward Y. Chang. Parallel spectral clustering in distributed systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(3):568–586, 2011.

[6] R. Bro, E. Acar, and Tamara G. Kolda. Resolving the sign ambiguity in the singular value decomposition. *Journal of Chemometrics*, 22(2):135–140, 2008.

[7] D. Cai, X. He, and J. Han. Document clustering using locality preserving indexing. *IEEE Transactions on Knowledge and Data Engineering*, 17(12):1624–1637, 2005.