Massively Distributed Clustering via Dirichlet Process Mixture

Khadidja Meguelati, Bénédicte Fontez, Nadine Hilgert, Florent Masseglia, Isabelle Sanchez

Massively Distributed Clustering via Dirichlet Process Mixture

Khadidja Meguelati[1],
Benedicte Fontez[2], Nadine Hilgert[2], Florent Masseglia[1], and Isabelle Sanchez[2]

[1] Inria, University of Montpellier, CNRS, LIRMM, France
firstname.lastname@inria.fr
[2] MISTEA, Univ. Montpellier, Montpellier SupAgro, INRAE, Montpellier, France
benedicte.fontez@supagro.fr, {nadine.hilgert, isabelle.sanchez}@inrae.fr

Abstract. Dirichlet Process Mixture (DPM) is a model used for multivariate clustering with the advantage of discovering the number of clusters automatically and offering favorable characteristics, but with prohibitive response times, which makes centralized DPM approaches inefficient. We propose a demonstration of two parallel clustering solutions : i) DC-DPM that gracefully scales to millions of data points while remaining DPM compliant, which is the challenge of distributing this process, ii) HD4C that addresses the curse of dimensionality by performing a distributed DPM clustering of high dimensional data such as time series or hyperspectral data.

Keywords: Gaussian random process · Dirichlet Process Mixture Model · Clustering · Parallelism · Reproducing Kernel Hilbert Space

1 Introduction

Clustering is the task of grouping similar data into the same cluster and separating dissimilar data in different clusters. It is a data mining technique intensively used for data analytics, with many applications. Clustering may be used for identification in the new challenge of digital agriculture or high throughput plant phenotyping, as illustrated in the demonstration. One of the main difficulties, for clustering, is the fact that we don't know, in advance, the number of clusters to be discovered. The Dirichlet Process Mixture (DPM) approach allows estimating that number and assigning observations to clusters, in the same process [1].

However, DPM is highly time consuming. Consequently, several attempts have been done to make it distributed but the resulting approaches usually suffer from convergence issues (imbalanced data distribution on computing nodes) [4] or do not fully benefit from DPM properties [3]. Furthermore, making DPM parallel is not straightforward since it must compare each record to the set of existing clusters, a highly repeated number of times. That impairs the global performances of the approach in parallel, since comparing all the records to all the clusters would call for a high number of communications and make the process impracticable.

In this demonstration, we show how our solutions combine the advantages of DPM (clustering quality) and of distributed computing (fast response time on large datasets).

The challenge is to build a consistent view of the clusters, despite a necessary split discovery mechanism. Obviously, such a mechanism need to guarantee low amounts of communications. This is one of the keys in distributed data science systems supporting algorithms often originally designed for centralized environments. Our solutions rely on sufficient statistics, by means of synchronization between machines in terms of proposed clusters at each step [1,2]. We provide an interactive demonstration of these previous results. This demonstration allows the user to check the results on various datasets and to evaluate the impact of parameter choices.

2 Distributed Clustering via Dirichlet Process Mixture

DC-DPM [1] presents a parallel clustering solution that takes advantage of the computing power of distributed systems by using parallel frameworks such as Spark.

The main novelty of DC-DPM is to propose a model and its estimation at the master level by exploiting the sufficient statistics from the workers, in a DPM compliant approach. The challenge of using sufficient statistics, in a distributed environment, is to remain in the DPM approach at all steps, including the synchronization between the worker and master nodes. Our approach approximates the DPM model even at the master level when local data is replaced by sufficient statistics between iterations.

In our work, the distributed DPM allows each node to have a view on the local results of all the other nodes, while avoiding exhaustive data exchanges. This is made by adding weights which represent proportions of observations from all clusters evaluated on the whole dataset. These weights are updated at the master level.

The workflow of DC-DPM is illustrated in Figure 1. It consists in identifying local clusters on the workers and synchronizing these clusters on the master. These clusters are then communicated as a basis among workers for local clustering consistency. By iterating this process we seek global consistency of DPM in a distributed environment and obtain DPM compliant results as shown in this demonstration and detailed in Section 5.

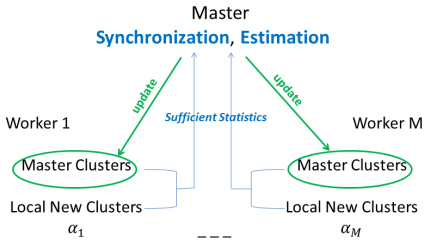

Fig. 1. Workflow of the DC-DPM approach

3 High Dimensional Data Distributed Dirichlet Clustering

HD4C [2] presents a parallel clustering approach adapted for high dimensional data. Actually, DC-DPM is a solution proposed to this issue when data is multivariate. This solution is based on a distributed DPM. In the case of high dimensional data or signals (infinite dimension), matrix computation is no more feasible (e.g., no inverse matrix, no matrix product). We need to replace a matrix product by an inner product in an adequate space of functions and to find the adequate measure. This inner product is mandatory to compute the likelihood and the posterior. To do that, HD4C [2] uses the properties of the Reproducible Kernel Hilbert Spaces (RKHS). We assume

that the random variable takes its values in a space of infinite dimension. Therefore, high dimensional data will be seen as trajectories of a random process. Our work focuses on Gaussian random process because of its ability to avoid simple parametric assumptions and integrate a lot of structure. For implementation, data are defined as an autocorrelated Gaussian process called Ornstein-Uhlenbeck

4 DC-DPM and HD4C in Spark

We implemented both DC-DPM and HD4C in Spark and made the code available for download and test at `https://github.com/khadidjaM`. To the best of our knowledge, this is the first time that a complete parallel solution for high dimensional data clustering is demonstrated, showing significant performance improvement over the state of the art and existing centralized solutions. Figure 2 illustrates the basic architecture of our solution, which is written in Scala. The main components of DC-DPM and HD4C are developed within Spark and deployed on top of Hadoop Distributed File System (HDFS) in order to efficiently read input data, as well as to store final results, and thus to overcome the bottleneck of centralized data storing.

The main feature of Spark is its distributed memory abstraction, called Resilient Distributed Datasets (RDD) and parallel operations used to handle it. RDD supports in-memory processing computation. As shown in figure 2, the intermediate results are stored in a distributed memory instead of disk storage and make the system faster.

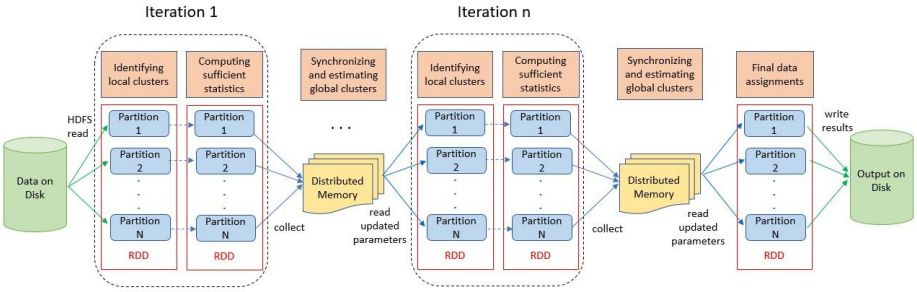

Fig. 2. Architecture of DC-DPM and HD4C in Spark

5 Demonstration

Parameters and Datasets. In this demonstration, we can choose the type of data according to the dimension: multivariate or high dimensional data. For both, the following parameters can be configured by the user: the dataset, the display level (individuals or cluster centers) and the number of displayed points.

Two examples from digital agriculture were used in this demonstration: *i)* the herd monitoring where animal activity is monitored using a collar-mounted accelerometer (high dimensional dataset), *ii)* clustering of image pixels to define plant labels as pre-treatment for plant phenotyping (multivariate dataset).

Synthetic data were generated using a two-steps principle. First, we generated cluster centers according to a multivariate normal distribution (multivariate data) or according to some polynomials (high dimensional data). Then, for each center, we generated data by using a multivariate normal distribution or a gaussian process.

The last dataset corresponds to more than 4K spectrum of 680 dimensions representing a protein rate measured on 10 different products.

Scenarios. The demonstration GUI is divided into three tabs. First, the "Method" tab page explains the clustering approaches proposed for each type of data: DC-DPM for multivariate data and HD4C for high dimensional data. Next, in the other tabs, the GUI enables the user to use drop-downs to set the parameters (see Figure 3 and 4). On the right, a plot of the clustering assignments for both the ground-truth and the chosen method: randomly selected individuals or all cluster centers are displayed. On the left, information about the clustering performance and scalability of each approach is given. The demo GUI is available at: `http://147.100.179.112:3838/team/kmeguelati/dpmclustering/`. The demonstration video is available at: `https://drive.google.com/file/d/1GHLF5csHk8Oa7PZK4dwA3RTK35KNIbne/view?usp=sharing`

Fig. 3. Users can vary the input datasets, display level and number of displayed data. The demo will report back information about the clustering performances and plot the clustering assignments for both the ground-truth and the chosen method.

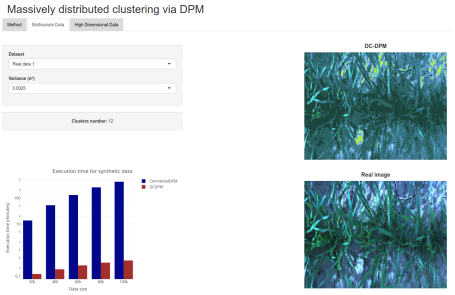

Fig. 4. The specific case of image pixel clustering. Users can vary the "variance" parameter σ^2 to control the number of clusters. Better view on: `http://147.100.179.112:3838/team/kmeguelati/dpmclustering/`.

References

1. Meguelati, K., Fontez, B., Hilgert, N., Masseglia, F.: Dirichlet process mixture models made scalable and effective by means of massive distribution. In: SAC: Symposium on Applied Computing. Limassol, Cyprus (Apr 2019). https://doi.org/10.1145/3297280.3297327
2. Meguelati, K., Fontez, B., Hilgert, N., Masseglia, F.: High Dimensional Data Clustering by means of Distributed Dirichlet Process Mixture Models. In: IEEE International Conference on Big Data (IEEE BigData). Los-Angeles, United States (Dec 2019)
3. Wang, R., Lin, D.: Scalable estimation of dirichlet process mixture models on distributed data. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence. pp. 4632–4639. IJCAI'17, AAAI Press (2017)
4. Williamson, S., Dubey, A., Xing, E.: Parallel markov chain monte carlo for nonparametric mixture models. In: International Conference on Machine Learning. pp. 98–106 (2013)