Advances in Intelligent Systems and Computing 950

Francisco Martínez Álvarez
Alicia Troncoso Lora
José António Sáez Muñoz
Héctor Quintián
Emilio Corchado *Editors*

14th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2019)

Seville, Spain, May 13–15, 2019, Proceedings

Springer

Advances in Intelligent Systems and Computing

Volume 950

The series “Advances in Intelligent Systems and Computing” contains publications on theory, applications, and design methods of Intelligent Systems and Intelligent Computing. Virtually all disciplines such as engineering, natural sciences, computer and information science, ICT, economics, business, e-commerce, environment, healthcare, life science are covered. The list of topics spans all the areas of modern intelligent systems and computing such as: computational intelligence, soft computing including neural networks, fuzzy systems, evolutionary computing and the fusion of these paradigms, social intelligence, ambient intelligence, computational neuroscience, artificial life, virtual worlds and society, cognitive science and systems, Perception and Vision, DNA and immune based systems, self-organizing and adaptive systems, e-Learning and teaching, human-centered and human-centric computing, recommender systems, intelligent control, robotics and mechatronics including human-machine teaming, knowledge-based paradigms, learning paradigms, machine ethics, intelligent data analysis, knowledge management, intelligent agents, intelligent decision making and support, intelligent network security, trust management, interactive entertainment, Web intelligence and multimedia.

The publications within “Advances in Intelligent Systems and Computing” are primarily proceedings of important conferences, symposia and congresses. They cover significant recent developments in the field, both of a foundational and applicable character. An important characteristic feature of the series is the short publication time and world-wide distribution. This permits a rapid and broad dissemination of research results.

**** Indexing: The books of this series are submitted to ISI Proceedings, EI-Compendex, DBLP, SCOPUS, Google Scholar and Springerlink ****

More information about this series at http://www.springer.com/series/11156

Francisco Martínez Álvarez ·
Alicia Troncoso Lora · José António Sáez Muñoz ·
Héctor Quintián · Emilio Corchado
Editors

14th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2019)

Seville, Spain, May 13–15, 2019, Proceedings

Springer

International Conference on
SOCO
Soft Computing Models in Industrial and Environmental Applications

Editors
Francisco Martínez Álvarez
Data Science and Big Data Lab
Pablo de Olavide University
Seville, Spain

Alicia Troncoso Lora
Data Science and Big Data Lab
Pablo de Olavide University
Seville, Spain

José António Sáez Muñoz
University of Salamanca
Salamanca, Spain

Héctor Quintián
Department of Industrial Engineering
University of A Coruña
A Coruña, Spain

Emilio Corchado
University of Salamanca
Salamanca, Spain

ISSN 2194-5357 ISSN 2194-5365 (electronic)
Advances in Intelligent Systems and Computing
ISBN 978-3-030-20054-1 ISBN 978-3-030-20055-8 (eBook)
https://doi.org/10.1007/978-3-030-20055-8

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

This volume of Advances in Intelligent and Soft Computing contains accepted papers presented at SOCO 2019 conference held in the beautiful and historic city of Seville (Spain), in May 2019.

Soft computing represents a collection or set of computational techniques in machine learning, computer science and some engineering disciplines, which investigate, simulate and analyze very complex issues and phenomena.

After a thorough peer review process, the 14th SOCO 2019 International Program Committee selected 57 papers which are published in these conference proceedings and represented an acceptance rate of 45%. In this relevant edition, a special emphasis was put on the organization of special sessions. Four special sessions were organized related to relevant topics as: Soft Computing Methods in Manufacturing and Management Systems; Soft Computing Applications in the Field of Industrial and Environmental Enterprises; Optimization, Modeling and Control by Soft Computing Techniques and Soft Computing in Aerospace, Mechanical and Civil Engineering: New methods and Industrial Applications.

The selection of papers was extremely rigorous in order to maintain the high quality of the conference, and we would like to thank the members of the Program Committees for their hard work in the reviewing process. This is a crucial process to the creation of a high-standard conference, and the SOCO conference would not exist without their help.

SOCO 2019 enjoyed outstanding keynote speeches by distinguished guest speakers: Prof. Dieu Tien Bui (University of South-Eastern Norway, Norway), Prof. Juan Manuel Corchado (University of Salamanca, Spain) and Prof. Julien Jacques (University of Lyon, France).

SOCO 2019 has teamed up with Neurocomputing (Elsevier) and Logic Journal of the IGPL (Oxford Academic) for a suite of special issues including selected papers from SOCO 2019. Furthermore, papers from a particular special session will be also considered for publication in special issues in Cybernetics and Systems: An International Journal (Taylor & Francis) and Expert Systems (Wiley).

Particular thanks go as well to the conference main sponsors, Startup Ole and IEEE SMC Spanish Chapter, who jointly contributed in an active and constructive manner to the success of this initiative.

We would like to thank all the special session organizers, contributing authors, as well as the members of the Program Committees and the Local Organizing Committee for their hard and highly valuable work. Their work has helped to contribute to the success of the SOCO 2019 event.

May 2019

Francisco Martínez Álvarez
Alicia Troncoso Lora
José António Sáez Muñoz
Héctor Quintián
Emilio Corchado

Organization

General Chairs

Francisco Martínez Álvarez	Pablo de Olavide University, Spain
Alicia Troncoso Lora	Pablo de Olavide University, Spain
Emilio Corchado	University of Salamanca, Spain

International Advisory Committee

Ashraf Saad	Armstrong Atlantic State University, USA
Amy Neustein	Linguistic Technology Systems, USA
Ajith Abraham	Machine Intelligence Research Labs (MIR Labs), Europe
Jon G. Hall	The Open University, UK
Paulo Novais	Universidade do Minho, Portugal
Amparo Alonso Betanzos (President)	Spanish Association for Artificial Intelligence (AEPIA), Spain
Michael Gabbay	King's College London, UK
Aditya Ghose	University of Wollongong, Australia
Saeid Nahavandi	Deakin University, Australia
Henri Pierreval	LIMOS UMR CNRS 6158 IFMA, France

Program Committee Chairs

Emilio Corchado	University of Salamanca, Spain
Francisco Martínez Álvarez	Pablo de Olavide University, Spain
Alicia Troncoso Lora	Pablo de Olavide University, Spain
Héctor Quintián	University of A Coruña, Spain

Program Committee

Albeto Herreros López	University of Valladolid, Spain
Alfredo Jimenez	KEDGE Business School, Spain
Andreea Vescan	Babes-Bolyai University, Romania
Angel Arroyo	University of Burgos, Spain
Anna Bartkowiak	University of Wroclaw, Poland
Anna Burduk	Wrocław University of Technology, Poland
Anton Koval	Zhytomyr State Technological University, Ukraine
Antonio Bahamonde	University of Oviedo, Spain
Camelia Chira	Babes-Bolyai University, Romania
Camelia Serban	Babes-Bolyai University, Romania
Camelia-M. Pintea	Technical University of Cluj-Napoca, North University Center at Baia Mare, Romania
Carlos Cambra	University of Burgos, Spain
Carlos Pereira	ISEC, Portugal
Carmen Benavides	University of León, Spain
Castejon Limas	University of León, Spain
Damian Krenczyk	Silesian University of Technology, Poland
Daniela Perdukova	Technical University of Kosice, Slovakia
David Alvarez Leon	University of León, Spain
David Griol	University Carlos III de Madrid, Spain
Dragan Simic	University of Novi Sad, Serbia
Eduardo Solteiro Pires	UTAD, Portugal
Eleni Mangina	University of A Coruña, Ireland
Eloy Irigoyen	University of the Basque Country, Spain
Enrique De La Cal Marín	University of Oviedo, Spain
Enrique Dominguez	University of Malaga, Spain
Enrique Onieva	University of Deusto, Spain
Esteban García-Cuesta	Universidad Europea de Madrid, Spain
Esteban Jove	University of A Coruña, Spain
Eva Volna	University of Ostrava, Czechia
Fernando Sanchez Lasheras	University of Oviedo, Spain
Florentino Fdez-Riverola	University of Vigo, Spain
Francisco Martínez Álvarez	Pablo de Olavide University, Spain
Georgios Ch. Sirakoulis	Democritus University of Thrace, Greece
Grzegorz Ćwikła	Silesian University of Technology, Poland
Héctor Quintián	University of A Coruña, Spain
Henri Pierreval	LIMOS-IFMA, France
Humberto Bustince	UPNA, Spain
Isaias Garcia	University of León, Spain
Iwona Pisz	Opole University, Poland
Jaime A. Rincon	Universitat Politècnica de València, Spain

Javier Sanchis Saez	Universitat Politècnica de València, Spain
Jesús D. Santos	University of Oviedo, Spain
Jiri Pospichal	University of Ss. Cyril and Methodius, Slovakia
Jorge García-Gutiérrez	University of Seville, Spain
Jose Gamez	University of Castilla-La Mancha, Spain
José Valente de Oliveira	Universidade do Algarve, Portugal
Jose Alfredo Ferreira Costa	Universidade Federal do Rio Grande do Norte, Brazil
José F. Torres	Pablo de Olavide University, Spain
Jose Luis Calvo-Rolle	University of A Coruña, Spain
José Luis Casteleiro-Roca	University of A Coruña, Spain
Jose M. Molina	University Carlos III de Madrid, Spain
Jose Manuel Gonzalez-Cava	University of La Laguna, Spain
Jose Manuel Lopez-Guede	University of the Basque Country, Spain
José Ramón Villar	University of Oviedo, Spain
Juan Gomez Romero	University of Granada, Spain
Juan Mendez	University of La Laguna, Spain
Julio César Puche Regaliza	University of Burgos, Spain
Krzysztof Kalinowski	Silesian University of Technology, Poland
Leocadio G. Casado	University of Almeria, Spain
Lidia Sánchez-González	University of León, Spain
Luis Magdalena	Universidad Politécnica de Madrid, Spain
Luis Alfonso Fernández Serantes	FH-Joanneum University of Applied Sciences, Spain
Luis Paulo Reis	University of Porto - FEUP/LIACC, Portugal
M. Chadli	University of Picardie Jules Verne, France
Maciej Grzenda	Warsaw University of Technology, Poland
Manuel Mejia-Lavalle	Cenidet, Mexico
Marcin Iwanowski	Warsaw University of Technology, Poland
Marcin Paprzycki	IBS PAN and WSM, Poland
Maria Tomas Rodriguez	The City University of London, UK
Maria Luisa Sanchez	University of Oviedo, Spain
Marius Balas	Aurel Vlaicu University of Arad, Romania
Matilde Santos	Universidad Complutense de Madrid, Spain
Mehmet Emin Aydin	University of the West of England, UK
Michal Wozniak	Wroclaw University of Technology, Poland
Mitiche Lahcene	University of Djelfa, Algeria
Oscar Castillo	Tijuana Institute of Technology, Mexico
Paul Eric Dossou	ICAM, France
Paulo Moura Oliveira	UTAD, Portugal
Paulo Novais	University of Minho, Portugal
Petr Dolezel	University of Pardubice, Czechia
Przemyslaw Korytkowski	West Pomeranian University of Technology, Szczecin, Poland
Reggie Davidrajuh	University of Stavanger, Norway

Richard Duro	University of A Coruña, Spain
Robert Burduk	Wroclaw University of Technology, Poland
Sebastian Saniuk	University of Zielona Gora, Poland
Sebastián Ventura	University of Cordoba, Spain
Stefano Pizzuti	Energy New technologies and sustainable Economic development Agency (ENEA), Italy
Sung-Bae Cho	Yonsei University, South Korea
Tzung-Pei Hong	National University of Kaohsiung, Taiwan
Urko Zurutuza	Mondragon University, Spain
Valeriu Manuel Ionescu	University of Pitesti, Romania
Vicente Matellan	University of Leon, Spain
Wei-Chiang Hong	Jiangsu Normal University, China
Wilfried Elmenreich	Alpen-Adria-Universität Klagenfurt, Austria
Zita Vale	GECAD - ISEP/IPP, Portugal

Special Sessions

Soft Computing Methods in Manufacturing and Management Systems

Program Committee

Damian Krenczyk (Organizer)	Silesian University of Technology, Poland
Bożena Skołud (Organizer)	Silesian University of Technology, Poland
Anna Burduk (Organizer)	Wrocław University of Technology, Poland
Krzysztof Kalinowski (Organizer)	Silesian University of Technology, Poland
Arkadiusz Gola	Lublin University of Technology, Poland
Bozena Skolud	Silesian University of Technology, Poland
Cezary Grabowik	Silesian University of Technology, Poland
Franjo Jovic	University of Osijek, Croatia
Grzegorz Ćwikła	Silesian University of Technology, Poland
Hongze Ma	Kone, Finland
Ivan Kuric	University of Zilina, Slovakia
Iwona Pisz	Opole University, Poland
Laszlo Dudas	University of Miskolc, Hungary
Paul Eric Dossou	ICAM, France
Reggie Davidrajuh	University of Stavanger, Norway
Sebastian Saniuk	University of Zielona Gora, Poland
Wojciech Bozejko	Wroclaw University of Technology, Poland

Soft Computing Applications in the Field of Industrial and Environmental Enterprises

Program Committee

Álvaro Herrero (Organizer)	University of Burgos, Spain
Alfredo Jimenez (Organizer)	KEDGE Business School, Spain
Alberto Rivas	University of Salamanca, Spain
Alfonso González Briones	University of Salamanca, Spain
Angel Arroyo	University of Burgos, Spain
Camelia Chira	Babes-Bolyai University, Romania
David Griol	Universidad Carlos III de Madrid, Spain
Dragan Simic	University of Novi Sad, Serbia
Jose Luis Calvo-Rolle	University of A Coruña, Spain
José Luis Casteleiro	University of A Coruña, Spain
José Ramón Villar	University of Oviedo, Spain
Julio César Puche Regaliza	University of Burgos, Spain
Manuel Grana	University of the Basque Country, Spain
Mercedes Rodríguez Sastre	Rey Juan Carlos I University, Spain
Montserrat Jimenez	Rey Juan Carlos I University, Spain
Pablo Chamoso	University of Salamanca, Spain
Pedro Antonio Gutierrez	University of Cordoba, Spain
Roberto Casado-Vara	University of Salamanca, Spain
Sung-Bae Cho	Yonsei University, South Korea

Optimization, Modeling and Control by Soft Computing Techniques

Program Committee

Eloy Irigoyen Gordo (Organizer)	University of the Basque Country, Spain
Matilde Santos (Organizer)	Universidad Complutense de Madrid, Spain
Jose Luis Calvo-Rolle (Organizer)	University of A Coruña, Spain
Mikel Larrea (Organizer)	University of the Basque Country, Spain
Agustin Jimenez	Polytechnic University of Madrid, Spain
Antonio Robles Alvarez	University of Oviedo, Spain
Antonio Sala	Universitat Politècnica de València, Spain
Antonio Javier Barragán	University of Huelva, Spain
Basil Al-Hadithi	Universidad Politécnica de Madrid, Spain
Fernando Castaño Romero	Polytechnic University of Madrid, Spain
Gerardo Beruvides	Hitachi Europe, Germany
Isabel García-Morales	University of Málaga, Spain
Javier Muguerza	University of the Basque Country, Spain

Jesus Fernandez-Lozano	University of Málaga, Spain
Jesus Lozano	University of Extremadura, Spain
Jose Manuel Lopez-Guede	University of the Basque Country, Spain
Jose-Luis Diez	Universitat Politècnica de València, Spain
Joseba Quevedo	Universitat Politècnica de Catalunya, Spain
Juan Albino Mendez Perez	Universidad de La Laguna, Spain
Juan Albino Mendez Perez	University of La Laguna, Spain
Luciano Alonso	University of Cantabria, Spain
Luis Magdalena	Polytechnic University of Madrid, Spain
Raquel Martinez	University of the Basque Country, Spain
Vicente Gomez-Garay	University of the Basque Country, Spain
Xavier Blasco	Universitat Politècnica de València, Spain

Soft Computing in Aerospace, Mechanical and Civil Engineering: New Methods and Industrial Applications

Program Committee

Soledad Le Clainche (Organizer)	Universidad Politécnica de Madrid, Spain
José M. Pérez (Organizer)	Universidad Politécnica de Madrid, Spain
Ricardo Vinuesa (Organizer)	KTH Royal Institute of Technology, Sweden
Esteban Ferrer	ETSIAE-Universidad Politecnica de Madrid (UPM), Spain
Francisco Martínez Álvarez	Pablo de Olavide University, Spain
Héctor Quintián	University of A Coruña, Spain
José F. Torres	Pablo de Olavide University, Spain
Jose Luis Calvo Rolle	University of A Coruña, Spain
Josem. Perez	Universidad Politécnica de Madrid, Spain

SOCO 2019 Organizing Committee

Francisco Martínez Álvarez	Pablo de Olavide University, Spain
Alicia Troncoso Lora	Pablo de Olavide University, Spain
José F. Torres Maldonado	Pablo de Olavide University, Spain
David Gutiérrez-Avilés	Pablo de Olavide University, Spain
Rubén Pérez Chacón	Pablo de Olavide University, Spain
Ricardo L. Talavera Llames	Pablo de Olavide University, Spain
Federico Divina	Pablo de Olavide University, Spain
Gualberto Asencio Cortés	Pablo de Olavide University, Spain
Miguel García Torres	Pablo de Olavide University, Spain
Cristina Rubio Escudero	University of Seville, Spain
María Martínez Ballesteros	University of Seville, Spain

Álvaro Herrero	University of Burgos, Spain
José Antonio Sáez-Muñoz	University of Salamanca, Spain
Héctor Quintián	University of A Coruña, Spain
Emilio Corchado	University of Salamanca, Spain

Contents

Special Session - Soft Computing Applications in the Field of Industrial and Environmental Enterprises

Machine Learning

Indexes to Find the Optimal Number of Clusters in a Hierarchical Clustering

José David Martín-Fernández(✉), José María Luna-Romera, Beatriz Pontes, and José C. Riquelme-Santos

University of Seville, 41012 Seville, Spain
jfernandez94@us.es
http://grupo.us.es/minerva/

Abstract. Clustering analysis is one of the most commonly used techniques for uncovering patterns in data mining. Most clustering methods require establishing the number of clusters beforehand. However, due to the size of the data currently used, predicting that value is at a high computational cost task in most cases. In this article, we present a clustering technique that avoids this requirement, using hierarchical clustering. There are many examples of this procedure in the literature, most of them focusing on the dissociative or descending subtype, while in this article we cover the agglomerative or ascending subtype. Being more expensive in computational and temporal cost, it nevertheless allows us to obtain very valuable information, regarding elements membership to clusters and their groupings, that is to say, their dendrogram. Finally, several sets of data have been used, varying their dimensionality. For each of them, we provide the calculations of internal validation indexes to test the algorithm developed, studying which of them provides better results to obtain the best possible clustering.

Keywords: Machine Learning · Hierarchical clustering · Internal validation indexes

1 Introduction

In recent years, the size of the information available for various types of studies has grown considerably. Areas like medicine [1], social networks [2], energy [3] or electronic consumption [4] are just a few examples of this, with an increasing amount of data. This information needs to be processed to get some useful knowledge.

Among the different possible solutions to data analysis we focus on Machine Learning techniques, allowing us to extract the main features and a model covering the main information in a dataset. One of the most used model is called clustering, which determines the number of instances of a certain grouping within the data under study. Within the existing grouping variants, hierarchical clustering provides us with very interesting additional information. We can see the

F. Martínez Álvarez et al. (Eds.): SOCO 2019, AISC 950, pp. 3–13, 2020.
https://doi.org/10.1007/978-3-030-20055-8_1

evolution of the clusters in each step of the algorithm, thus studying the grouping of X elements within the data. There exists two subtypes within hierarchical clustering: dissociative or descending, starting from a group with all the elements, and ending in a cluster for each instance in the dataset; or agglomerative or ascending, starting with as many clusters as exists elements in the dataset and ending with a single agglomerative cluster with all of them.

Nowadays, there exists several frameworks to work with Machine Learning techniques to obtain knowledge. One of the most known is Apache Hadoop [5], that is built around the programming model based on the Google paradigm MapReduce [6]. Moreover, one of the most widely used open source projects is Apache Spark [7]. In the Google paradigm, it is read and written from the hard disk on many occasions, which reduces produces a detriment in the speed of data processing. Spark, the number of write/read cycles on the disk, so that intermediate calculations are logically and quickly stored in RAM. To do this, Spark uses a data structure called *"Resilient Distributed Datasets"* (RDD), that are specially designed to parallelize cache calculations with high data volume. In addition, this system contains the scalable library for Machine Learning (*MLlib*), with a series of such as algorithms classification, regression, recommendation systems and clustering techniques, will be of great help to achieve our goal [8].

The purpose of this article is to present a new agglomerative clustering technique implemented in Apache Spark. We have tested our algorithm using diverse datasets, that were created by means of a random database generator. Furthermore, we have applied different internal clustering validation indexes (CVIs) [9] in order to test our clustering results and compare the CVIs performance between the agglomerative hierarchical clustering implemented (AHC) and that provided by Spark as a dissociative modality, Bisecting K-Means (BK-Means) [10].

The rest of the document is organized as follows. Section 2 reviews different types of clustering techniques, as well as the CVIs used during our experimentation. Section 3 describes the algorithm and its implementation. Section 4 presents the experiments carried out and, finally, Sect. 5 summarizes the main conclusions of this work.

2 Related Work

In this section are reviewed the main grouping methods, as well as the internal validation indexes that have been used in our experimentation.

2.1 Clustering Methods

There are several types of clustering algorithms, which could be classified into the following categories depending on the method we use [11]:

- *Grouping by partitions:* Given a set of n elements, the partition method builds K groups, where each partition represents a cluster and $K \leq n$. It's based on

the principle of distance between the individuals, so given an initial K, a first solution could be obtained. Then the process consists in iterating over the dataset, moving objects between groups and trying to improve the previous solution.
- *Density-based methods:* In this approach it is possible to obtain a clustering whose groups are made up of high-density areas and separated from each other by low-density areas.
- *Grid-based methods:* It consist in dividing the elements into a finite cell space which is part of a grid structure. It is applied independently to the size of the data, and the difference is given by the number of cells in each dimension of the generated space.
- *Hierarchical clustering:* It groups data to form a set, or to separate some already existing sets to give origin to other two. Thus, the distance is minimized or the similarity between them is maximized. It is possible to choose different measures to quantify both distance and similarity in this type of grouping.

Within the family of hierarchical algorithms there are two versions or strategies that can be used:

- *Dissociative or descending:* It starts with a cluster that includes all the objects, from which successive divisions are made, forming smaller groups until as many groups as there are elements in the dataset are obtained.
- *Agglomerative or ascending:* It works the opposite way to the descending version. It starts with as many clusters as there are elements in our dataset. At each step, more and more clusters of instances are formed until you end up with a single cluster made up of all available data.

From the first group of algorithms, we can find numerous examples in the literature [12–14]. However, in this work we present a version of the agglomerative option. The main problem of this implementation lies in the computation time needed when treating with large amount of data. Hence, there are few examples in the literature of implementations of this type of strategies [15,16].

2.2 Validation Indexes

The validation of the results obtained by clustering algorithms is a fundamental part of the clustering process. CVI have been typically used to evaluate the partition obtained. Most popular CVIs are *Dunn* [17] and *Silhouette* [18]. Furthermore, indexes presented in [9], have been used in this work, being some of them a simplification of *Dunn* and *Silhouette*.

Dunn and Silhouette. We describe in the following the two most used validation indexes in the literature, which have been implemented in this work for our experiments:

- *Dunn*: Measure widely applied in literature, but open to the possibility of choosing between several variants for calculation. The index is defined by:

$$Dunn = \frac{Min(Inter\text{-}cluster)}{Max(Intra\text{-}cluster)}, \tag{1}$$

where $Inter\text{-}cluster$ is computed as the distance between all the points of a certain cluster M to all the points of a cluster N, and divided by the product of the number of elements in both clusters. $Intra\text{-}cluster$ represents the distance between all the elements that are part of a cluster, divided by the number of instances within that set.
- *Silhouette*: Silhouette distance is calculated for each point i of a cluster:

$$Silhouette_i = \frac{(b_i - a_i)}{Max(b_i, a_i)}, \tag{2}$$

where b_i is the shortest distance between point i to the rest of points of any cluster in which i is not a part of; and a_i is the average distance between point i and the rest of the points of the clusters to which it belongs. Silhouette value is in the interval $-1 \leq Silhouette_i \leq 1$, being its optimal value equals to 1.

Other Indexes. As aforementioned, we have used three other validation indexes from [9] in order to check and compare the goodness of our clustering algorithm. *Davis-Bouldin* is included [19], which uses data object quantities and features inherent to the dataset to set the compactness and separation of the clusters; *BD-Silhouette* is a simplification of the traditional *Silhouette* index based on using intra-cluster and inter-cluster distances to the centroid of each cluster, rather than every element within them; and *BD-Dunn*, which also simplifies the calculations of the internal validation index *Dunn* is used the centroid of each group of data.

3 Our Proposal

In this section we present our approach for AHC. Our technique starts from as many clusters as instances and, in an ascending way, groups them until it reaches a single cluster. In the next, we show the pseudocode of our strategy is shown in Algorithm 1:

The algorithm receives as parameters: a RDD of objects of type *"Distance"* [20], (class created internally to represent the distance between any two elements of the database); the number of clusters to be obtained; the strategy for computing the distances; and the total number of instances in the dataset.

Line 10 refers to the calculation of the Cartesian product, which is necessary to find the distances between the points or clusters of each iteration with respect to the other elements of the RDD of objects of type *"Distance"*. In addition, the step performed on line 11 is configurable according to the designated strategy for calculating the distance between elements in the database, being the implemented options *"min"*, *"max"* and *"avg"*. They refer to the minimum, maximum or mean distance between the distances of the remaining points from each of the points that make up the pair found during line 2 of the algorithm, respectively. In the literature, each of these strategies establishes a different hierarchical clustering typology. Being "minimum or simple link grouping" (*simple linkage*), "maximum or complete link grouping" (*complete linkage*) and "average or average link grouping" (*average linkage*), respectively [21].

Algorithm 1. Agglomerative Hierarchical Clustering (AHC)

Input: RDD with objects *Distance*, number of clusters, strategy for the distance between elements and number of elements of the DataSet.
Output: Hierarchical clustering model.
1: **for** $a \leftarrow 0$ to $(elementsDataSetNumber - numClusters)$ **do**
2: Find the next cluster as the pair of elements with the shortest distance between them within the RDD of *Distance* (they can be two DataSet points; a DataSet object and a cluster; or two clusters).
3: Save the elements of the pair that make up the new cluster.
4: Update the hierarchical clustering model with the cluster found on line 2 and its elements.
5: **if** a < (elementsDataSetNumber - numClusters - 1) **then**
6: Delete from the RDD of *Distance* the match found on line 2.
7: Search the RDD for *Distance* for all the relationships between the first point or cluster found on line 2 and the rest of the elements.
8: Search the RDD for *Distance* for all the relationships between the second point or cluster found on line 2 and the rest of the elements.
9: Delete from the RDD of *Distance* all items found on lines 7 and 8.
10: Calculate the Cartesian product from the elements found in lines 7 and 8.
11: Add to the RDD of *Distance* the distances from the cluster found on line 2 to the other elements calculated on line 10.
12: **end if**
13: Every 5 iterations make a backup copy of the RDD of *Distance*.
14: **end for**

3.1 Implementation

For the creation of this hierarchical grouping, several variants can be made depending on the basis for storing the information, using *"Resilient Distributed Dataset"* (RDD) or *"DataFrames"*. Both objects are provided by Apache Spark. In addition, some functions from MLlib were used, which allows us to delegate some calculations of our algorithm.

Following Spark recommendations, *collect()* and *coalesce()* [22] methods were used in order to accelerate the process. With the *collect()* method, it is possible to obtain data stored in memory during previous calculations, so that it is not necessary to wait for the executions *lazy* of the Spark framework. Through the use of the *coalesce()* method, it is possible to reduce considerably the partitions in which the data are parallelized during the execution of our algorithm. Spark divides the stored data into four times the number of working nodes being used.

Spark offers other alternatives to solve this issue, such as using the *count()* method, which count the number of elements of a given RDD, thus forcing the system to use the data stored in memory. Finally, this alternative was replaced by the former one in our implementation due to the results in terms of execution time between both during the performance tests of the algorithm.

In addition, through experimentation, *checkpoints* each several iterations helps to achieve better performance of the algorithm. Therefore, it is necessary to remember the data that are within the RDD of the distances between all elements. After several performance tests, 5 iterations were inferred as the optimum number for these backups. As for the *coalesce()* method, tests were carried out with several multiples of 4, following Spark's recommendations that establish to use them by multiplying by the number of CPUs used during the execution. The best configuration was to use 8 partitions to distribute the data since the equipment where the tests have been performed has 4 CPUs (it is described in Sect. 4.1). Following Spark's recommendations, it is one of the configurations that usually work best.

4 Experimentation

In this section we present the experimental setup and results obtained using our AHC approach and the comparison with respect to the use of the dissociative clustering algorithm BK-Means.

4.1 Working Environment and Datasets

Our goal is to check the goodness of the different grouping obtained by our algorithm, using several datasets and evaluating the results by means of multiple CVIs. The experiments were executed in: IntelliJ IDEA development environment; the Apache Spark framework using the Scala language; and the Machine Learning library provided by the MLlib framework; a computer with an Intel Core i7-7700HQ CPU with 4 cores of 2.8 GHz, 16 GB of RAM, an SSD of 256 GB and a HDD of 1 TB.

A total of 60 datasets have been used in our experimentation, which were generated by using the database generator in [9]. This tool allowed us to configure the desired number of clusters, dimensions, and the number of points for each cluster.

For the experimentation, three different configurations for the number of clusters (K) have been used: 3, 5 and 7; 20 different configurations for the dimensionality of the data: from 1 to 20; and 100 points for each of cluster. In order to achieve the 20 different dimensions expressed above, a dataset has been taken as the basis for each different K with 20 total dimensions, from which the different dimensions from 1 to 20 have been selected.

4.2 Experimental Results

In order to study which CVI offers the best results using our hierarchical algorithm as the basis for clustering, we must first define how to measure this goodness. In our experiments, the modality *"avg"* distance of the hierarchical clustering algorithm has been chosen, explained in Sect. 3. As for the configuration of the BK-Means algorithm, all the default values for parameters have been set. Since the number of clusters (K) for each of the datasets is known beforehand, we would check in how many datasets each of the indexes matches K. After executing the algorithm for each of the datasets, different cluster numbers on the resulting model for each CVI has been tested. Specifically, K values from 3 to 9 have been used ($3 < K < 9$). This interval has been chosen in order to guarantee that all K values can be found, since, the minimum would be 3 and the maximum would be 7 in the data used during the experimentation.

The results have been grouped according to two criteria: by the number of clusters, and by the number of dimensions in each of the databases. Following the first of the criteria, each index could obtain a maximum of 20 hits in each of the numbers of clusters. Whereas for the second criterion, each index could obtain a maximum of 3 hits for each of the dimensions available in our databases. In this sense, each of the indexes will have two different hits: one grouping all the cluster numbers for each dimension, being able to obtain a maximum of 20; and another grouping all dimensions for each cluster number, being able to obtain a maximum of 3. The results can be summarized in the following Tables 1, 2 and 3:

Table 1. Summary of hits of each index grouped by the number of clusters in each dataset.

Index	AHC (K3)	AHC (K5)	AHC (K7)	BK-Means (K3)	BK-Means (K5)	BK-Means (K7)	AHC (Total)	BK-Means (Total)
Silhouette	**20**	18	14	**20**	18	7	**52**	**45**
Dunn	19	18	14	17	18	7	51	42
Silhouette-BD	17	15	9	13	14	3	41	30
Dunn-BD	19	18	13	17	18	7	50	42
Davis-Bouldin	**20**	18	14	18	18	7	**52**	43

Table 2. Summary of hits of each index grouped by the number of dimension in each dataset in AHC execution.

Index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	Total
Silhouette	1	1	2	2	2	2	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**14**
Dunn	1	1	1	2	2	2	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**14**
Silhouette-BD	0	0	1	2	1	2	**3**	2	**3**	2	**3**	**3**	2	2	2	2	2	**3**	**3**	**3**	7
Dunn-BD	0	1	2	2	2	2	**3**	**3**	2	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	13
Davis-Bouldin	1	1	2	2	2	2	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**14**

Table 3. Summary of hits of each index grouped by the number of dimension in each dataset in BK-Means execution.

Index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	Total
Silhouette	1	1	2	2	2	2	**3**	2	**3**	**3**	**3**	**3**	**3**	2	**3**	2	2	2	2	2	**7**
Dunn	0	0	1	2	2	2	**3**	2	**3**	**3**	**3**	**3**	**3**	2	**3**	2	2	2	2	2	**7**
Silhouette-BD	0	0	0	1	0	0	1	2	2	**3**	**3**	2	2	2	2	2	2	2	2	2	2
Dunn-BD	0	0	1	2	2	2	**3**	2	**3**	**3**	**3**	**3**	**3**	2	**3**	2	2	2	2	2	**7**
Davis-Bouldin	1	0	1	2	2	2	**3**	2	**3**	**3**	**3**	**3**	**3**	2	**3**	2	2	2	2	2	**7**

Calculating the total success percentage of each of the CVIs studied would be as simple as dividing the "Total" columns of the previous tables by the maximum number of clusters and dimensions of those groupings, 60 and 20, respectively. Figure 1 summarizes the global results for each CVI, showing the percentage of total hits, and grouping the tests carried out by the number of clusters, and by the number of dimensions for each dataset.

The best indexes taking into account the grouping of databases by the number of clusters have been *Silhouette* and *Davis-Bouldin* in the case of using our AHC algorithm, both obtaining a success rate of 87%. Studying the case of the BK-Means algorithm it can be observed how the best index has been the only *Silhouette*, with a 75% success rate. If we study the percentages by the number of dimensions, we find the same previous winners plus *Dunn* in the case of using our AHC algorithm, all obtaining a 70% success rate on the number of dimensions. However, in the case of using the BK-Means algorithm, the *Dunn-BD* index would have to be added to the previous winners, all with a 35% success rate.

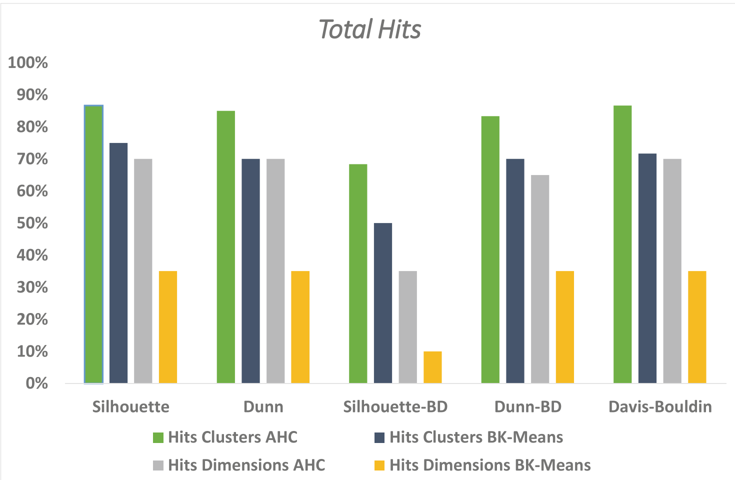

Fig. 1. Percentage of total hits for each of the indexes.