Revista Cubana de Ciencias Informáticas
Vol. 17, No. 4, Octubre-Diciembre, 2023
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 16-26

Tipo de artículo: Artículo original

Temática: Programación paralela y distribuida

Recibido: 25/08/2023| Aceptado: 12/10/2023

Agrupamiento de datos desde un enfoque paralelo

Data clustering from a parallel aproach

Wilfredo Quiala Fonseca [1*] http://orcid.org/0000-0001-9390-8103

[1] Universidad de Oriente. Avenida Patricio Lumumba S/N Santiago de Cuba. Cuba Código Postal 90500. wquiala@uo.edu.cu

*Autor para la correspondencia. (wquiala@uo.edu.cu)

RESUMEN

El algoritmo de agrupamiento DBSCAN es uno de los métodos de agrupamiento por densidad más conocidos debido a su eficiencia y simplicidad. Sin embargo, por su funcionamiento, no puede resolver problemas con una gran cantidad de muestras donde el tiempo de ejecución se considera relevante. En la actualidad, el agrupamiento de grandes cantidades de datos se está convirtiendo en una tarea indispensable. Este problema se conoce como Big Data, donde las técnicas estándar de minería de datos no pueden hacer frente a estos volúmenes de datos. En esta contribución, se propone un enfoque basado en paralelismo con intercambio de mensajes para el agrupamiento DBSCAN. Este modelo nos permite agrupar una gran cantidad de casos desconocidos al mismo tiempo. Para esto, la fase de mapeo determinará los

Revista Cubana de Ciencias Informáticas
Vol. 17, No. 4, Octubre-Diciembre, 2023
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 16-26

conglomerados en las diferentes particiones de los datos. Después, la fase de reducción mezclará y actualizará los conglomerados obtenidos en la fase anterior. Este modelo permite escalar con conjuntos de datos de tamaño arbitrario, simplemente agregando más nodos de computación si es necesario. Además, esta implementación obtiene una velocidad de agrupación, similar a la agrupación del algoritmo clásico DBSCAN.

Palabras clave: agrupamiento por densidades; agrupamiento; programación paralela; DBSCAN.

ABSTRACT

The DBSCAN clustering method is one of the best known density clustering methods due to its efficiency and simplicity. However, by its operation, it cannot address problems with a large number of samples where the execution time is considered relevant. At present, the grouping of large amounts of data is becoming an indispensable task. This problem is known as big data, where standard data mining techniques cannot cope with these data volumes. In this contribution, an approach based on parallelism with message exchange for DBSCAN clustering by density is proposed. This model allows us to classify a large number of unknown cases at the same time. For this, the mapping phase will determine the clusters in the different partitions of the data. Afterwards, the reduction phase will mix and update the clusters obtained from the previous phase. This model allows you to scale with data sets of arbitrary size, simply adding more compute nodes if necessary. In addition, this implementation obtains a clustering rate, similar to the clustering of the classical DBSCAN algorithm.

Keywords: density clustering; clustering; parallel programming; DBSACN.

Revista Cubana de Ciencias Informáticas
Vol. 17, No. 4, Octubre-Diciembre, 2023
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 16-26

Introducción

El agrupamiento de grandes conjuntos de datos se está convirtiendo en una tarea esencial en muchos ámbitos como: la biomedicina, las redes sociales, el marketing, etc. Los avances recientes en la recopilación de datos traen consigo un aumento inexorable de los datos a manejar. El volumen, diversidad y complejidad de dichos datos dificulta los procesos de análisis y extracción del conocimiento, de manera que los modelos de minería de datos estándar necesitan ser rediseñados para funcionar adecuadamente.

El algoritmo DBSCAN (Density-based spatial clustering, DBSCAN) (Ester, Kriegel, Sander y Xu, 1996) pertenece a la familia de algoritmos de agrupamiento por densidades, es un algoritmo de agrupamiento de datos espacial basado en densidad para aplicaciones con ruido (DBSCAN Density-based spatial clustering of applications with noise), propuesto por Martin Ester, Hans-Peter Kriegel, Jörg Sander y Xiaowei Xu en 1996. Este tipo de algoritmos se basan en el concepto de densidad de la vecindad de un punto y miden el número de puntos alcanzables desde este, teniendo en cuenta un radio concreto.

DBSCAN fundamenta el agrupamiento en los siguientes conceptos. Clasifica los puntos como puntos núcleo, frontera (densamente alcanzables), o ruido de la siguiente forma:

1. Un punto P es un punto núcleo si al menos minPts puntos están a una distancia ε de él o, esos puntos son directamente alcanzables desde P. No es posible tener puntos directamente alcanzables desde un punto que no sea un núcleo.
2. Un punto Q es alcanzable desde P si existe una secuencia de puntos p1 ,…, pn donde p1 = P y pn=Q tal que cada punto pi+1 es directamente alcanzable desde pi; es decir, todos los puntos de la secuencia deben ser puntos núcleos, con la posible excepción de Q.
3. Un punto que no sea alcanzable desde cualquier otro punto es considerado ruido.

Si P es un punto núcleo, esté forma un grupo junto a otros puntos (núcleo o no) que sean alcanzables desde él. Cada grupo contiene al menos un punto núcleo. Los puntos no núcleos alcanzables pueden pertenecer al grupo, pero actúan como frontera puesto que no es posible alcanzar más puntos desde estos.

Por lo tanto, el tiempo de respuesta se ve comprometido cuando se aplica en el contexto de grandes volúmenes de datos.

Las recientes tecnologías basadas en la nube ofrecen un entorno ideal para dar solución a este problema. El esquema propuesto destaca un paradigma de programación simple y robusto con la capacidad de afrontar grandes conjuntos de datos en un grupo de nodos de cómputo (cluster node). En los últimos años, diversas técnicas de minería de datos se han adaptado satisfactoriamente mediante el uso de este paradigma, como se muestra en (Microsoft Academic, 2013), (Dean y Ghemawat, 2008). Algunos trabajos relacionados utilizan el modelo de programación MapReduce para el agrupamiento por densidades. Por ejemplo, en (White, 2015) y (Berger y Bokhari, 1987) los autores utilizan consultas DBSCAN dentro de un proceso MapReduce.

En este trabajo se presenta un enfoque paralelo del algoritmo DBSCAN basado en intercambio de mensajes (MPI) para un agrupamiento en un juego de datos considerablemente grande. El PP-DBSCAN ha sido implementado usado 3 etapas o fases. La fase de particionado, de mapeo (agrupamiento local) y la de reducción (actualización y mezcla). La fase de particionado consiste en generar regiones irregulares de grupos de datos que puedan ser procesadas de forma independiente. La fase mapeo consiste en desplegar el cómputo de similitud entre los ejemplos y crear los grupos por cada partición a través de nodos de computó. Como resultado de cada mapeo, cada ejemplo está agrupado y etiquetado (con las etiquetas núcleo, frontera o ruido) y son enviados a la etapa de reducción. La fase reducción determinará cuáles son los grupos finales mezclados y actualizados proporcionados por la fase anterior. En este escrito, se denota este enfoque como PP-DBSCAN. Para probar el rendimiento de este modelo, los experimentos realizados se han llevado a cabo en un conjunto de datos con 768 objetos con 9 atributos (8 numéricos y la clase). El estudio experimental incluye un análisis de la precisión y tiempo de ejecución.

Métodos o Metodología Computacional

En esta sección se explica cómo paralelizar el algoritmo DBSCAN sobre una herramienta MPI. El cómputo se organiza en tres operaciones principales: particionado, mapeo y reducción, las dos últimas fases

decidimos nombrarla igual a como lo hace la terminología Hadoop MapReduce (White, 2015), para simplificar el procedimiento. La fase de particionado debe realizarse por el nodo maestro, el cual es el encargado de repartir los elementos a procesar tal que las particiones sean disjuntas, se pueden utilizar técnicas de particionado de longitud fijas o irregular. La fase de mapeo calculará los grupos utilizando el algoritmo clásico DBSCAN del conjunto ERI (Espacio de Representación Inicial) en las diferentes particiones, guardando para cada ejemplo el grupo y el tipo de objeto (núcleo, frontera y aislado o ruido). La fase reducción procesará los datos resumidos en la fase anterior, con la información enviada por cada nodo vecino, realizando las operaciones de actualización de los grupos y la mezcla de los mismos.

A. Fase de particionado

Sean ERI un conjunto de tamaño arbitrario almacenados en ficheros independientes. La fase de particionado divide el conjunto ERI en un número de particiones tal que a cada nodo se le asigna una sola partición, además identifica cuales van a ser las particiones vecinas, para que cada nodo sepa con quien debe comunicarse en las siguientes fases. En esta etapa se generan particiones formadas por un conjunto de objetos agrupados en regiones irregulares disjuntas, cumpliendo además que las densidades de cada región sean similares (aproximadamente la misma cantidad de elementos).

En la bibliografía se reportan varios métodos de particionados tales como: BSP (Binary Space Partitioning), ESP (Even Split Partitioning) y el RBP (Reduced Boundary Partitioning). Solo se implementó BSP, por su simplicidad en el cómputo. El objetivo fundamental de esta fase es realizar un primer agrupamiento que permita realizar las siguientes etapas de forma paralela y distribuida.

B. Fase de mapeo – Agrupamiento Local

Sea m el número de nodos (procesos de mapeo), cada tarea map (Map1, Map2, ..., Mapm) creará varios sub-agrupamientos de datos. Como resultado del particionado se logra: primero un análisis de forma individual (particiones) que se adapta a la filosofía de paralelización de datos logrando un procesamiento de forma

independiente disminuyendo así el intercambio de datos entre los nodos. En segundo lugar, por estar identificado como repartir los datos aumenta la eficiencia en mecanismo de tolerancia a fallos.

Dado que el objetivo es obtener una implementación que mejore el rendimiento del algoritmo DBSCAN clásico, las tareas map se ejecutan sobre un subconjunto de datos (particiones) correspondiente al conjunto ERIj. Se calculan las distancias entre cada par de objetos, se determinan las eVecindades por objeto y con esto se comienzan a etiquetar los objetos en grupos usando el DBSCAN clásico. Como resultado se producen tablas con la forma <IdPart, IdG, idElem, TElem> (IdPart - identificador de la partición, IdG - identificador del grupo, IdElem - apuntador a la tupla del objeto, TElem - tipo de objeto [Núcleo, Frontera o Ruido]). El Algoritmo 1 expone el pseudocódigo de la función de mapeo. Cuando finaliza cada tarea map, envía a cada nodo vecino los objetos que pertenecen a la frontera de la partición (no se refiere a los elementos clasificados como frontera por el agrupamiento BDSCAN clásico) y la cantidad de grupos que se han creado.

Algoritmo 1. Fase de Mapeo– Agrupamiento Local

```
Entrada: Partición DB = {p1, p2, …, pn}, e, MinPts
Salida: cada elemento de pi se le asocia una bandera (núcleo, frontera o ruido) y el identificador de grupo.
IdGrupo=0
Para cada P no visitado ϵ DB
Marcar p como visitado
nbhdP =GetNeighborhood(p,e,MinPts)
if (Sizeof(nbhdP)<MinPts)
        P.flag=ruido
else
                p.IdGrupo=IdGrupo
        p.flag=nucleo
        para cada q ϵ nbhdP
                if (q no visitado)
                        marcar q como visitado
                        q.IdGrupo=IdGrupo
                        nbhdQ=GetNeighborhood(q,e,MinPts)
                        if (Sizeof(nbhdQ)≥MinPts)
```

Revista Cubana de Ciencias Informáticas
Vol. 17, No. 4, Octubre-Diciembre, 2023
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 16-26

```
                                    q.flag=nucleo
                                    nbhdP=Union(nbhdP, nbhdQ)
                            else
                                    q.flag=frontera
                    else if (q.flag es ruido)
                            q.IdGrupo=IdGrupo
                            q.flag=frontera
IdGrupo++
```

C. Fase Reducción – Actualización y mezcla

El objetivo de la fase de reducción consiste en actualizar y mezclar los grupos encontrados en la fase de mapeo. Teniendo en cuenta que nuestro objetivo es diseñar un modelo que pueda escalar para conjuntos de ERI de tamaño arbitrario independientemente del número de muestras. El Algoritmo 2 describe el funcionamiento de esta fase, una explicación detallada es la siguiente: cuando la fase mapeo termina, cada nodo envía la información necesaria a los nodos vecinos, comenzando el proceso de reducción: primero actualiza los grupos a partir del valor recibido de su vecino, modificando los identificadores de grupos con el incremento de la cantidad de grupos producidos en el vecino inmediato inferior. Luego se realiza la mezcla de grupos. Para lograrlo, analiza cada punto que se encuentra en la frontera de la partición de los nodos vecinos. Si existe algún punto que este en la épsilon vecindad eVecidad de la partición vecina, se actualiza el grupo de todos los elementos que pertenecen a la épsilon vecindad y a los densamente alcanzables por ellos con el identificador de grupo del punto analizado.

Algoritmo 2. Fase de Reducción – Mezcla y actualización de los grupos

```
Entrada: e, MinPts, NGr, CF - conjunto de puntos que son frontera de cada partición
Salida: agrupación <IdGr,Vector>.
Para cada P ϵ DB
 Actualizar p.IdGrupo+= NGr
Para cada x ϵ CF
 nbhdX =GetNeighborhood(x,e,MinPts)
 if (Sizeof(nbhdX)≥MinPts)
```

Revista Cubana de Ciencias Informáticas
Vol. 17, No. 4, Octubre-Diciembre, 2023
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 16-26

```
        para cada q ϵ nbhdX
                q.IdGrupo=x.IdGrup
```

Resultados y discusión

En esta sección se presentan todas las cuestiones planteadas en el estudio experimental.

A. Marco experimental

Se tendrán en cuenta las siguientes medidas para evaluar el rendimiento de la técnica propuesta:

- Tiempo de ejecución: se anotará el tiempo dedicado por PP-DBSCAN en cada fase, así como el tiempo global para el agrupamiento del conjunto completo.

En este trabajo no utilizamos el criterio de precisión para medir la calidad del agrupamiento.

Los experimentos se han llevado a cabo en un cluster formado por 5 nodos: uno maestro y 4 de cómputo. Cada uno de los nodos de cómputo tiene un procesador Intel Core i7 4600, 8 núcleos por procesador a 2.10 Ghz y 8 GB de memoria RAM.

En términos de software, se ha utilizado la distribución de código abierto de MPI. Debe tenerse en cuenta que el número de nodos que pueden ejecutarse en paralelo está configurado a 5.

El estudio experimental se centra en el análisis del efecto del número de nodos (1, 2 y 4). El número de vecinos se fijó en 7 y el radio en 0.3 para el modelo propuesto PP-DBSCAN. Se utilizó el juego de datos “diabetes.arff” de UCI Weka.

Revista Cubana de Ciencias Informáticas
Vol. 17, No. 4, Octubre-Diciembre, 2023
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 16-26

B. Resultados obtenidos y análisis

En esta sección se presenta y analiza los resultados obtenidos en el estudio experimental. Los resultados del agrupamiento del algoritmo DBCAN clásico se utilizan como nuestra referencia base. La tabla 1 recoge el tiempo de ejecución (en segundos) de DBSCAN clásico y de la propuesta del artículo para: 2 y 4 procesadores en paralelo.

Tabla 1 - Comparación de DBSCAN vs PP-DBSCAN.

Algoritmo	**Tiempo (seg.)**	**Procesadores**
DBSCAN	0.11	1
PP-DBSCAN	0.050	2
PP-DBSCAN	0.021	4

Conclusiones

La paralelización es la solución más común para enfrentar problemas de rendimiento en el mundo computacional. Todo problema secuencial que se pueda modelar desde un enfoque paralelo presupone menor tiempo de cómputo. En el caso del agrupamiento de datos, es necesario en algunos casos una modelación diferente del problema secuencial para poder hacerle frente al aprendizaje local, ya que los procesos paralelos trabajan con una porción de los datos. Para solucionar esto, en el agrupamiento basado en densidad fue necesario utilizar algoritmos de particionado, que permitan combinar los resultados intermedios de la paralelización alcanzando resultados similares al proceso secuencial sobre todo los datos como se ha explicado en el presente trabajo. La propuesta abordada propone un agrupamiento en tres fases donde el mapeo y la reducción se ejecutan en paralelo siguiendo la lógica del algoritmo DBSCAN obteniendo resultados superiores en término de velocidad de procesamiento conservando la calidad del agrupamiento secuencial.

Agradecimientos

Al recientemente fallecido a causa de la COVID 19, profesor Ernesto Rodríguez Fernández, al proyecto VLIR de la Universidad de Oriente (http://www.vlir.uo.edu.cu) y al grupo HPC de la Dirección de Informatización de la Universidad de Oriente (http://hpc.uo.edu.cu).

Referencias

Spark, Apache. Apache Spark. Retrieved January, 2018, Vol. 17, P. 2018.

Zeebaree, Subhi Rm, Et Al. Characteristics And Analysis Of Hadoop Distributed Systems. Technology Reports Of Kansai University, 2020, Vol. 62, No 4, P. 1555-1564.

Allam, Sudhir. An Exploratory Survey Of Hadoop Log Analysis Tools. Sudhir Allam," An Exploratory Survey Of Hadoop Log Analysis Tools", International Journal Of Creative Research Thoughts (Ijcrt), Issn, 2018, P. 2320-2882.

Ester M, Kriegel H P, Sander J, Xu X. A Densitybased Algorithm For Discovering Clusters In Large Spatial Databases. Data Mining And Knowledge Discovery, 1996, 96: 226–231.

Hu, Xiaojuan, Et Al. A Mapreduce-Based Improvement Algorithm For Dbscan. Journal Of Algorithms & Computational Technology, 2018, Vol. 12, No 1, P. 53-61.

Chen, Zhihua; Guo, Jianming; Liu, Qing. Dbscan Algorithm Clustering For Massive Ais Data Based On The Hadoop Platform. En 2017 International Conference On Industrial Informatics-Computing Technology, Intelligent Technology, Industrial Information Integration (Iciicii). Ieee, 2017. P. 25-28.

Martínez Blanco, Miquel. Big Data Technologies For High Performance Computing. 2020. Tesis De Licenciatura. Universitat Politècnica De Catalunya.

Caminero Lozano, R. A. (2020). Clasificación De Fallos Con Métodos No Lineales Y Algoritmos De Agrupación Basados En Densidad.

Rodríguez Cuenca, Francisco. Development Of A System For The Extraction And Analysis Of Public Data Form The Stackoverflow Network Using Big Data And Machine Learning Techniques. 2020.

Devidasmandaokar, Rajendra; Jaloree, Shailesh. Extensive Analysis Of Clustering Algorithm For Large Datasets Using Density-Based Clustering And Swarm Intelligence. Annals Of The Romanian Society For Cell Biology, 2021, Vol. 25, No 6, P. 6368-6382.

Bryant, Avory; Cios, Krzysztof. Rnn-Dbscan: A Density-Based Clustering Algorithm Using Reverse Nearest Neighbor Density Estimates. Ieee Transactions On Knowledge And Data Engineering, 2017, Vol. 30, No 6, P. 1109-1121.

González Caminero, Juan, Et Al. Análisis Comparativo De Dos Modelos De Programación Paralela Heterogénea. 2020.

Microsoft Academic Search. Top Publications In Data Mining. Http://Academic.Research.Microsoft.Com/Csdirectory/Paper_Category_7.Html. 2013.

Dean J, Ghemawat S. Mapreduce: Simplified Data Processing On Large Clusters. 2008, 107–113.

White T. Hadoop: The Definitive Guide, 4th Edition. O'reilly Media, Inc., 2015.

Berguer M, Bokhari S. A Partitioningstrategy For Nonuniform Problems On Multiprocessors. Ieee Transactions On Computers, 1987, 36: 570–580.

Wilkinson B, Allen M. Parallel Programming. Prentice-Hall. (1999). Http://Www.Coe.Uncc.Edu/Abw/Parallel/Par_Prog/.

Uci Weka Datasets In-Seasr. Http://Repository.Seasr.Org/Datasets/Uci/Arff/Diabetes.Arff

Bozdemir, Beyza, Et Al. Privacy-Preserving Density-Based Clustering. En Proceedings Of The 2021 Acm Asia Conference On Computer And Communications Security. 2021. P. 658-671.

Corain, Matteo; Garza, Paolo; Asudeh, Abolfazl. Dbscout: A Density-Based Method For Scalable Outlier Detection In Very Large Datasets. En 2021 Ieee 37th International Conference On Data Engineering (Icde). Ieee, 2021. P. 37-48.

Chen, Yewang, Et Al. Knn-Block Dbscan: Fast Clustering For Large-Scale Data. Ieee Transactions On Systems, Man, And Cybernetics: Systems, 2019.

Schubert, Erich, Et Al. Dbscan Revisited, Revisited: Why And How You Should (Still) Use Dbscan. Acm Transactions On Database Systems (Tods), 2017, Vol. 42, No 3, P. 1-21.