ارائه یک روش خوشه‌بندی برای داده‌های حجیم مبتنی بر الگوریتم خوشه بندی سی-میانگین و معماری نگاشت-کاهش

سید محمد رضوی[۱]، محسن کاهانی[۲]
[۱]دکتری مهندسی کامپیوتر، دانشگاه فردوسی مشهد
[۲]استاد دانشکده مهندسی کامپیوتر، دانشگاه فردوسی مشهد
seyyedmohammadrazavi@gmail.com, kahani@um.ac.ir

چکیده

مانند تمامی زمینه‌های تحقیقاتی دیگر در علم کامپیوتر، در خوشه‌بندی نیز همیشه مباحث مربوط به آنالیز الگوریتم‌های خوشه-بندی و پیچیدگی زمانی و فضایی آن مطرح می‌باشد. پدیدار شدن مبحث داده‌های حجیم در سالیان اخیر نیز چالش‌های بسیار زیادی برای پیچیدگی الگوریتم‌های خوشه‌بندی به وجود آورده است. تکنیک‌های سنتی خوشه‌بندی داده‌ها نمی‌توانند برای این حجم از داده مورد استفاده قرار گیرند، دلیل این امر هم پیچیدگی بالا و زمان اجرای بالای آن‌ها می‌باشد. از همین‌رو در این تحقیق الگوریتم خوشه‌بندی نوآورانه‌ای برای کارایی بهتر بر روی داده‌های حجیم ارائه شده است. در این الگوریتم از قدرت الگوریتم کلونی زنبور عسل و همچنین سرعت بالای خواندن و نوشتن در پایگاه داده Apache Hbase کمک گرفته شده است تا الگوریتم خوشه‌بندی با کارایی مناسب و دقت بالا برای حجم بسیار زیادی از داده ارائه شود. نتایج شبیه‌سازی بر روی مجموعه‌داده NUS-WIDE که دارای ۸ گروه از تصاویر مختلف است نشان می‌دهد که الگوریتم ارائه‌شده در مقایسه با سایر روش‌های خوشه‌بندی داده‌های حجیم از کارایی و دقت بالاتری برخوردار است.

کلمات کلیدی: **ارائه یک روش خوشه‌بندی برای داده‌های حجیم مبتنی بر الگوریتم خوشه بندی سی-میانگین و معماری نگاشت-کاهش**

۱. مقدمه

در سالیان گذشته بعد از اینکه چالش‌های مربوط به جمع‌آوری داده به نحوی مرتفع گشتند، اکنون سوال اصلی به چگونگی پردازش بر روی این حجم عظیمی از داده‌ها تبدیل شده است. دانشمندان و پژوهشگران معتقدند که امروزه موضوع داده‌های حجیم، مهم‌ترین چالش پیش رو در علوم کامپیوتر می‌باشد. وب‌سایت‌های اجتماعی مانند فیسبوک[۱] و توئیتر[۲] دارای میلیاردها کاربر هستند که در هر دقیقه صدها گیگابایت اطلاعات تولید می‌کنند. در سایت یوتیوب[۳] نیز یک میلیارد کاربر وجود دارد که در هر دقیقه صدها ساعت فیلم تولید و آپلود می‌کنند [۱] و [۲].

به‌منظور بهره‌برداری و اکتشاف دانش از این حجم بسیار زیاد داده ضروری است که ابزارها و زیرساخت‌های مناسب آن ایجاد و استفاده گردد. تکنیک‌های داده‌کاوی[۴] یکی از معروف‌ترین و معتبرترین شیوه‌ها برای استخراج دانش از داده‌ها هستند [۳] و [۴] و [۵]. خوشه‌بندی یکی از تکنیک‌های داده‌کاوی است که به صورت زیر تعریف می‌شود:

روشی برای افراز داده‌ها به چند گروه به نحوی که داده‌های موجود در یک گروه بیشترین شباهت و داده‌های گروه‌های مختلف کمترین شباهت را داشته باشند [۱].

خوشه‌بندی داده‌ها یکی از روش‌های معمول و محبوب در علوم کامپیوتر و دامنه‌های مربوطه می‌باشد. اگرچه منشا اصلی خوشه‌بندی داده‌ها مربوط به داده‌کاوی می‌باشد، اما خوشه‌بندی در بسیاری از فیلدهای دیگر مانند بیوانفورماتیک[۵]، یادگیری ماشین[۶]، شبکه[۷]، شناسایی الگو[۸] و بسیاری از زمینه‌های تحقیقاتی دیگر مورد استفاده قرار می‌گیرد [۶] و [۷] و [۸].

مانند تمامی زمینه‌های تحقیقاتی دیگر در علم کامپیوتر، در خوشه‌بندی نیز همیشه مباحث مربوط به آنالیز الگوریتم‌های خوشه‌بندی و پیچیدگی زمانی و فضایی آن مطرح می‌باشد. پدیدار شدن مبحث داده‌های حجیم در سالیان اخیر نیز چالش‌های بسیار زیادی برای پیچیدگی الگوریتم‌های خوشه‌بندی به وجود آورده است. از همین‌رو محققان و پژوهشگران سعی در بهبود الگوریتم‌های خوشه‌بندی برای کارایی بهتر بر روی داده‌های حجیم نموده‌اند.

قبل از اینکه بر روی خوشه‌بندی بر روی داده‌های حجیم متمرکز شویم، لازم است تا به سوال اینکه چه مقدار داده به عنوان داده حجیم شناخته می‌شود، پاسخ دهیم. بزدک و هاتاوی تقسیم‌بندی مانند جدول ۱-۱ انجام داده‌اند که تا حد زیادی به پرسش مطرح شده پاسخ داده می‌شود [۹].

[1] Facebook
[2] Tweeter
[3] Youtube
[4] Data Mining
[5] Bio informatics
[6] Machine Learning
[7] Networking
[8] Pattern Recognition

جدول-۱ تقسیم‌بندی داده‌های حجیم [۹]

			Big Data		
Bytes	10^6	10^8	10^{10}	10^{12}	$10^{>12}$
Size	Medium	Large	Huge	Monster	Very Large

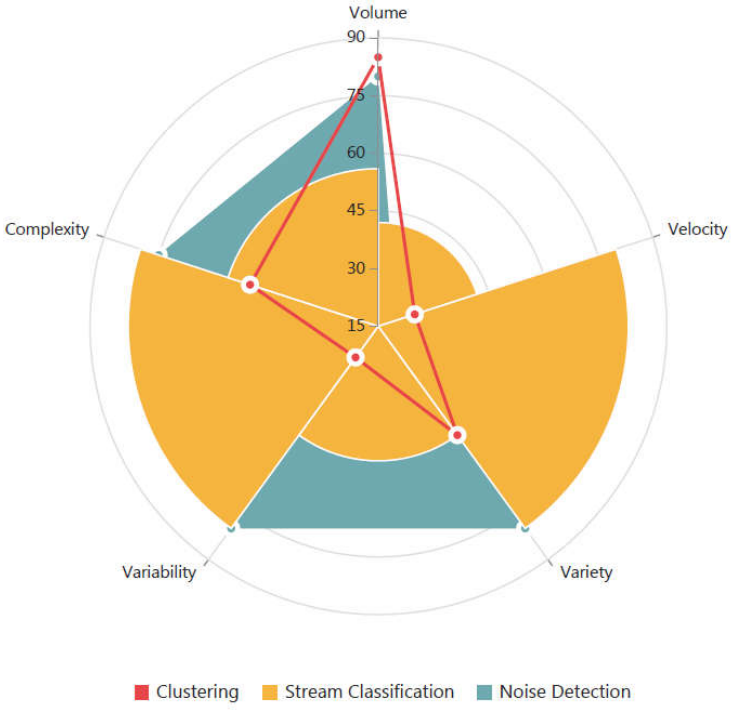
به‌طور کلی چالش‌های موجود در حوزه داده‌های حجیم به ۵ ویژگی زیر خلاصه می‌شود: [۱۰]

- **حجم**[9]: اولین ویژگی حجم است. حجم بسیار زیاد داده غیر ساخت‌یافته[10] که چالش اصلی چگونگی تعیین ارتباط و پیوند میان این حجم عظیمی از داده‌ها می‌باشد.
- **سرعت**[11]: داده با سرعت بسیار زیادی در حال تولید شدن و جریان است و می‌بایست در زمان معقولی بتوان با آن مقابله کرد. پاسخ سریع به داده یکی از چالش‌های حوزه داده‌های حجیم است.
- **تنوع**[12]: چالش بعدی، مدیریت، ادغام و یکپارچه‌سازی حجم زیادی از داده است که از منابع مختلف با ویژگی‌های متفاوت تولید می‌شوند، مانند ایمیل، صوت، تصویر، داده غیرساخت یافته و ... .
- **تغییرپذیری**[13]: عدم سازگاری در جریان داده یکی دیگر از چالش‌های این حوزه می‌باشد. برای مثال در داده‌های شبکه اجتماعی ممکن است به صورت هفتگی یا روزانه، بارگذاری داده بسیار زیادی داشته باشیم که موجب سخت‌تر شدن پردازش این داده‌های غیرساخت یافته می‌شود.
- **پیچیدگی**[14]: داده‌ها از منابع مختلف با ساختارهای متفاوت می‌آیند، کاملا ضروری است که ارتباط و پیوستگی میان این داده‌ها برای شناسایی ارتباط و اتصال میان آن‌ها پردازش شود تا داده‌ها از کنترل خارج نشوند. این موضوع بسیار پیچیده و یکی از چالش‌های این حوزه می‌باشد.

این چالش‌ها در کاربردهای مختلف می‌توانند متفاوت باشند. به‌عبارت دیگر برای هر کاربرد میزان و اندازه چالش‌های فوق می‌تواند متفاوت باشد. در شکل-۱ چندین کاربرد با تفاوت در چالش‌های فوق نشان داده شده‌اند.

[9] Volume
[10] Unstructured
[11] Velocity
[12] Variety
[13] Variability
[14] Complexity

تکنیک‌های سنتی خوشه‌بندی داده‌ها نمی‌توانند برای این حجم از داده مورد استفاده قرار گیرند، دلیل این امر هم پیچیدگی بالا و زمان اجرای بالای آن‌ها می‌باشد. برای مثال، مساله خوشه‌بندی معمول K-means حتی اگر مقدار k برابر ۲ باشد، یک مساله مهارنشدنی[۱۵] می‌باشد. درنتیجه، مقیاس‌پذیری[۱۶] مهم‌ترین چالش حوزه داده‌های حجیم می‌باشد. هدف اصلی افزایش حجم و افزایش سرعت الگوریتم‌های خوشه‌بندی با کمترین تاثیر بر کیفیت خوشه‌بندی آن‌ها می‌باشد.



شکل-۱ تفاوت میزان چالش‌های ذکر شده در کاربردهای متفاوت در داده‌های حجیم

۲. کارهای پیشین

همانطور که در بخش قبلی نیز به آن اشاره شد، امروزه حجم زیادی از داده‌ها از منابع آنلاین و سرویس‌هایی که برای ارائه خدمات به مشتریان طراحی شده است، تولید می‌شود. سرویس‌ها و منابعی مانند شبکه‌های سنسور[۱۷]، مراکز داده محیط ابر[۱۸]، شبکه‌های

[15] NP-hard
[16] Scalability
[17] Sensor Network
[18] Cloud Storage

اجتماعی[19] و ... حجم بسیاری زیادی داده تولید می‌کنند و نیاز ضروری جهان امروز در مدیریت و استفاده مجدد و تحلیل این حجم از داده می‌باشد. یک راه‌حل عمده برای غلبه بر این مشکلات، داشتن یک حجم داده بسیار بزرگ خوشه‌بندی شده می‌باشد که همچنان ویژگی‌ها و خاصیت اصلی داده را حفظ کند و هدف اصلی تکنیک‌های خوشه‌بندی بر روی داده‌های حجیم نیز، افزایش کیفیت افراز و جداسازی این حجم بسیار زیاد از داده می‌باشد. از طرف دیگر با جداسازی و افراز داده‌های غیرساخت یافته باتوجه به کاربردهای متفاوت آن‌ها مانند داده‌کاوی، یادگیری ماشین، شناسایی الگو و ... مابقی فرآیند مربوطه با پیچیدگی کمتر و سرعت بیشتری ممکن می‌باشد. در ادامه انواع مختلف الگوریتم‌های خوشه‌بندی داده‌های حجیم آورده شده است.

در حالت کلی، الگوریتم‌های خوشه‌بندی که بر روی داده‌های حجیم اعمال می‌شوند را می‌توان به دو دسته عمده تقسیم کرد:

۱- الگوریتم‌های خوشه‌بندی که بر روی یک ماشین اجرا می‌شوند.[20]

۲- الگوریتم‌های خوشه‌بندی که بر روی چند ماشین اجرا می‌شوند.[21]

در سالیان اخیر، الگوریتم‌هایی که بر روی چندین ماشین اجرا می‌شوند به دلیل اینکه قابلیت افزایش اندازه و زمان اجرای کمتری دارند، بیشتر از الگوریتم‌هایی که روی یک ماشین اجرا می‌شوند مورد توجه قرار گرفته‌اند.

در شکل-۲ نشان داده شده است که هر کدام از الگوریتم‌های یک-ماشین و چند-ماشین از تکنیک‌های مختلفی تشکل شده‌اند:

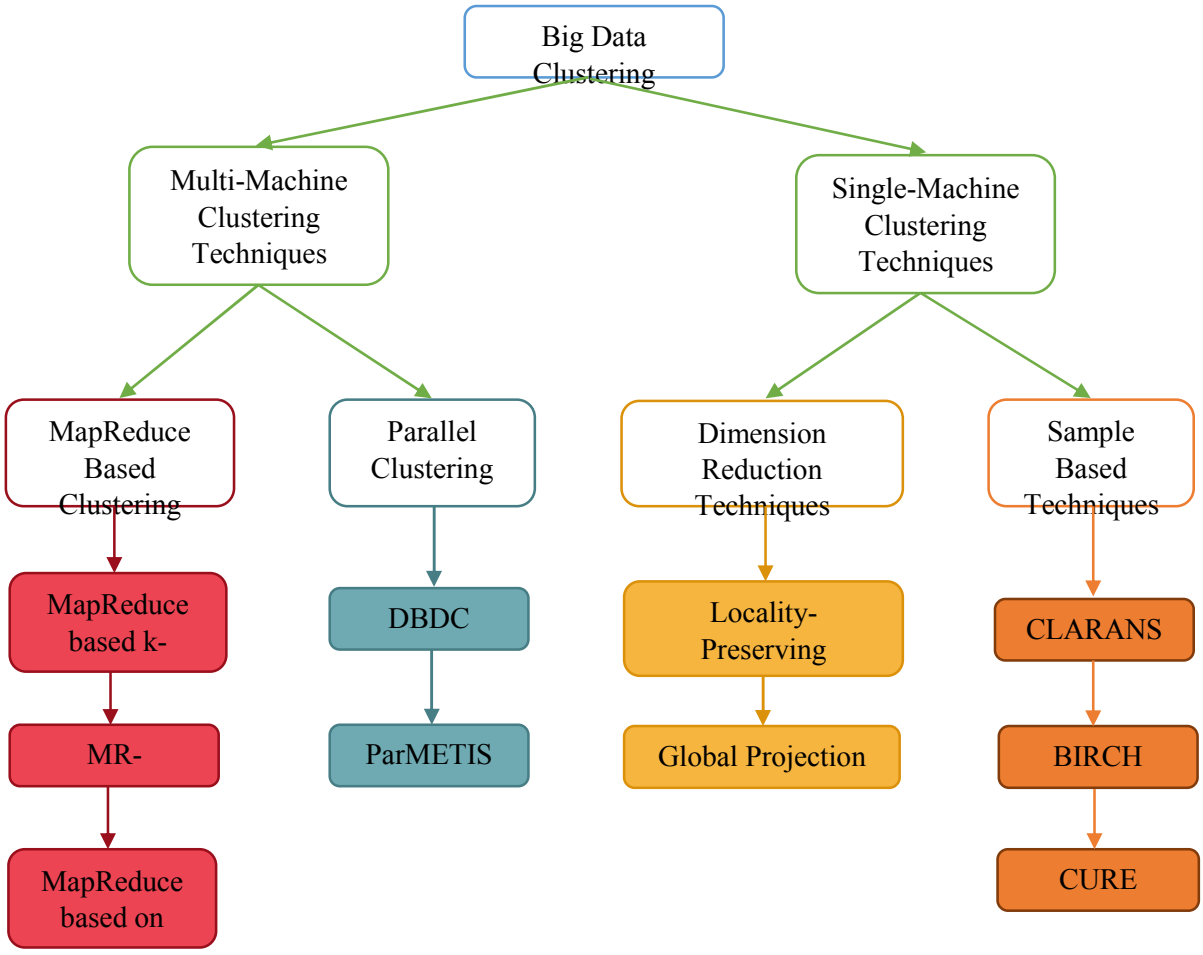
- خوشه‌بندی تک- ماشین
  - تکنیک‌های مبتنی بر نمونه‌برداری[22]
  - تکنیک‌های مبتنی بر کاهش بعد[23]
- خوشه‌بندی چند-ماشین
  - خوشه‌بندی موازی[24]
  - خوشه‌بندی مبتنی بر نگاشت-کاهش[25]

در این بخش باتوجه به تقسیم‌بندی فوق، الگوریتم‌های خوشه‌بندی که بر روی داده‌های حجیم اعمال شده‌اند، مورد بررسی و مرور فنی قرار می‌گیرند.

[19] Social Network
[20] Single-machine clustering algorithm
[21] Multiple-machine clustering algorithm
[22] Sampling Techniques
[23] Dimension Reduction Techniques
[24] Parallel Clustering
[25] Mapreduce-Based Clustering

۱.۲ تکنیک‌های خوشه‌بندی تک-ماشین

تکنیک‌های خوشه‌بندی تک-ماشین شامل دو تکنیک‌های نمونه‌برداری و تکنیک‌های کاهش بعد هستند که در ادامه توضیح داده می‌شوند. این الگوریتم‌ها اولین گروهی از الگوریتم‌های خوشه‌بندی بودند که برای افزایش کارایی و افزایش اندازه مورد استفاده قرار گرفتند و هدف آن‌ها مقابله با فضای حالت نمایی در مسائل خوشه‌بندی می‌باشد. این الگوریتم‌ها از آن جهت مبتنی بر نمونه‌برداری نامیده می‌شوند که به‌جای اینکه عمل خوشه‌بندی را بر روی تمام مجموعه داده انجام دهند، بر روی نمونه‌ای از مجموعه داده انجام می‌دهند و نتیجه را به تمام مجموعه داده تعمیم می‌دهند. این امر موجب تسریع الگوریتم می‌شود زیرا انجام محاسبات بر روی حجم کمتری از داده‌های اعمال می‌شود و در نتیجه پیچیدگی فضایی و زمانی این الگوریتم‌ها کمتر می‌باشد.



شکل-۲ تقسیم‌بندی الگوریتم‌های خوشه‌بندی داده‌های حجیم [۱۱]

۲.۲ تکنیک‌های خوشه‌بندی چند-ماشین

اگرچه تکنیک‌های نمونه‌برداری و کاهش بعد که در بخش قبلی به آن اشاره کردیم توانست کارایی الگوریتم‌های خوشه‌بندی را از نظر زمان و سرعت اجرا بهبود بخشد، اما امروزه سرعت تولید و رشد اطلاعات بسیار بیشتر از سرعت پیشرفت پردازنده‌ها و حافظه‌ها می‌باشد. درنتیجه نمی‌توان ماشینی با ظرفیت حافظه چند صد ترابایت و یا چند پتابایت داشت، از این‌رو می‌بایست به سراغ الگوریتم‌های برویم که قادر به اجرا بر روی چندین ماشین باشند. این تکنیک با قطعه‌بندی و تقسیم حجم بسیار زیادی از داده‌ها به قطعات کوچکتر و تقسیم آن‌ها میان ماشین‌های مختلف از قدرت پردازشی هر ماشین برای تکه‌ای از داده‌های استفاده می‌کند.

الگوریتم‌های خوشه‌بندی که بر روی چند ماشین اجرا می‌شوند، در حالت کلی به دو دسته زیر تقسیم می‌شوند:

۱- توزیع‌شدگی خودکار – نگاشت-کاهش[۲۶]

۲- توزیع‌شدگی غیر خودکار – موازی[۲۷]

در خوشه‌بندی موازی، توسعه‌دهندگان نه تنها درگیر چالش‌های موازی سازی هستند، بلکه درگیر مشکلات و چالش‌های ناشی از چگونگی تقسیم داده‌ها میان ماشین‌ها و چگونگی جمع‌آوری و انتقال نتایج می‌باشند که باعث پیچیدگی بسیار زیاد این الگوریتم‌ها می‌شود. تفاوت میان چارچوب موازی و نگاشت-کاهش در راحتی زیرساخت نگاشت-کاهش برای توسعه‌دهندگان می‌باشد. در این زیرساخت تمامی ارتباطات و تقسیم‌بندی داده‌ها و چگونگی انتقال اطلاعات میان ماشین‌ها به صورت خودکار توسط زیر ساخت نگاشت-کاهش انجام می‌شود. این ویژگی افزایش حجم موازی‌سازی و بالا رفتن قابلیت اعتماد می‌شود. الگوریتم‌های موازی و توزیع-شده خوشه‌بندی از یک چرخه کلی مانند شکل-۳ پیروی می‌کنند.

در اولین گام، داده شکسته می‌شود و میان ماشین‌های مختلف توزیع می‌شود. سپس، هر ماشین خوشه‌بندی را بر روی مجموعه داده‌ای که به آن اختصاص یافته است انجام می‌دهد.

دو چالش اصلی برای خوشه‌بندی‌های موازی و توزیع‌شده عبارتند از:

۱- مینیمم کردن ترافیک انتقال اطلاعات میان ماشین‌ها

۲- دقت پایین‌تر الگوریتم‌ها نسبت به مدل سریال آن‌ها

دقت پایین‌تر در الگوریتم‌های خوشه‌بندی توزیع‌شده به دو دلیل بروز می‌کند:

۱- این امکان وجود دارد که الگوریتم‌های خوشه‌بندی متفاوتی در ماشین‌های متفاوت اجرا شوند.

[26] Automated distributing - MapReduce
[27] Un-automated distributing - parallel

۲- اگر تمام ماشین‌ها نیز از الگوریتم خوشه‌بندی واحد استفاده کنند، در برخی حالات ممکن است که چگونگی تقسیم داده باعث تغییر در نتایج نهایی الگوریتم شود.

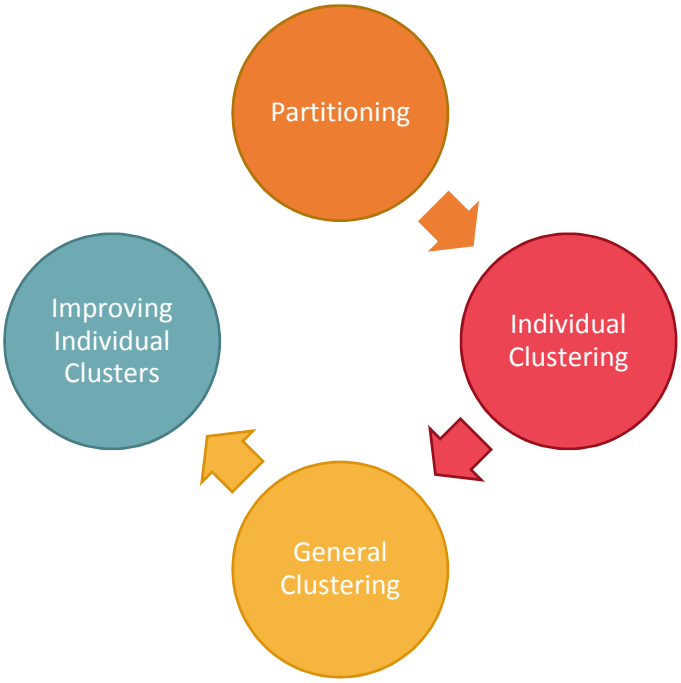
- خوشه‌بندی موازی

اگرچه الگوریتم‌های خوشه‌بندی موازی، پیچیدگی‌های زیادی را برای توسعه‌دهندگان به‌وجود می‌آورد، اما همچنان راه‌حلی منطقی و ارزشمند است زیرا عامل اصلی برای افزایش حجم و سرعت الگوریتم‌های خوشه‌بندی می‌باشد. در ادامه تعدای از این الگوریتم‌ها بررسی می‌شود.

✓ DBDC

DBDC [۱۲, ۱۳] یک الگوریتم توزیع‌شده و مبتنی بر چگالی می‌باشد. تعیین و شناسایی خوشه‌ها از روی اشکال اولیه هدف اصلی در الگوریتم‌های مبتنی بر چگالی می‌باشد. چگالی نقاط یا داده‌ها در داخل خوشه خیلی بیشتر از خارج خوشه می‌باشد درحالی‌که چگالی ناحیه نقاط نویزی بسیار کمتر از چگالی خوشه‌ها می‌باشد. برای انجام خوشه‌بندی‌های محلی که در هر ماشین رخ می‌دهد از یک الگوریتم از پیش‌تعریف شده استفاده می‌شود و برای خوشه‌بندی سراسری از یک الگوریتم مبتنی بر چگالی با نام DBSCAN[۲۸] برای نهایی سازی نتایج استفاده می‌شود [۱۴]. نتایج نشان می‌دهد که با وجود اینکه DBSCAN کیفیت خوشه‌بندی را حفظ می‌کند، سرعت اجرای آن ۳۰ برابر سریع‌تر از نسخه سریال آن می‌باشد.

[28] Density-Based Spatial Clustering of Applications with Noise



شکل-۳ چرخه عمومی الگوریتم‌های خوشه‌بندی چند-ماشین [۱۱]

✓ ParMETIS

ParMETIS [۱۵] نسخه موازی شده الگوریتم METIS [۱۶] می‌باشد و یک الگوریتم افرازبندی چند سطحی می-باشد. افراز گراف یک مساله خوشه‌بندی است که هدف آن یافتن خوشه‌های مناسب از رئوس یک گراف می‌باشد. METIS از سه مرحله اصلی تشکیل شده است:

۱- در این مرحله تطبیق بیشینه[۲۹] بر روی گراف اصلی داده‌ها انجام می‌شود و رئوس با بیشترین تطابق زیرگراف‌ها را تشکیل می‌دهند و این روال تازمانی ادامه می‌یابد که تعداد رئوس به اندازه کافی کوچک شود.

۲- در این مرحله افراز k-تایی با استفاده از الگوریتم چند سطحی دوبخشی بر روی گراف مرحله قبل اعمال می‌شود.

۳- در این مرحله عمل انتساب مجدد داده‌ها از فاز دوم به فاز اول برای نگاشت به داده‌های اصلی صورت می-پذیرد.

[29] Maximal Matching

ParMETIS نسخه توزیع‌شده الگوریتم METIS می‌باشد. به دلیل ماهیت گرافی که در METIS وجود دارد نحوه موازی‌سازی آن با دیگر الگوریتم‌ها متفاوت می‌باشد. در ابتدا تعداد برابر از رئوس گراف بین ماشین‌ها توزیع می‌شود، سپس مساله رنگ‌آمیزی گراف در ماشین‌ها اجرا می‌شود. بعد از آن، رئوس گرافی که دارای رنگ یکسان باشند با یکدیگر به عنوان تطابق در نظر گرفته می‌شوند. افراز چند سطحی نیز به صورت یک سطحی در هر ماشین به صورت جداگانه اعمال می‌شود. مرحله نهایی نیز مشابه الگوریتم METIS در سرتاسر ماشین‌ها انجام می‌شود. آزمایشات نشان می‌دهد که ParMETIS بین ۱۴ تا ۳۵ بار سریع‌تر از METIS می‌باشد درحالی‌که همچنان کیفیت خوشه‌بندی را حفظ می‌کند.

- ✓ GPU-based

مبحث جدیدی که اخیرا در پردازش موازی مورد توجه قرار گرفته است، استفاده از GPU[30] به‌جای CPU برای افزایش سرعت محاسبات می‌باشد چون GPU در پردازش مقدار زیادی داده استاد است چرا که باید حداقل میلیون‌ها و بلکه بیلیون‌ها محاسبه را تنها در ۱ ثانیه انجام دهد. G-DBSCAN [۱۷] نسخه توزیع‌شده و مبتنی بر GPU الگوریتم خوشه‌بندی مبتنی بر چگالی DBSCAN می‌باشد. این الگوریتم یکی از روش‌های جدیدی است که تاکنون در این حوزه ارائه شده است. در این روش از ایندکس کردن داده‌های مبتنی بر گراف برای افزایش قابلیت اطمینان به الگوریتم به منظور افزایش موازی‌سازی و سرعت آن استفاده می‌کنند. G-DBSCAN دارای دو فاز اصلی می‌باشد که هر دو این فازها موازی‌سازی شده‌اند:

۱- ساخت گراف: هر داده معرف یا گره است و زمانی‌که فاصله میان دو داده از یک مقدار از پیش‌تعریف شده کمتر باشد، کمانی میان این دو داده برقرار می‌شود.

۲- تعیین خوشه‌ها: با استفاده از الگوریتم جستجوی اول سطح[31] (BFS) که بر روی گراف ساخته شده در مرحله قبل اعمال می‌شود.

نتایج نشان می‌دهد که G-DBSCAN، ۱۱۲ مرتبه از نسخه سریال خود سریع‌تر است.

- ✓ Mapreduce-based

اگرچه موازی‌سازی الگوریتم‌های خوشه‌بندی موجب بهبود مقیاس‌پذیری و افزایش کارایی آن‌ها شده است، اما پیچیدگی‌ها و سختی‌های توزیع حافظه و پردازنده میان ماشین‌های مختلف همچنان به عنوان چالش اصلی وجود دارد. نگاشت-کاهش چارچوبی که اولین بار توسط گوگل ارائه شده و نسخه متن-باز[32] آن هدوپ می‌باشد، برای حل این چالش ارائه

[30] Graphical Processing Unit
[31] Breath Fisrt Search (BFS)
[32] Open-Source

شده است. در این بخش به بررسی و مرور فنی الگوریتم‌هایی که بر طبق الگو و معماری نگاشت-کاهش پیاده شده‌اند می-پردازیم. شکل-۴ معماری این چارچوب را نشان می‌دهد. چارچوب نگاشت کاهش به عنوان یک مدل قدرتمند در پردازش داده های حجیم استفاده می شود. این مدل برای حل مشکلات گسترده ی محاسباتی در مقیاس های بزرگ و پردازش مجموعه داده های بزرگ در محیط های محاسبات توزیع شده استفاده می شود. الگوریتم‌هایی را که در این بخش به آن-ها اشاره خواهیم کرد، از سه منظر زیر از نظر کارایی و بهبود مورد بررسی قرار خواهیم داد:

۱- افزایش سرعت:[۳۳]

یعنی نسبت زمان اجرای سیستم در حالتی که مجموعه داده ثابت است و تعداد ماشین‌ها افزایش می‌یابد.

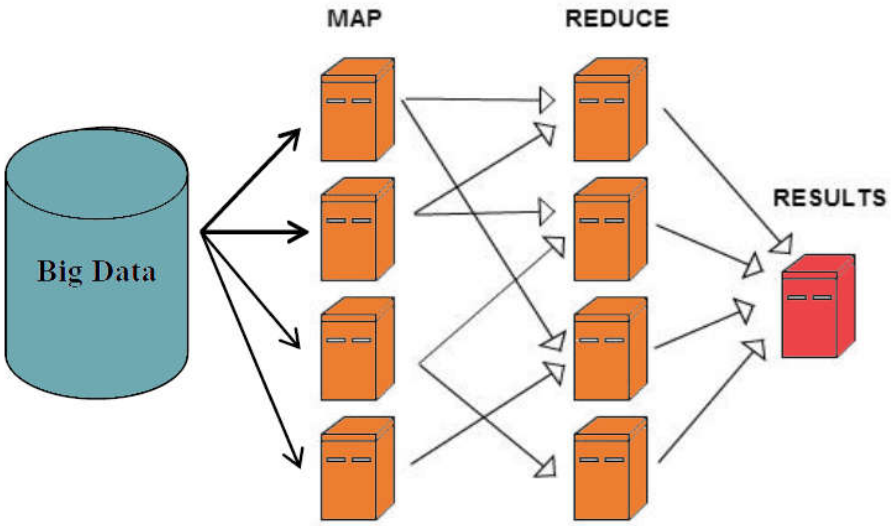
۲- افزایش مقیاس:[۳۴]

به این صورت اندازه گیری می‌شود که آیا سیستمی با x بار بزرگتر قادر به اجرای jobای با x بار بزرگتر در همان زمان می‌باشد.

۳- افزایش اندازه:[۳۵]

تعداد ماشین‌ها ثابت باشد، با افزایش حجم داده زمان اجرا افزایش یابد.

[33] Speed Up
[34] Scale Up
[35] Size Up



شکل-۴ معماری چارچوب hadoop

PKMeans [۱۸] نسخه توزیع‌شده الگوریتم خوشه‌بندی شناخته شده K-Means [۱۹] می‌باشد. هدف اصلی الگوریتم K-Means خوشه‌بندی مجموعه داده‌ها به k خوشه مطلوب است به نحوی که داده‌های داخل هر خوشه بیشترین شباهت را با یکدیگر داشته باشند و داده‌های خوشه‌های متفاوت بیشترین عدم شباهت را داشته باشند. این الگوریتم k داده را در ابتدا به صورت تصادفی انتخاب می‌کند و دو فاز زیر را مرتبا تکرار می‌کند:

۱- هر داده را به نزدیک‌ترین خوشه منتسب می‌کند.

۲- بعد از اتمام تمام داده‌ها به نزدیک‌ترین خوشه، مراکز خوشه‌ها را بامیانگین داده‌های هر خوشه بروزرسانی می‌کند.

PKMeans محاسبات را بین ماشین‌های مختلف برای افزایش سرعت و مقیاس توزیع می‌کند. خوشه‌بندی ابتدایی در فاز نگاشت انجام می‌شود و خوشه‌بندی نهایی و سراسری در فاز کاهش صورت می‌گیرد. PKMeans دارای speed up و size up خطی می‌باشد. برای ۴ ماشین دارای scale up با مقدار ۰.۷۵ می‌باشد. علاوه بر این از نظر کیفیت خوشه‌بندی نیز، PKMeans دارای کیفیتی مشابه نسخه سریال خود دارد.

الگوریتم MR-DBSCAN [۲۰] نسخه مبتنی بر نگاشت-کاهش برای الگوریتم DBSCAN می‌باشد. چالش‌های اصلی که در DBSCAN موازی شده وجود دارد، عبارت است از:

۱- عدم موفقیت در توازن بار میان ماشین‌ها

۲- عدم موفقیت در scale up چون اکثر توابع ضروری آن امکان موازی‌سازی ندارند.

۳- معماری آن به صورتی است که قابلیت حمل آن بسیار پایین است و تطابق خاصی با دیگر معماری‌های موازی ندارد.

در MR-DBSAN مکانیزم جدیدی برای افراز و تقسیم‌بندی محاسبات لحاظ شده است به نحوی که اکثر توابع ضروری نیز قابلیت موازی‌سازی پیدا کرده‌اند. آزمایشات بر روی مجموع داده‌های بسیار بزرگ حاکی از کارایی و scale up این روش دارند.

در این روش [۲۱] الگوریتمی برای خوشه‌بندی داده‌های حجیم با رویکرد کاهش حجم اطلاعات انتقال یافته میان ماشین‌ها ارائه شده است. در واقع هدف اصلی در این الگوریتم این است که با رویکرد Local-Remote Coordination حجم داده بسیار زیادی که به هنگام خوشه‌بندی میان ماشین‌ها در چارچوب نگاشت-کاهش انتقال می‌یابد، کاهش دهد. این الگوریتم به سه فاز اصلی تقسیم می‌شود:

۱- در هر ماشین، داده‌های نمایش دهنده هر خوشه با استفاده از الگوریتم کانوپی[۳۶] جستجو می‌شوند تا بهترین آن‌ها شناسایی شود. در این روش نیز به دلیل اینکه شکل داده‌ها ممکن کروی شکل نباشد بجای در نظر گرفتن یک نقطه به عنوان نماینده هر خوشه از چندین نقطه استفاده کرده است تا اشکال پیچیده داده را نیز پشتیبانی کند. نوع فاصله‌ای که هم که برای تعیین این نقاط مورد استفاده قرار گرفته است فاصله ماهالونوبیس می‌باشد. توجه داشته باشید که بعد از تعیین این نقاط، فقط همین نقاط میان ماشین‌ها برای خوشه‌بندی نهایی انتقال می‌یابند.

۲- یک خوشه‌بندی وزنی بر روی نقاطی که از ماشین‌های مختلف به دست آمده است، اعمال می‌شود و الگوهای کلی میان داده‌ها استخراج می‌گردد.

۳- از روش بیزین برای حل تداخل اینکه یک نقطه به چندین خوشه تعلق داشته باشد استفاده می‌شود.

الگوریتم C-Means احتمالاتی (PCM) یکی از الگوریتم‌های شناخته شده و معروف خوشه‌بندی در داده‌کاوی و حوزه‌های مربوطه می‌باشد. با افزایش حجم داده، پیچیدگی زمانی و داده‌های نویزی در این روش افزایش می‌یابد. در [۲۲] یک الگوریتم PCM وزن‌دار (wkPCM) برای اجرا بر روی داده‌های حجیم و چارجوب نگاشت-کاهش ارائه شده است. در روش ارائه شده از وزن برای تعیین میزان شباهت داده به خوشه‌ها استفاده می‌شود، روشی که به‌راحتی از داداده‌های نویزی صرف نظر می‌کند و آن‌ها را در تعیین خوشه‌ها دخیل نمی‌کند. نسخه توزیع‌شده این الگوریتم که بر روی چارچوب نگاشت-کاهش اجرا می‌شود دارای کارایی بسیار خوبی می‌باشد. آزمایشات نشان می‌دهد که این الگوریتم قادر به خوشه‌بندی داده‌های حجیم در زمان معقول می‌باشد. پیچیدگی زمانی این الگوریتم برای هر ماشین $O(n^2)$ می‌باشد و

[36] Canopy

پیچیدگی فضایی آن نیز $O(n^2)$ می‌باشد. این الگوریتم از نظر کارایی با الگوریتم PCM معمولی و الگوریتم Possibilistic Fuzzy C-Means (PFCM) مقایسه شده است.

در روش [۲۳]، الگوریتم خوشه‌بندی مبتنی بر الگوریتم K-Means [۲۴] تکراری (Iterative K-Means (IKM)) ارائه شده است. در این الگوریتم دو فاز نگاشت-کاهش به صورت پیاده شده‌اند:

۱- در این فاز داده‌ها بارگذاری می‌شوند و IKM بر روی قطعه داده بارگذاری شده اجرا می‌شود.

۲- سپس در فاز کاهش بر روی نتایج به دست آمده از فاز نگاشت، مجددا الگوریتم IKM اجرا می‌شود و نتایج نهایی تولید می‌گردد.

در این روش ادعا شده است که فقط نیاز به یک بار اسکن مجموعه داده وجود دارد که بدین ترتیب حجم تبادل اطلاعات میان نگاشت و کاهش بسیار پایین می‌باشد.

در روش [۲۵] از نسخه بهبود یافته نگاشت-کاهش یعنی نگاشت-برهم‌زدن-کاهش[۳۷] برای ارائه الگوریتم خوشه‌بندی برای داده‌های حجیم استفاده شده است. تمرکز اصلی این الگوریتم بر روی خوشه‌بندی داده‌های مکانی[۳۸] با حجم بسیار زیاد می‌باشد. این الگوریتم در فازهای نگاشت، برهم‌زدن و کاهش به صورت زیر عمل می‌کند:

۱- فاز نگاشت:

مجوعه داده در این فاز به قسمت‌های بسیار کوچک تقسیم می‌شود و نزدیکترین همسایه به هر مجموعه داده مکانی (یک مسیر) در این فاز تعیین می‌شود.

۲- فاز برهم‌زدن:

نتایج به دست آمده در مرحله نگاشت در این قسمت مرتب می‌شوند و به فاز کاهش ارسال می‌گردند.

۳- فاز کاهش:

مجموع تمام نقاط داخل هر خوشه محاسبه و مرکز جدید خوشه از طریق میانگین‌گیری تعیین می‌شود.

۳. روش پیشنهادی

برای بهینه‌سازی خوشه‌بندی در این تحقیق از الگوریتم کلونی زنبور عسل استفاده می‌شود. هدف این است تا با استفاده از قدرت الگوریتم کلونی زنبور عسل دقت و کیفیت خوشه‌بندی را افزایش دهیم. الگوریتم کلونی زنبور عسل از روی نحوه رفتار زنبورهای

[37] Map-Shuffle-Reduce
[38] Spatial Data

عسل برای یافتن مواد غذایی الگو گرفته است. در واقع در این الگوریتم منابع غذایی در قالب پایخ‌های مسئله و روش یافتن منابع غذایی توسط زنبورها در قالب روش رسیدن به پایخ یک مسئله مدل‌سازی شده است. در شکل-۵ شبه‌کد الگوریتم کلونی زنبور عسل آورده شده است.

Algorithm 1: Artifical Bee Colony Algorithm

```
Initialize the set of possible solutions;
while until termination requirements met do
    Select sites for local search ;
    Recruit bees for the selected sites and to evaluate fitness;
    Select the bee with the best fitness;
    Assign the remaining bees to looking for randomly;
    Evaluate the fitness of remaining bees;
    Update probabilities;
end
return BestSolution
```

شکل-۵ شبه‌کد الگوریتم کلونی زنبور عسل

با توجه به این‌که در روش پیشنهادی از آپاچی اچبیس نیز به عنوان فضای ذخیره‌سازی داده‌های میانی در الگوریتم c-means استفاده شده است، در ادامه این پایگاه داده نیز توضیح داده شده است.

اچبیس (HBase) پایگاه‌داده‌ای توزیع شده، متن باز و غیرِ رابطه‌ای است که پس از مدل‌سازی جدول بزرگ گوگل به زبان جاوا نوشته شد [26]. این نرم‌افزار به عنوان بخشی از بنیاد نرم‌افزاری آپاچی توسعه می‌یابد و بر روی اچ‌دی‌اف‌اس اجرا می‌شود و امکاناتی مانند جدول بزرگ را برای هادوپ فراهم می‌آورد [27]. به طور دقیق‌تر، این برنامه راهی با تحمل‌پذیری خطا، برای ذخیره‌سازی تعداد زیادی از داده‌های تُنُکْ را فراهم می‌آورد. اچبیس ویژگی‌های فشرده‌سازی، انجام عملیات در حافظه و فیلتر بلوم را بر اساس مقاله‌ی اصلی که جدول بزرگ برآن اساس نگارش یافت، را ارائه می‌دهد. جداول در اچبیس می‌توانند به عنوان ورودی یا خروجی برای کارهای نگاشت-کاهش که بر روی هادوپ اجرا می‌شوند عمل نمایند. همچنین قابلیت دستیابی از طریق واسط برنامه‌نویس برای جاوا نیز دارا هستند.

۱.۳ الگوریتم ارائه‌شده

برای بهینه‌سازی الگوریتم سی-میانگین با استفاده از کلونی زنبور عسل، هر زنبور را معادل با یک راه‌حل در k dimensional در فضای k بعدی مسئله برای مسئله خوشه‌بندی در نظر می‌گیریم که k تعداد خوشه‌ها می‌باشد. هر المان در یک راه‌حل نشان‌دهنده مرکز هر خوشه خواهد بود که یک بردار p بعدی می‌باشد که p برابر تعداد ویژگی‌های مجموعه داده می‌باشد. در فاز اولیه هر المان در راه‌حل به صورت تصادفی بین دو مقدار مینیمم و ماکزیمم خواهد بود که مقدار مینیمم و ماکزیمم کمترین و بیشترین مقدار برای ویژگی مربوطه در مجموعه داده خواهند بود. هر بردار داده x_i به نزدیک‌ترین مرکز منتسب می‌شود و پس از آن با استفاده از تابع مطلوبیت، مطلوبیت هر راه‌حل محاسبه می‌شود. در واقع تابع مطلوبیت همان تابع هزینه در الگوریتم fcm است که در نتیجه اجرای الگوریتم کلونی زنبور عسل تابع هزینه fcm بهینه می‌شود و خوشه‌بندی با کیفیت بهتری انجام می‌شود.

اکنون می‌بایست الگوریتم فوق را در قالب معماری نگاشت-کاهش طراحی و توسعه دهیم. طبق مدل برنامه‌نویسی نگاشت-کاهش، برای انجام هر وظیفه یا job بر روی داده‌های توزیع‌شده می‌بایست دو تابع mapper و Reducer طراحی و تعریف شود. باتوجه به اینکه در نگاشت-کاهش ورودی و خروجی در HDFS(Hadoop file system) خوانده و نوشته می‌شود سرعت دسترسی به داده‌ها کند است. به‌عبارت دیگر باتوجه به اینکه HDFS حافظه از نوع دسترسی ترتیبی است سرعت خواندن و نوشتن در آن بسیار کند است. از طرف دیگر باتوجه به این‌که الگوریتم خوشه‌بندی یک الگوریتم تکراپذیر است و در آن چندین مرحله تکرار می‌شود می‌بایست در ابتدای هر تکرار داده‌ها از HDFS خوانده شود و در انتهای اجرا، مجددا در HDFS نوشته شود تا در تکرار بعدی مورد استفاده قرار گیرد.

در این تحقیق به‌جای این‌که ورودی و خروجی را در HDFS ذخیره کنیم از Hbase استفاده کرده‌ایم. در واقع برای افزایش سرعت دسترسی به داده‌ها در ابتدا و انتهای هر Iteration از پایگاه داده Hbase استفاده کرده‌ایم. بنابراین مخصوصا در داده‌های با حجم‌های بسیار بالا (بیش از ۱۰۰ ترابایت) استفاده از Hbase تاثیر چشم‌گیری در افزایش سرعت و کارایی الگوریتم خواهد داشت. در ادامه هر کدام از فازهای map و reduce برای الگوریتم ارائه‌شده توضیح داده می‌شود.

فاز نگاشت

فاز map وظیفه انجام پیش‌پردازش بر روی بلوکی از داده‌ها را دارد. این بلوک از داده توسط زیرساخت داده‌های حجیم یا همان ابزار Apache Hadoop در اختیار mapper قرار می‌گیرد. شبه‌کد فاز map در شکل-۶ آورده شده است. با توجه به این‌که individualها در HBase ذخیره شده‌اند، منبع داده ورودی برای هر mapper یک individual از جدول individualها (جمعیت اولیه زنبورها که به‌صورت تصادفی تولید و در HBase ذخیره شده است) در Hbase است که به mapper پاس داده می‌شود. در خط ۲ از mapper، داده‌ها و خوشه‌های اولیه از individual استخراج می‌شود و در خط ۳ مقدار تابع مطلوبیت یا fitness برای این individual و در خط ۴ مقدار احتمال آن محاسبه می‌شود. با توجه به این‌که خروجی فاز map می‌بایست به‌صورت جفت کلید-مقدار باشد، در خط ۵ این الگوریتم کلید individual به همراه مقدار تابع fitness و احتمال آن به‌عنوان خروجی mapper در نظر گرفته شده است.

Algorithm 2 Mapper Phase of FCM bee colony clustering Algorithm

```
1: procedure MAPPER(INDIVIDUAL ID)
2:     points, clusters = translate(ID)
3:     fitness = calculate—fitness(points,clusters)
4:     probability = Calculate the probability values for the individual
5:     submit (ID, (fitness,probability))
6: end procedure
```

شکل-۶ شبه‌کد فاز نگاشت

فاز کاهش

ورودی فاز کاهش همان خروجی مرتب شده فاز نگاشت است. در این مرحله از میان لیست ورودی، جوابی که بیشترین مقدار تابع مطلوبیت را دارد انتخاب می‌شود. سپس جواب جدیدی با استفاده از معادله (۳) تولید می‌شود که در صورت بهتر بودن جایگزین best_solution می‌شود.

Algorithm 3 Reducer Phase of FCM bee colony clustering Algorithm

```
1: procedure REDUCER(LIST<ID,(fitness,probability)>
2:     Select a solution z depending on fitness and probability
3:     Produce new solution z_new and calculate its fitness
4:     If z_new is better than z, replace z with z_new
5:     Submit best solution
6: end procedure
```

شکل-۷ شبه‌کد فاز کاهش

در نهایت نیز بهترین جواب به‌عنوان خروجی ارسال می‌شود.

بنابراین ساختار نگاشت-کاهش فوق به تعداد اجراهای الگوریتم کلونی زنبور عسل اجرا می‌شود و در هر مرحله خروجی فاز نگاشت به همراه individualها در Hbase ذخیره می‌شوند تا به عنوان ورودی برای iteration بعدی مورد استفاده قرار گیرند. استفاده از Hbase باعث می‌شود که روند خواندن ورودی و نوشتن در خروجی با سرعت بسیار بیشتری رخ دهد و مخصوصا در داده‌های با حجم بسیار زیاد، موجب افزایش کارایی و سرعت اجرای الگوریتم می‌شود.

۴. نتایج

برای ارزیابی و محاسبه میزان دقت و کارایی الگوریتم خوشه‌بندی ارائه‌شده از مجموعه داده NUS-WIDE[28] استفاده شده است. این مجموعه داده شامل ۲۶۰۰۰۰ عکس است که از سایت flikr.com دانلود شده است که هر عکس یک object را نشان می‌دهد که به ۸۱ کلاس مختلف افراز شده است. الگوریتم پیاده‌سازی شده بر روی کلاستری با ۶ ماشین فیزیکی اجرا شده است. مشخصات هرکدام از ماشین‌های فیزیکی در جدول-۲ آمده است. از میان ۶ ماشین فیزیکی، یک ماشین به عنوان NameNode و ۵ ماشین دیگر به عنوان DataNode در نظر گرفته شده‌اند. بر روی این کلاستر Apache Hadoop ورژن ۲.۵ و Apache Hbase ورژن ۱.۲ نصب شده است و زبان برنامه‌نویسی استفاده شده نیز python می‌باشد.

جدول-۲ مشخصات ماشین‌های فیزیکی کلاستر

سیستم‌عامل	ابونتو ۱۴.۱۰، نسخه ۶۴ بیتی
پردازنده	Intel Core i7 4.3 GHZ
حافظه اصلی	16 GB
فضای دیسک	1 TB
پهنای باند	100 Mbit/s

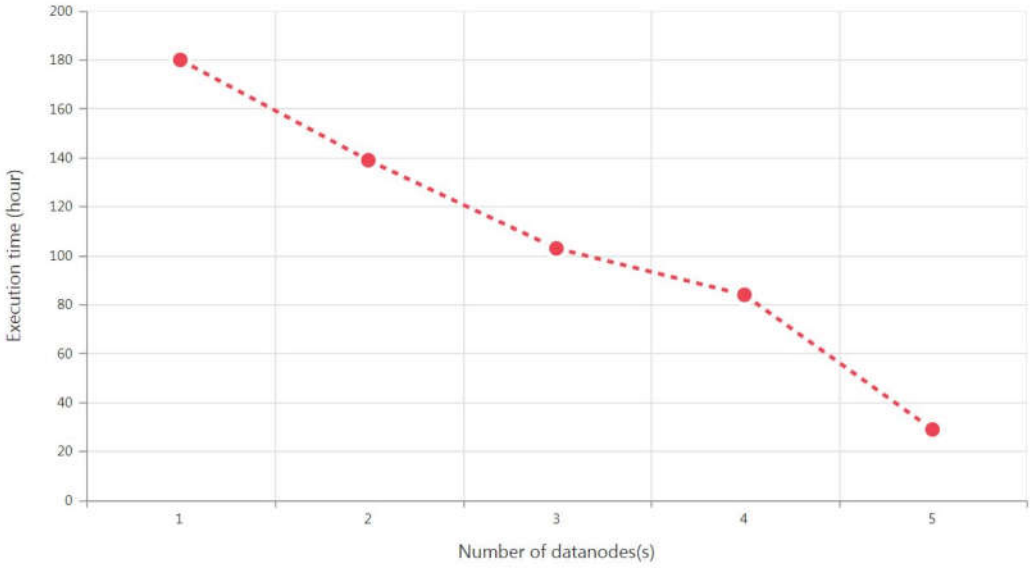
برای ارزیابی میزان کیفیت خوشه‌بندی از پارامتر ARI (Adjusted Rand Score) استفاده شده است که در واقع میزان شباهت میان دو مجموعه (برچسب‌های واقعی و برچسب‌های پیش‌بینی شده) را مشخص می‌کند. بنابراین هرچه ARI بیشتر باشد دقت و کارایی خوشه‌بندی بیشتر بوده است.

روش ارائه‌شده در این تحقیق با روش‌های PCM، wPCM، HOPCM-15 و HOPCM [۲۹] مقایسه شده است. در جدول-۳ نتایج مقایسه روش ارائه‌شده در این تحقیق با روش‌های فوق بر روی ۸ مجموعه داده NUS-WIDE آورده شده است. همان‌طور که مشخص است روش ارائه‌شده در مقایسه با سایر روش‌ها دارای بهبود ۳ درصدی می‌باشد.

جدول-۳ نتایج کیفیت خوشه‌بندی بر روی مجموعه داده NUS-WIDE

Algorithm	1	2	3	4	5	6	7	8	Overall
PCM	0.63	0.59	0.72	0.75	0.71	0.64	0.73	0.59	0.71
wPCM	0.69	0.71	0.78	0.81	0.75	0.79	0.77	0.69	0.77
HOPCM-15	0.91	0.84	0.94	0.91	0.88	0.92	0.82	0.84	0.90
HOPCM	0.90	0.87	0.92	0.94	0.89	0.93	0.85	0.82	0.91
Our method	**0.94**	**0.93**	**0.91**	**0.90**	**0.95**	**0.94**	**0.89**	**0.91**	**0.92**

در شکل-۸ نیز زمان اجرای الگوریتم ارائه‌شده با در نظر گرفتن تعداد datanodeها بین ۱ تا ۵ ارائه شده است که میزان تسریع اجرای الگوریتم به ازای تعداد datanodeهای مختلف نشان داده شده است.



شکل-۸ زمان اجرا الگوریتم ارائه‌شده برای تعداد datanodeهای متفاوت

۵. بحث و نتیجه‌گیری

این تحقیق روشی برای خوشه‌بندی داده‌های با حجم بسیار زیاد ارائه گردید، به‌نحوی‌که علاوه بر حفظ کیفیت و مطلوبیت خوشه-بندی داده‌ها، مرتبه زمانی اجرای آن نیز برای اجرا بر روی حجم بسیار زیادی از داده‌ها مناسب باشد. روشی که در این تحقیق بیان

گردید با استفاده از الگوریتم کلونی زنبور عسل و الگوی معماری نگاشت-کاهش می‌باشد. علاوه بر این، استفاده از پایگاه داده Hbase موجب کارایی بالای این روش نسبت به سایر روش‌ها مخصوصا در حجم داده‌های بسیار بالا گردید. همچنین روش ارائه-شده بر روی یک کلاستر اجرا و ارزیابی گردید تا در محیط توزیع‌شده عملکرد آن مشخص گردد. برای کارهای آتی می‌توان سایر الگوریتم‌های اکتشافی را در قالب یک چارچوب یکپارچه به همراه یک واسط کاربری مناسب بر روی حجم‌های بسیار زیاد اجرا و پیاده‌سازی نمود.

مراجع

1. Havens, T.C., J.C. Bezdek, and M. Palaniswami. *Scalable single linkage hierarchical clustering for big data*. in *Intelligent Sensors, Sensor Networks and Information Processing, 2013 IEEE Eighth International Conference on*. 2013. IEEE.
2. Statistics, Y., *YouTube*. Retrieved December, 2012.
3. Priya, V. and A. Vadivel, *User behaviour pattern mining from WebLog*. International Journal of Data Warehousing and Mining (IJDWM), 2012. **8**(2): p. 1-22.
4. Taniar, D., et al., *Exception rules in association rule mining*. Applied Mathematics and Computation, 2008. **205**(2): p. 735-750.
5. Williams, P.K., C.V. Soares, and J.E. Gilbert, *A clustering rule based approach for classification problems*. International Journal of Data Warehousing and Mining (IJDWM), 2012. **8**(1): p. 1-23.
6. Meyer, F.G. and J. Chinrungrueng, *Spatiotemporal clustering of fMRI time series in the spectral domain*. Medical Image Analysis, 2005. **9**(1): p. 51-68.
7. Ernst, J., G.J. Nau, and Z. Bar-Joseph, *Clustering short time series gene expression data*. Bioinformatics, 2005. **21**(suppl 1): p. i159-i168.
8. Iglesias, F. and W. Kastner, *Analysis of similarity measures in times series clustering for the discovery of building energy patterns*. Energies, 2013. **6**(2): p. 579-597.
9. Hathaway, R.J. and J.C. Bezdek, *Extending fuzzy and probabilistic clustering to very large data sets*. Computational Statistics & Data Analysis, 2006. **51**(1): p. 215-234.
10. McAfee, A., et al., *Big data*. The management revolution. Harvard Bus Rev, 2012. **90**(10): p. 61-67.
11. Shirkhorshidi, A.S., et al., *Big data clustering: a review*, in *Computational Science and Its Applications–ICCSA 2014*. 2014, Springer. p. 707-720.
12. Januzaj, E., H.-P. Kriegel, and M. Pfeifle. *DBDC: Density based distributed clustering*. in *International Conference on Extending Database Technology*. 2004. Springer.
13. Aggarwal, C.C. and C.K. Reddy, *Data clustering: algorithms and applications*. 2013: CRC press.
14. Ester, M., et al. *A density-based algorithm for discovering clusters in large spatial databases with noise*. in *Kdd*. 1996.
15. Karypis, G. and V. Kumar, *Parallel multilevel series k-way partitioning scheme for irregular graphs*. Siam Review, 1999. **41**(2): p. 278-300.
16. Karypis, G. and V. Kumar, *Multilevelk-way partitioning scheme for irregular graphs*. Journal of Parallel and Distributed computing, 1998. **48**(1): p. 96-129.
17. Andrade, G., et al., *G-dbscan: A gpu accelerated algorithm for density-based clustering*. Procedia Computer Science, 2013. **18**: p. 369-378.

18. Zhao, W., H. Ma, and Q. He, *Parallel k-means clustering based on mapreduce*, in *Cloud computing*. 2009, Springer. p. 674-679.
19. Mirkin, B., *Clustering: a data recovery approach*. 2012: CRC Press.
20. He, Y., et al., *MR-DBSCAN: a scalable MapReduce-based DBSCAN algorithm for heavily skewed data.* Frontiers of Computer Science, 2014. **8**(1): p. 83-99.
21. Ma, C., X. Liang, and Y. Ma. *A Succinct Distributive Big Data Clustering Algorithm Based on Local-Remote Coordination*. in *Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on*. 2015. IEEE.
22. Zhang, Q. and Z. Chen, *A weighted kernel possibilistic c□means algorithm based on cloud computing for clustering big data.* International Journal of Communication Systems, 2014. **27**(9): p. 1378-1391.
23. Nguyen, C.D., D.T. Nguyen, and V.-H. Pham, *Parallel two-phase K-means*, in *Computational Science and Its Applications–ICCSA 2013*. 2013, Springer. p. 224-231.
24. Pham, D.T., S.S. Dimov, and C. Nguyen, *An incremental K-means algorithm.* Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 2004. **218**(7): p. 783-795.
25. Hu, C., et al. *Parallel clustering of big data of spatio-temporal trajectory*. in *Natural Computation (ICNC), 2015 11th International Conference on*. 2015. IEEE.
26. Lämmel, R., *Google's MapReduce programming model—Revisited.* Science of computer programming, 2008. **70**(1): p. 1-30.
27. Eadline, D., *Hadoop Fundamentals LiveLessons (Video Training)*. 2013: Addison-Wesley Professional.
28. Chua, T.-S., et al. *NUS-WIDE: a real-world web image database from National University of Singapore*. in *Proceedings of the ACM international conference on image and video retrieval*. 2009. ACM.
29. Zhang, Q., et al., *PPHOPCM: privacy-preserving high-order possibilistic c-means algorithm for big data clustering with cloud computing.* IEEE Transactions on Big Data, 2017.