ارائه یک الگوریتم خوشه‌بندی مبتنی بر چگالی توسعه‌یافته در کلان‌داده‌ها

رضا قائمی[*1]

استادیار، گروه مهندسی کامپیوتر، واحد قوچان، دانشگاه آزاد اسلامی، قوچان، ایران

یعقوب آراد

دانشجوی دکتری، گروه مهندسی کامپیوتر، واحد علوم و تحقیقات، دانشگاه آزاد اسلامی، نیشابور، ایران

فرشته حاج‌قاضی

دانشجوی دکتری، گروه مهندسی کامپیوتر، واحد علوم و تحقیقات، دانشگاه آزاد اسلامی، نیشابور، ایران

مدیریت اطلاعات

دوره ۸، شماره ۲

پاییز و زمستان ۱۴۰۱

چکیده: امروزه تولید داده از طریق تجهیزات هوشمند، ازجمله تلفن‌های همراه، با رشد چشم‌گیری روبه‌رو بوده و خوشه‌بندی یکی از تکنیک‌های پرکاربرد کشف دانش در کلان‌داده‌ها است. خوشه‌بندی مبتنی بر چگالی (DBSCAN)، از الگوریتم‌های خوشه‌بندی کارا در داده‌کاوی بوده و با وجود داشتن مزایا، دارای مشکلاتی ازجمله سختی در تعیین پارامترهای ورودی و همچنین، نداشتن توانایی در کشف خوشه‌هایی با چگالی متفاوت نیز هست. در الگوریتم پیشنهادی این مقاله، از الگوریتم K-DBSCAN در گروه‌بندی داده‌های حجیم با هدف کاهش زمان اجرای خوشه‌بندی الهام گرفته شده است. به‌علاوه، با استفاده از الگوریتم‌های K-Means و H-DBSCAN، چگالی‌های مختلف مجموعه‌داده تشخیص داده می‌شود، برای هر چگالی یک شعاع Eps تعیین شده و سپس، الگوریتم پیشنهادی خوشه‌بندی مبتنی بر چگالی توسعه‌یافته با پارامترهای منطبق روی داده‌ها اعمال می‌شود. در واقع، نوآوری این مقاله استفاده از خوشه‌بندی K-Means و تخمین چگالی‌های مختلف در روش خوشه‌بندی DBSCAN است. الگوریتم پیشنهادی روی چهار مجموعه‌داده استاندارد Image segmentation، Pendigit، Letters و Shuttle control با الگوریتم خوشه‌بندی DBSCAN ساده و دو الگوریتم توسعه‌یافته K-DBSCAN و H-DBSCAN مقایسه شده است. نتایج نشان می‌دهد که الگوریتم پیشنهادی در زمانی که هر دو معیار زمان و دقت در خوشه‌بندی ملاک باشند، در مقایسه با الگوریتم‌های دیگر، الگوریتم برتری است.

کلیدواژه‌ها: کلان‌داده‌ها، خوشه‌بندی، DBSCAN، K-DBSCAN، H-DBSCAN، K-Means.

---

۱. نویسنده مسئول: r.ghaemi@iauq.ac.ir

مقدمه

امروزه، با رشد اینترنت و شبکه‌های اجتماعی با حجم زیادی از داده‌ها مواجه هستیم، به همین دلیل، سیستم‌ها و الگوریتم‌های سنتی نمی‌توانند در زمان‌های پذیرفته‌شده‌ای پاسخ‌گو باشند. الگوریتم‌های سنتی یادگیری ماشین نیز از این قاعده مستثنا نبوده و روی داده‌های بزرگ با استفاده از یک ماشین تک‌پردازنده اجرایی نیستند (Li, 2020). بنابراین، روش‌هایی وجود دارد که الگوریتم‌ها را بهینه‌تر کرده و سرعت اجرای آنها را افزایش دهد. برای تکنیک خوشه‌بندی، الگوریتم‌های مختلفی ارائه شده است که می‌توان از الگوریتم‌های DBSCAN[1] (Li, 2020) و K-Means (De Moura Ventorim, et al., 2021) به‌عنوان روش‌های رایج این زمینه اشاره کرد.

الگوریتم پایه روش‌های خوشه‌بندی، مبتنی بر چگالی DBSCAN است (Li, 2020). این الگوریتم قابلیت کشف خوشه‌هایی با اندازه و اشکال متفاوت از حجم زیادی از داده‌ها را دارد و در مقابل نویز نیز مقاوم است (Sharma, & Upadhyay, 2018). به‌رغم وجود این مزایا، این الگوریتم چند مشکل اساسی نیز دارد. نخست، به دو پارامتر ورودی Minpts و Eps نیاز داشته که تعیین مقدار دقیق این پارامترها به‌خصوص در پایگاه‌داده‌هایی با حجم بالا بسیار دشوار است. دوم، این الگوریتم قابلیت کشف خوشه‌هایی با چگالی متفاوت را ندارد (Wang, Gu & Shun, 2020).

DBSCAN برای داده‌های حجیم بسیار کند بوده و پیچیدگی بالایی دارد. به همین دلیل، برای داده‌های بزرگ نمی‌توان از آن به‌تنهایی استفاده کرد (Li, 2020). از طرفی دیگر، K-Means نیز یکی از الگوریتم‌های پرکاربرد شناخته‌شده مبتنی بر مرکزیت است که تلاش می‌کند مجموع مربعات فواصل اقلیدسی[2] را از مرکز هر خوشه حداقل کند (Sheridan et al, 2020). هدف این مقاله، گروه‌بندی کلان‌داده‌ها در کمترین زمان و با دقت بالا با استفاده از الگوریتم K-Means و برگرفته‌شده از الگوریتم‌های K-DBSCAN (Gholizadeh et al., 2020) و H-DBSCAN (Shaoyuan Weng, 2020) است، چگالی‌های مختلف مجموعه‌داده را تشخیص داده و برای هر چگالی یک شعاع Eps تعیین می‌شود. از آنجا که در DBSCAN محلیت و چگالی داده‌ها اهمیت دارد و در ایجاد خوشه‌ها نیز مهم است، نیازی نیست که فاصله با نقاط دور محاسبه شود. در روش پیشنهادی این مقاله، ابتدا داده‌ها با استفاده از K-Means و با یک تکرار مشخص خوشه‌بندی می‌شوند، سپس چگالی مختلف مجموعه‌داده‌ها تشخیص داده شده و DBSCAN در هر یک از خوشه‌های ایجادشده به‌صورت جداگانه اعمال می‌شود. در واقع، نوآوری در روش پیشنهادی، استفاده از خوشه‌بندی K-Means و تخمین چگالی‌های متفاوت در خوشه‌بندی DBSCAN است. هدف اصلی در این مقاله، رفع مشکل سرعت و تغییرات چگالی الگوریتم DBSCAN است.

در ادامه این مقاله، در بخش ۲ برخی از توسعه و بهبودهای ارائه‌شده برای الگوریتم DBSCAN معرفی شده است. در بخش ۳ روش پیشنهادی تشریح شده است. در بخش ۴ نتایج آزمایش‌ها ارزیابی شده و در نهایت، در بخش ۵ به نتیجه‌گیری و کارهای آینده پرداخته شده است.

1. Density-based spatial clustering of applications with noise (DBSCAN)
2. Euclidean

پیشینه پژوهش

آنکرست و همکاران[1] (۱۹۹۹)، نخستین بهبود برای رفع مشکل تغییرات چگالی الگوریتم DBSCAN را با عنوان الگوریتم OPTICSارائه دادند. هدف کلیه بهبودهای ارائه‌شده عبارت است از: افزایش دقت الگوریتم و به‌طور هم‌زمان، کاهش حساسیت به پارامترهای ورودی و قابلیت تشخیص هر نوع از خوشه‌ها. تعداد زیاد این الگوریتم‌ها، موجب سردرگمی کاربران در انتخاب الگوریتم مناسب شده است.

در سال‌های اخیر، برای داده‌های حجیم، الگوریتم‌های خوشه‌بندی بی‌شماری ارائه شده است. در حالت کلی، این الگوریتم‌ها را می‌توان در دو دسته قرار داد. دسته‌ای از الگوریتم‌ها که روی یک ماشین اجرا شده و دسته دیگر که روی چند ماشین اجرا می‌شوند. روش H-K-Means (Wu, Cheng, Zurita-Milla & Song, 2020)، جزء الگوریتم‌هایی است که روی یک ماشین اجرا می‌شوند و در گروه تکنیک‌های کاهش داده قرار می‌گیرند. در این الگوریتم، داده‌ها در یک ساختار سلسله‌مراتبی کاهش پیدا می‌کنند. به این صورت که مراکز هر خوشه به‌عنوان نماینده داده‌های آن خوشه به سطح بعد انتقال پیدا می‌کنند.

در مقاله حاضر سعی شده است در خوشه‌بندی مبتنی بر چگالی، بهبودی ایجاد شود و این کار با استفاده از ادبیات پژوهشی این حوزه انجام می‌شود که استفاده از خوشه‌بندی K-Means برای ایجاد خوشه‌هایی کوچک‌تر و پرداختن به مشکل تغییرات چگالی با استفاده از الگوریتم OPTICS را شامل می‌شود.

الگوریتم OPTICS (Ankerst et al,1999)، الگوریتم DBSCAN را به‌منظور حل مسئله تغییر چگالی تطبیق داده است. این الگوریتم، برای حل مسئله تغییر چگالی دو فیلد اضافی فاصله دسترسی‌پذیری و فاصله مرکز را ذخیره می‌کند. دقت این الگوریتم بسیار پایین است، به‌گونه‌ای که فقط روی برخی مجموعه‌داده‌های خاص عملکرد مناسبی دارد. از طرفی، اگرچه الگوریتمی مانند DVBSCAN (Wang et al, 2020) دقت مناسبی دارد، اما تعداد زیاد پارامترهای این الگوریتم باعث شده است تا انتخاب دقیق این پارامترها برای کاربران مشکل باشد. از این رو، ارائه یک الگوریتم واحد که به حل کلیه مشکلات ذکرشده بپردازد، بسیار ضروری است.

با توجه به مطالب بیان‌شده، در این مقاله برای رفع مشکل سرعت و همچنین، تغییرات چگالی DBSCAN، الگوریتمی ارائه شده است که ضمن ساده و فهمیدنی بودن، کلیه جنبه‌های ذکرشده را در نظر داشته است. الگوریتم ارائه‌شده ضمن تشخیص خوشه‌هایی با چگالی متفاوت، به پارامترهای ورودی حساسیت پایینی دارد و قابلیت تشخیص خوشه‌هایی با اندازه و اشکال متفاوت و خوشه‌های چسبیده به‌هم و تودرتو را نیز دارد. ایده الگوریتم پیشنهادی به این صورت است که ابتدا، با استفاده از تکنیکی مقادیر مختلف پارامتر Eps به دست می‌آید. سپس، الگوریتم DBSCAN برای اعمال روی مجموعه‌داده با پارامترهای به‌دست‌آمده تطبیق داده می‌شود. الگوریتم پیشنهادی روی مجموعه‌داده‌های استاندارد آزمایش شده و نتایج به‌دست‌آمده با نتایج حاصل از الگوریتم پایه DBSCAN و چند الگوریتم بهبودیافته آن نیز مقایسه شده است.

1. Ankerst, Breunig, Kriegel & Sander

در پژوهش چن و همکارانش[1] (۲۰۲۰)، به‌منظور رفع مشکل، الگوریتم DBSCAN در تجزیه و تحلیل خوشه‌هایی با چگالی متفاوت ارائه شده است. ایده این الگوریتم به این صورت است که قبل از اعمال الگوریتم DBSCAN با استفاده از مفهوم k-dist plot چگالی‌های مختلف را شناسایی کرده و برای هر چگالی یک مقدار Eps متناسب را برمی‌گزینند. بعد از تعیین مقادیر مختلف Eps، الگوریتم DBSCAN را به تعداد چگالی‌های به‌دست‌آمده با استفاده از مقادیر مختلف Eps به‌دست‌آمده روی مجموعه‌داده اعمال می‌کند. منحنی k-dist plot از مرتب‌سازی نقاط مجموعه‌داده براساس فاصله هر نقطه از kامین نزدیک‌ترین همسایه‌اش ساخته می‌شود. بعد از ساخت منحنی k-dist plot، هر تغییر شدید در این منحنی یک چگالی متفاوت را مشخص می‌کند. الگوریتم VDBSCAN به تعیین دقیق مقدار پارامتر k وابستگی دارد، به‌گونه‌ای که انتخاب نادرست آن باعث کاهش دقت نتایج می‌شود.

پژوهش سابور و همکاران[2] (۲۰۲۱)، یکی دیگر از بهبودهای الگوریتم DBSCAN است که از مفهوم فاکتور دورافتادگی محلی و چگالی دسترسی[3]پذیری محلی برای تشخیص خوشه‌هایی با چگالی متفاوت استفاده می‌کند. در این الگوریتم، به‌منظور تشخیص نویز از فاکتور دورافتادگی محلی استفاده شده است، به‌گونه‌ای که اگر فاکتور دورافتادگی محلی یک نقطه کمتر از یک حد آستانه باشد، آن نقطه یک نقطه مرکزی است و در غیر این صورت، نویز محسوب می‌شود. هنگام بسط یک خوشه، یک نقطه در صورتی بسط داده می‌شود که چگالی دسترسی‌پذیری محلی آن نقطه به چگالی دسترسی‌پذیری محلی نقطه مرکزی خوشه متعلق به آن نزدیک باشد. در غیر این صورت، آن نقطه به‌طور ساده به خوشه افزوده شده و بسط داده نمی‌شود. انتخاب پارامترهای مناسب برای الگوریتم LDBSCAN در مقایسه با الگوریتم DBSCAN ساده‌تر است، اما با توجه به اینکه این الگوریتم به چهار پارامتر ورودی نیاز دارد، برای پایگاه‌داده‌های حجیم، ممکن است این تعداد زیاد پارامترهای ورودی مشکل‌ساز شود.

در پژوهش گلان[4] (۲۰۱۹)، از مفاهیم واریانس چگالی خوشه[5] و شاخص شباهت خوشه[6] به‌منظور جلوگیری از بسط خوشه از ناحیه متراکم به ناحیه متراکم‌تر و برعکس استفاده می‌شود. الگوریتم با انتخاب یک نقطه مرکزی شروع به شکل‌دهی خوشه‌ها می‌کند، سپس، کلیه نقاطی که در همسایگی Eps نقطه مرکزی انتخابی باشند را به یک صف وارد می‌کند. این نقاط در صورتی اجازه بسط پیدا می‌کنند که واریانس چگالی خوشه کمتر یا مساوی از حد آستانه α بوده و شاخص شباهت خوشه یعنی اختلاف بین حداقل و حداکثر شیء قرارگرفته در خوشه نیز کمتر از حد آستانه γ باشد. در غیر این صورت، نقطه به‌طور ساده به خوشه افزوده شده و دیگر بسط داده نمی‌شود. این الگوریتم علاوه بر دو پارامتر استفاده‌شده در DBSCAN، به تعیین دو پارامتر α و γ که به‌منظور محدود کردن مقدار تغییر چگالی محلی اجازه داده‌شده در داخل خوشه استفاده می‌شوند، نیز نیاز دارد. الگوریتم DVBSCAN قابلیت

1. Chen et al.
2. Sabor et al.
3. Access density
4. Galán
5. Cluster density variance (CDV)
6. Cluster similarity index (CSI)

تشخیص خوشه‌های با اندازه، اشکال و چگالی متفاوت را دارد و در مقابل نویز نیز مقاوم است. با این حال، این الگوریتم به تعیین چهار پارامتر ورودی نیاز دارد که تعیین چهار پارامتر به‌مراتب دشوارتر از تعیین دو پارامتر نسبت به الگوریتم DBSCAN است. این در حالی است که نتایج این الگوریتم بسیار وابسته به تعیین دقیق این پارامترها است.

در پژوهش چن و همکاران (۲۰۱۸)، الگوریتمی دیگر برای یافتن خوشه‌هایی با چگالی متفاوت است و در این الگوریتم از مفهوم شیء مرکزی هم‌جنس استفاده شده است. شیء مرکزی هم‌جنس به شیء‌ای گفته می‌شود که اولاً، یک شیء مرکزی باشد و ثانیاً، اختلاف چگالی با همسایه‌های خود در حد α باشد. الگوریتم با انتخاب یک شیء مرکزی هم‌جنس کار خود را آغاز می‌کند و تا زمانی خوشه را بسط می‌دهد که به یک شیء مرکزی غیرهم‌جنس که نشان‌دهنده تغییر وسیع در چگالی است، برسد. پیچیدگی زمانی این الگوریتم مانند الگوریتم DBSCAN است و یکی از مزایای آن، T کاهش وابستگی به پارامتر Eps است و این در حالی است که الگوریتم پیشنهادی برای رسیدن به این هدف، پارامتر سومی (پارامتر α) را نیز به الگوریتم افزوده است.

در پژوهش ونگ، گو و شان[1] (۲۰۲۰)، توسعه از الگوریتم DBSCAN آمده است که قابلیت تشخیص خوشه‌هایی با چگالی متفاوت را دارد. الگوریتم در ابتدا چگالی هر نقطه را محاسبه می‌کند، سپس، متراکم‌ترین نقطه را به‌عنوان شیء مرکزی در نظر می‌گیرد و شروع به بسط خوشه به‌وسیله نقاط همسایه با چگالی مشابه می‌کند. هنگام بسط خوشه از بین نقاطی که جزء K-نزدیک‌ترین همسایه شیء مرکزی هستند، فقط نقاطی که چگالی آنها کمتر از میانگین چگالی خوشه باشد، به خوشه افزوده شده و بسط پیدا می‌کنند. در واقع در این الگوریتم، میانگین چگالی خوشه و چگالی هر نقطه تصمیم می‌گیرند که نقطه‌ای متعلق به خوشه‌ای باشد یا خیر. در این مقاله، به مسائل پرنویز و شکاف خوشه‌ها نیز اشاره شده است و این دو مسئله به‌دلیل انتخاب نامناسب مقدار k رخ می‌دهند. بنابراین، با انتخاب یک مقدار مناسب و دقیق برای k می‌توان بر این مشکلات غلبه کرد.

در پژوهش لیا، ژائو، هو، بیان و سانگ[2] (۲۰۱۹)، الگوریتمی به‌منظور غلبه بر مسئله تغییرات چگالی خوشه‌ها آمده است. این الگوریتم، ابتدا تابع چگالی هر نقطه را به دست می‌آورد، سپس، با اعمال DBSCAN روی مجموعه‌داده، مرکز هر خوشه را به‌دست می‌آورد. بعد از آن، به‌ازای هر شیء، تابع هم‌چنان تعداد خوشه‌های ناصحیح را به‌کار برده است.

در پژوهش دنگ[3] (۲۰۲۰)، الگوریتمی با قابلیت تشخیص خوشه‌هایی با چگالی متفاوت آمده است که از ساختار داده KD-Tree برای پردازش کارآمد داده‌هایی با ابعاد بالا، استفاده می‌کند. در واقع، استفاده از ساختار داده KD-Tree محاسبه کارآمد kامین نزدیک‌ترین همسایه‌ها را به‌خصوص برای مجموعه‌داده‌های بزرگ ممکن می‌کند. روال کار این الگوریتم به این صورت است که برای هر نقطه فاصله تا kامین نزدیک‌ترین همسایه را با استفاده از ساختار داده KD-Tree محاسبه کرده، سپس، با مشخص‌کردن زانوها

1. Wang, Gu & Shun
2. Lai, Zhou, Hu, Bian & Song
3. Deng

از روی منحنی k-dist، مجموعه پارامترهای Eps را تخمین می‌زند. یکی از مشکلات این الگوریتم، نیاز به پارامتر ورودی k است. با بررسی روش توضیح داده‌شده در این مقاله، به‌راحتی می‌توان گفت که این الگوریتم نیز نسخه‌ای از الگوریتم VDBSCAN است که با داشتن پرس‌وجوهای ناحیه‌ای بهینه، به‌سبب استفاده از ساختار شاخص KD-Tree برای مجموعه‌داده‌های حجیم نیز استفاده می‌شود. بنابراین، مشکلات اساسی الگوریتم VDBSCAN برای این الگوریتم نیز مطرح است.

در پژوهش هارتمن، ما، هامورلین، پرنال و وانگر[1] (۲۰۱۸) برای تشخیص خوشه‌هایی با چگالی متفاوت، الگوریتمی ارائه شده است که در آن، با استفاده از روش‌های آماری دو پارامتر Eps و Minpts استخراج می‌شوند. یکی از ایرادهای اساسی این مقاله، مقایسه نکردن الگوریتم ارائه‌شده با سایر الگوریتم‌ها است، به‌گونه‌ای که الگوریتم ارائه‌شده فقط با الگوریتم پایه DBSCAN مقایسه شده و عملکرد آن در برابر سایر الگوریتم‌هایی که برای حل مشکل تغییرات چگالی ارائه شده‌اند، مشخص نیست.

اکثر روش‌های ارائه‌شده برای حل مشکل تغییرات چگالی الگوریتم DBSCAN روش‌های بدون نظارت هستند که در آنها از دانش قبلی برای بهبود نتایج خوشه‌بندی استفاده نمی‌شود. در پژوهش کیم و همکاران[2] (۲۰۱۸) نشان داده شده است که می‌توان خوشه‌بندی مبتنی بر چگالی را برای تشخیص خوشه‌هایی با چگالی متفاوت استفاده کرد. روش ارائه‌شده در این مقاله به این صورت است که در مرحله نخست، مجموعه‌داده را به سطوح چگالی متفاوت تقسیم کرده و برای هر سطح چگالی، پارامترهای چگالی مناسب مربوط به آن سطح را تعیین می‌کند. در ادامه، الگوریتم مربوطه با استفاده از محدودیت‌های دوبه‌دو فرایند خوشه‌بندی را بر مبنای پارامترهای به‌دست‌آمده بسط می‌دهد. نتایج ارزیابی این الگوریتم نشان می‌دهد که در مقایسه با برخی از الگوریتم‌های خوشه‌بندی نیمه‌نظارتی و بدون نظارت، نتایج بهتری ارائه شده است.

در پژوهش لوهیچی، گزرا و بن عبدالله[3] (۲۰۱۸)، بهبود دیگری از الگوریتم DBSCAN آمده است که قابلیت تشخیص خوشه‌هایی با چگالی متفاوت را دارد. این الگوریتم در دو فاز انجام می‌شود. در فاز نخست، با استفاده از فرایند ریاضی (یک تابع هموار چندضابطه‌ای ـ چندجمله‌ای) روی فواصل k نزدیک‌ترین همسایه، تعداد سطوح چگالی مشخص می‌شود. در مرحله بعدی، از سطوح چگالی به‌دست‌آمده در مرحله نخست، به‌عنوان آستانه‌های چگالی محلی برای تشخیص خوشه‌هایی با چگالی و اشکال مختلف استفاده می‌شود. این الگوریتم در مقایسه با الگوریتم DBSCAN از دقت بالاتری برخوردار است. الگوریتم‌های خوشه‌بندی مبتنی بر چگالی، روش‌های بسیار ارزشمندی برای خوشه‌بندی جریان‌های داده هستند. به‌تازگی، تعدادی الگوریتم خوشه‌بندی مبتنی بر چگالی برای خوشه‌بندی جریان داده‌ها ارائه شده که یکی از ایرادهای این الگوریتم‌ها، کاهش کیفیت خوشه‌بندی به‌دلیل وجود خوشه‌هایی با چگالی متفاوت است.

1. Hartmann, Ma, Hameurlain, Pernul & Wagner
2. Kim et al.
3. Louhichi, Gzara & Ben-Abdallah

در پژوهش یو و همکاران[1] (۲۰۲۱)، الگوریتمی ارائه شده که توانایی خوشه‌بندی جریان‌های داده‌ای با چگالی‌های متفاوت را دارد. الگوریتم ارائه‌شده از روش مبتنی بر Grid برای مدیریت نویز و داده‌های با چگالی متفاوت و همچنین، برای کاهش زمان ادغام خوشه‌ها استفاده کرده است. همچنین، در پژوهش حیدری، البرزی، رادفر، افشارکاظمی و رجب‌زاده[2] (۲۰۱۹)، الگوریتم دیگری برای خوشه‌بندی جریان‌های داده‌ای غیرایستا ارائه شده است که توانایی تشخیص خوشه‌هایی با چگالی متفاوت از جریان‌های داده‌ای را دارد. یکی از مزایای الگوریتم ارائه‌شده، کاهش وابستگی به پارامترهای ورودی است.

با توجه به اینکه وقتی خوشه‌ها نزدیک به یکدیگر باشند، ممکن است الگوریتم DBSCAN با شکست مواجه شود، در پژوهش لوهیچی و همکاران (۲۰۱۸) الگوریتمی به‌منظور رفع مشکل خوشه‌های مجاور ارائه شده است. در این الگوریتم به‌جای استفاده از مفهوم دسترسی‌پذیری چگالی از مفهوم دسترسی‌پذیری چگالی مرکزی استفاده شده است. در واقع، این الگوریتم در زنجیره دسترسی‌پذیری بهبود انجام داده، به‌گونه‌ای که این زنجیره فقط شامل اشیای مرکزی است. روال کار به این صورت است که ابتدا، خوشه‌های متشکل از اشیای مرکزی پیدا شده، سپس، اشیای حاشیه‌ای به نزدیک‌ترین شیء مرکزی تخصیص داده می‌شوند. این الگوریتم روی داده‌های مکانی و غیرمکانی اعمال‌شدنی است. با توجه به هدف اصلی الگوریتم ارائه‌شده، این الگوریتم به‌خصوص در مجموعه‌داده‌های متراکم شامل خوشه‌های نزدیک به یکدیگر به‌خوبی عمل می‌کند و در سایر موارد نتایج نزدیک به الگوریتم DBSCAN را تولید می‌کند.

یکی از مشکلات الگوریتم DBSCAN، بحث پیچیدگی زمانی بالای این الگوریتم در مجموعه‌داده‌هایی با حجم و بعد بالا است. در پژوهش بنچینی، کریسیون، دوکانژ، مارسلونی و رندا[3] (۲۰۲۰) با هدف بهبود زمان الگوریتم DBSCAN یک راه‌کار ارائه شده است. در این الگوریتم، اشیایی که توسط الگوریتم به‌عنوان بذر برای بسط دادن انتخاب می‌شوند، بهبود داده شده‌اند تا به کارایی بهتری برسد. با توجه به اینکه می‌توان از بسیاری از پرس‌وجوهای ناحیه‌ای برای یافتن همسایه‌های اشیا چشم‌پوشی کرد، الگوریتم FDBSCAN تعدادی شیء را به نمایندگی از کلیه اشیا شامل داده‌هایی با چگالی متفاوت انتخاب می‌کند. این الگوریتم علاوه بر افزودن افزایش نقاط، توانایی افزودن افزایشی خوشه‌ها را نیز دارد. الگوریتم ارائه‌شده با عنوان IMD-DBSCAN، نسخه افزایشی الگوریتم MDDBSCAN (لوهیچی و همکاران، ۲۰۱۸) است که این الگوریتم در مقایسه با DBSCAN کارایی بهتر و پیچیدگی محاسباتی و زمانی کمتری نیز دارد، اما در مقایسه با DBSCAN دقت پایین‌تری داشته و اشیای بیشتری در آن از دست می‌روند. افزون بر این، الگوریتم ارائه‌شده در پژوهش‌های کائو، ون و سابل[4] (۲۰۱۸) و پاولیس، دولگا و سینگلتن[5] (۲۰۱۷) نیز روش‌هایی برای بهبود پیچیدگی زمانی DBSCAN ارائه شده است.

1. Yu et al.
2. Heidari, Alborzi, Radfar, Afsharkazemi & Rajabzadeh
3. Bechini, Criscione, Ducange, Marcelloni & Renda
4. Kuo, Wen & Sabel
5. Pavlis, Dolega & Singleton

الگوریتم DBSCAN متکی بر مفهوم چگالی خوشه‌ها است و به‌منظور کشف خوشه‌هایی با اشکال مختلف به‌همراه نویز است. در پژوهش باتس[1] (۲۰۲۱) الگوریتمی ارائه شده که قادر است اشکال هندسی غیر از نقطه مانند چندضلعی‌های دوبعدی را نیز خوشه‌بندی کند. همچنین، این الگوریتم توانایی این را دارد تا اشیای نقطه را به‌خوبی اشیای بسط‌یافته مکانی طبق هر دو خصوصیت مکانی و غیرمکانی آن اشیا خوشه‌بندی کند. افزون بر این، در این پژوهش، کاربردهایی از دنیای واقعی مانند علوم زمین‌شناسی، زیست شناسی، نجوم و جغرافیا برای این الگوریتم ارائه شده است. الگوریتم ارائه‌شده، به‌خصوص در پایگاه داده‌های بسیار حجیم، خوب عمل می‌کند. این الگوریتم، خوشه‌بندی یک‌سطحی را ایجاد می‌کند و این در حالی است که ممکن است خوشه‌بندی سلسله‌مراتبی مفیدتر باشد، به‌خصوص زمانی که پارامترهای ورودی مناسب را نتوان به‌دقت برآورد کرد.

در پژوهش چیموای و آنورادها[2] (۲۰۱۸) یکی دیگر از توسعه‌های الگوریتم DBSCAN آمده است که برخلاف الگوریتم DBSCAN، قابلیت کشف خوشه‌ها را مطابق با مقادیر مکانی، غیرمکانی و زمانی اشیا دارد. این الگوریتم از سه جهت با الگوریتم DBSCAN تفاوت دارد. نخست، برخلاف الگوریتم DBSCAN و سایر الگوریتم‌های مبتنی بر چگالی موجود، قابلیت خوشه‌بندی داده‌های مکانی ـ زمانی را طبق خصوصیت‌های مکانی، غیرمکانی و زمانی اشیا دارد. دوم، برخلاف DBSCAN، در مواقعی که خوشه‌هایی با چگالی متفاوت در مجموعه‌داده وجود داشته باشند نیز قابلیت تشخیص نویز را دارد و درنهایت، اگر مقادیر غیرمکانی اشیای همسایه تفاوت اندکی داشته باشند و خوشه‌ها مجاور یکدیگر باشند، مقادیر اشیای حاشیه‌ای در یک طرف خوشه ممکن است بسیار متفاوت با مقادیر اشیای حاشیه‌ای در طرف مقابل باشند. الگوریتم ST-DBSCAN این مشکل را با مقایسه مقدار میانگین یک خوشه با مقادیر اشیای جدید افزوده‌شده به خوشه حل می‌کند. پیچیدگی زمانی این الگوریتم مانند DBSCAN است. الگوریتم ST-DBSCAN کاربردهای متعددی دارد که از جمله این کاربردها می‌توان به سیستم اطلاعات جغرافیا، تصاویر پزشکی و پیش‌بینی وضع هوا اشاره کرد. یکی از ضعف‌های این الگوریتم، ناتوانی در کشف خوشه‌هایی با چگالی متفاوت است. همچنین، پارامترهای ورودی آن به‌صورت خودکار تولید نمی‌شوند.

با توجه به مرور ادبیات مشخص می شود که در محیط‌های انباره داده، به‌صورت دوره‌ای حجمی از داده‌ها به داده‌های موجود در انباره اضافه می‌شوند. از این رو، نیاز است قبل از اینکه انباره داده در دسترس کاربران قرار بگیرد، خوشه‌هایی که از قبل کشف شده‌اند با توجه به داده‌های جدید افزوده‌شده، به‌روز شوند و این خوشه‌بندی می‌تواند به ارائه خدمات با کیفیت در حوزه‌های مختلف منجر شود. در پژوهش‌های مرورشده، مشخص است که برای خوشه‌بندی، از روش‌های مبتنی بر چگالی به‌خوبی استفاده شده که برخی از موارد در این مقاله بررسی شد. نقطه اشتراک کلیه این پژوهش‌ها، لزوم خوشه‌بندی در داده‌های حجیم است و وجه تمایز آنها در دو معیار سرعت و دقت خلاصه می‌شود. با توجه به نتایج مقاله‌های موجود در این حوزه، مشخص است که هر دو مسئله سرعت و دقت در خوشه‌بندی اهمیت دارد،

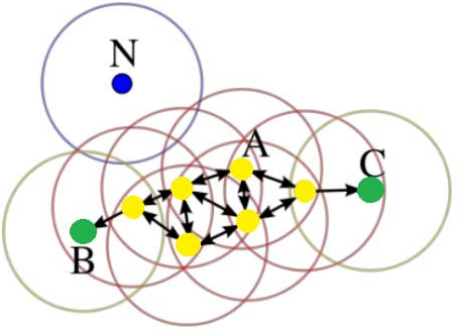
1. Botts
2. Chimwayi & Anuradha

بنابراین، روش‌های مختلف ترکیبی با خوشه‌بندی مبتنی بر چگالی توسعه یافته است. از این رو، با توجه به مرور ادبیات، در این مقاله هدف روش پیشنهادی ارتقای هم‌زمان سرعت و دقت است و در این خصوص، مدلی برگرفته از خوشه‌بندی ترکیبی با روش K-Means و همچنین، تخمین چگالی مختلف در داده‌ها ارائه شده است.

روش پژوهش

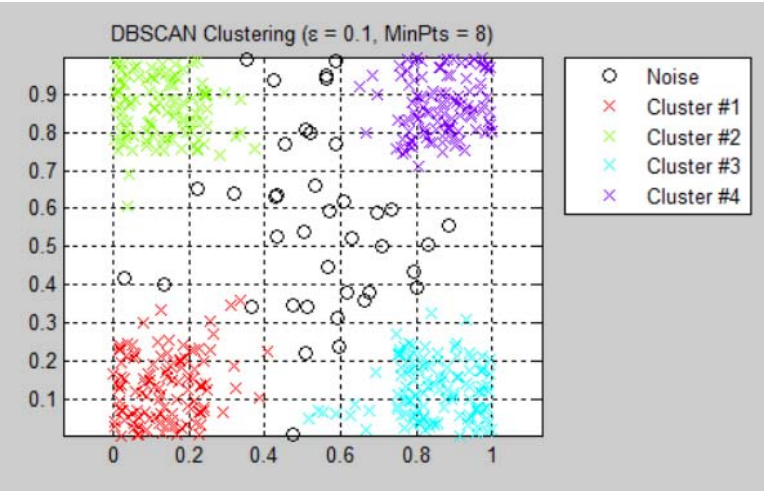
خوشه‌بندی یک مجموعه‌داده حجیم توسط الگوریتم‌های داده‌کاوی شناخته‌شده، زمان‌بر است. بنابراین، با توجه به افزایش حجم داده‌ها و کاهش توان الگوریتم‌ها در پردازش حجم عظیمی از داده‌ها، نیاز به ارائه روش‌های جدید بیش از پیش احساس می‌شود. الگوریتم پیشنهادی در این مقاله، با هدف افزایش سرعت خوشه‌بندی داده در عین حفظ کیفیت خوشه‌بندی ارائه شده است و بهبودیافته الگوریتم مشهور DBSCAN است. الگوریتم DBSCAN جزء الگوریتم‌های مبتنی بر چگالی است که وجود نویزها در داده‌های اصلی را به‌خوبی تشخیص می‌دهد. این روش، نقاط را به سه گروه شامل نقاط core، reachable-density و noise طبقه‌بندی می‌کند.

یک نقطه، core در نظر گرفته می‌شود، در صورتی که در فاصله ε از آن به‌اندازه Minpts نقطه وجود داشته باشد (با احتساب خود نقطه core) که این نقاط به‌صورت مستقیم از طریق نقاط core قابل دستیابی هستند. سایر نقاط هر خوشهDBSCAN نقطه reachable-density در نظر گرفته می‌شوند که این نقاط به‌طور غیرمستقیم به نقاط core متصل هستند. به‌طبع نقاطی که نتوانند به‌طور مستقیم یا غیرمستقیم به نقاط core بپیوندند، خارج از خوشه‌ها قرار گرفته و به‌عنوان نویز در نظر گرفته می‌شوند (شکل ۱). Minpts با مقدار عددی ۴ است و نقطه *A* و سایر نقاط زرد رنگ اطراف آن به‌عنوان نقاط core شناخته شده‌اند و نقطه *N* نیز noise تشخیص داده شده است. شکل ۲ نیز خوشه‌بندی داده، پس از اعمال الگوریتم DBSCANرا نشان می‌دهد.



شکل ۱. نمایش خوشه‌بندی داده‌ها با الگوریتم DBSCAN
(Heidari et al., 2019)

شکل ۲، اجرای خوشه‌بندی DBSCAN معمولی را نشان می‌دهد که برای شناسایی نقاط core در یک خوشه و نقاط noise، به محاسبه فاصله هر نقطه تا کلیه نقاط دیگر نیاز است که این خود، افزایش حجم محاسبات، زمان اجرا و به‌دنبال آن کاهش سرعت، به‌خصوص برای داده‌های حجیم را به‌همراه دارد. بنابراین، برای افزایش سرعت DBSCAN روی داده‌های حجیم در روش پیشنهادی این مقاله، از الگوریتم K-Means در ابتدای الگوریتم DBSCAN استفاده می‌شود. هدف این ایده، این است که داده‌های نزدیک به یکدیگر تا حد ممکن در یک خوشه قرار گیرند و الگوریتم DBSCAN نقاط با فواصل دور را در محاسبات خود در نظر نگیرد. به این ترتیب، این الگوریتم در هر خوشه به‌دست‌آمده از K-Means به‌طور مجزا اجرا شده و نقاط core را تشخیص می‌دهد.

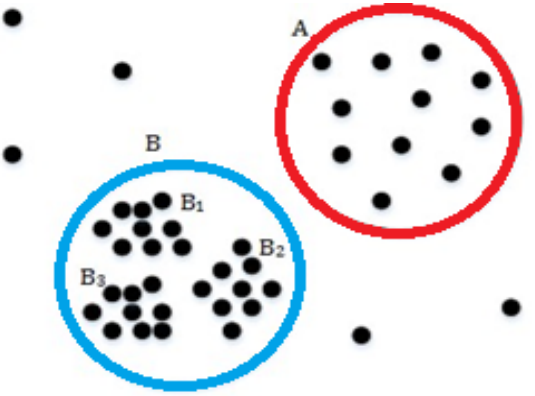


شکل ۲. خوشه‌بندی داده با اعمال الگوریتم DBSCAN

(Heidari et al., 2019)

از طرفی، اکثر روش‌های خوشه‌بندی موجود از جمله DBSCAN به پارامترهای ورودی نیاز دارند و انتخاب دقیق مقادیر این پارامترها روی خروجی الگوریتم بسیار تأثیرگذار است. هرچند برخی از الگوریتم‌ها کاربر را در انتخاب پارامتر صحیح کمک می‌کنند، اما در مجموعه‌داده‌هایی با حجم و ابعاد بالا، انتخاب دقیق این پارامترها مشکل‌آفرین است. الگوریتم VDBSCAN و بهبودهای انجام‌شده روی آن تلاش کرده‌اند که مقادیر این پارامترها را به‌طور خودکار تعیین کنند و تا حدی نیز در این کار موفق بوده‌اند، اما با پیاده‌سازی و انجام آزمایش‌های متعدد روی مجموعه‌داده‌های مختلف، به این نتیجه رسیدیم که الگوریتم VDBSCAN فقط روی مجموعه‌داده‌هایی قادر به تشخیص پارامترهاست که منحنی ملایم در k-dist plot مربوطه خود را نداشته باشند.

یکی از خصوصیت‌های مهم مجموعه‌داده‌های مختلف این است که خوشه‌های موجود در این مجموعه‌داده‌ها به‌دلیل وجود چگالی‌های محلی متفاوت، فقط با یک تنظیم پارامتر سراسری تشخیص داده می‌شوند، بنابراین، به بیش از یک چگالی محلی برای تشخیص خوشه‌ها نیاز است. برای نمونه، در مجموعه‌داده‌هایی که در شکل ۳ نشان داده شده است، خوشه‌های A، B_1، B_2 و B_3 را نمی‌توان فقط با یک تنظیم پارامتر سراسری تشخیص داد. اگر پارامترها را مطابق با چگالی محلی خوشه‌ها تنظیم کنیم، خوشه A به‌عنوان نویز محسوب می‌شود. اگر پارامترها مطابق با چگالی محلی خوشه A تنظیم شوند، خوشه‌های B_1، B_2 و B_3 به‌اشتباه با یکدیگر ترکیب می‌شوند. بنابراین، با یک تنظیم پارامتر سراسری نمی‌توان خوشه‌ها را به‌درستی تشخیص داد.



شکل ۳. خوشه‌هایی با چگالی متفاوت

یکی از مشکلاتی که ممکن است الگوریتم DBSCAN در مواجهه با مجموعه‌داده‌های حجیم با آن روبه‌رو شود، بحث پیچیدگی زمانی بالای این الگوریتم است. الگوریتم DBSCAN به‌ازای کلیه نقاط موجود در پایگاه‌داده، عمل پرس‌وجوی ناحیه‌ای را انجام می‌دهد. در پایگاه‌داده‌های بزرگ، زمان انجام این عمل شایان توجه خواهد بود و در نتیجه، کارایی الگوریتم تنزل می‌یابد. همچنین، الگوریتم DBSCAN زمانی که خوشه‌ها نزدیک به یکدیگر باشند، ممکن است در تشخیص صحیح خوشه‌ها با مشکل مواجه شود.

همان‌طور که بیان شد، یکی از مشکلات الگوریتم DBSCAN پشتیبانی نکردن از تغییرات چگالی داخل خوشه‌ها است. برای غلبه بر این مشکل، در روش پیشنهادی، ابتدا مقادیر مختلف پارامتر Eps محاسبه می‌شود، سپس، الگوریتم DBSCAN برای اعمال روی مجموعه‌داده با پارامترهای به‌دست‌آمده، تطبیق داده می‌شود. الگوریتم پیشنهادی بر مبنای مفهوم چگالی محلی نقاط عمل می‌کند. چگالی یک نقطه می‌تواند از طریق شمارش تعداد نقاط موجود در یک شعاع مشخص از آن نقطه محاسبه شود، اما این روش تقریب خوبی از چگالی نقاط را حاصل نمی‌کند. همان‌طور که در شکل ۴ مشاهده می‌شود، در

صورتی که تعداد نقاط موجود در شعاع α به‌عنوان چگالی نقاط در نظر گرفته شود، دو نقطـه p و q دارای چگالی مشابه هستند، اما همان‌طور که مشاهده می‌شـود، نقطـه q دارای تـراکم بـالاتری اسـت. بنـابراین، استفاده از این روش تقریب خوبی از چگالی نقاط را حاصل نمی‌کند. این در حالی است که روش دقیق‌تـر، روشی است که چگالی نقاط را براساس فاصله نقاط از همسایه‌های آنها محاسبه کند.

(ب) (الف)

شکل ۴. چگالی مبتنی بر شمردن نقاط موجود در شعاع α

با فرض داشتن مجموعه‌داده D، چگالی محلی شیء $x \in D$ از طریق محاسبه مجموع فاصـله اقلیدسـی شیء x از L نزدیک‌ترین همسایه آن طبق رابطه ۱ به دست می‌آید (تـابع چگـالی محلـی) (Weng et al., 2021):

$$Local\ Density(x) = \sum_{i=1}^{L} d(x.xi) \qquad \text{رابطه ۱)}$$

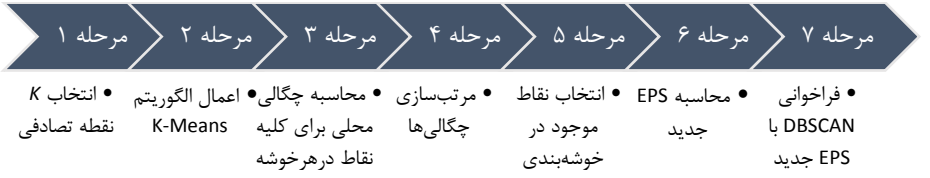
که در آن، x_i معادل با iامین نزدیک‌ترین همسایه از شیء x است و همچنین، $d(x,x_i)$ فاصله اقلیدسـی بین دو شیء را برمی‌گرداند. در این تعریف، تصمیم‌گیری در خصوص مقدار L بسیار مهم است، به‌گونـه‌ای که انتخاب نادرست آن به تنزل دقت نتایج خوشه‌بندی منجر می‌شود. به دو دلیـل مقـدار L را نمـی‌تـوان بزرگ در نظر گرفت. نخست، همان‌گونه که در بخش ارزیابی نتایج آزمایش نشان داده شده است، در نظـر گرفتن مقادیر بزرگ برای L تقریب مناسبی از چگالی نقاط را حاصل نمی‌کند. همچنین، با توجه به اینکـه بخش غالب پیچیدگی زمانی الگوریتم پیشنهادی مربوط به محاسبه چگالی محلی نقاط است، هرچه مقدار L بزرگ‌تر باشد، پیچیدگی زمانی الگوریتم نیز افزایش می‌یابد، به‌طوری که به‌ازای $L=n$ پیچیـدگی زمـانی الگوریتم از مرتبه $O(n^2 \log n)$ می‌شـود. عـلاوه بـر پـارامتر L، الگـوریتم پیشـنهادی بـه دو پـارامتر ورودی Minpts و k نیز نیاز دارد. پارامتر Minpts حداقل تعداد نقاط موجود در یک خوشه را مشخص مـی‌کنـد و پارامتر k نیز برای محاسبه مقادیر Eps استفاده می‌شود.

روال کار الگوریتم پیشنهادی به این صورت است که ابتدا، چگالی محلی کلیه نقاط طبق تعریف تـابع چگالی محلی محاسبه شده، سپس، نقاط براساس چگالی محلی خود به‌صـورت نزولـی مرتـب مـی‌شـوند. شایان ذکر است که نقطه با مقدار چگالی محلی کمتر، از چگالی بیشتری برخوردار اسـت. سـپس، از بـین

مجموعه نقاطی که هنوز خوشه‌بندی نشده‌اند، متراکم‌ترین نقطه (مانند p) انتخاب می‌شود. هدف مقاله این است که خوشه با تراکم بالاتر، زودتر تشخیص داده شود. در واقع، با توجه به اینکه در مجموعه‌داده‌هایی با چگالی متفاوت، برای هر چگالی یک مقدار Eps متفاوت وجود دارد، در صورتی که کار از خوشه‌هایی با چگالی کمتر آغاز شود، یعنی خوشه‌هایی که مقدار Eps آنها بزرگ است، خوشه‌هایی با چگالی بالا نیز ممکن است به‌اشتباه با خوشه‌هایی با چگالی پایین ترکیب شوند. به بیان دیگر، خوشه‌هایی با چگالی کمتر، خوشه‌هایی با چگالی بالاتر از خود را نیز شامل می‌شوند. بنابراین، کار همیشه از متراکم‌ترین نقاط آغاز می‌شود تا خوشه‌هایی با چگالی بالاتر زودتر تشخیص داده شوند. سپس، با صرف‌نظر کردن از این نقاط که خوشه‌بندی شده‌اند، از خوشه‌بندی شدن مکرر یک نقطه طی تکرارهای بعدی جلوگیری شده است. در تکرار iام بعد از انتخاب متراکم‌ترین نقطه p از بین نقاطی که هنوز خوشه‌بندی نشده‌اند، فاصله تا kامین نزدیک‌ترین همسایه از p به‌عنوان Epsi در نظر گرفته می‌شود.

بعد از آن، الگوریتم DBSCAN با پارامتر Minpts که از ورودی گرفته شده و پارامتر Epsi که در مرحله قبل محاسبه شده است، فراخوانی می‌شوند. بعد از پایان کار الگوریتم DBSCAN و صرف‌نظر کردن از نقاطی که خوشه‌بندی شده‌اند، از طریق برچسب خوشه‌ای که به نقاط داده می‌شود، طی یک فرایند تکراری از بین مجموعه نقاطی که هنوز خوشه‌بندی نشده‌اند، متراکم‌ترین نقطه انتخاب شده و فاصله آن نقطه تا kامین نزدیک‌ترین همسایه خود به‌عنوان پارامتر Epsi+1 در نظر گرفته می‌شود و الگوریتم DBSCAN با پارامتر جدید Epsi+1 فراخوانی می‌شود. این فرایند تا زمانی ادامه دارد که کلیه نقاط خوشه‌بندی شوند یا تعداد نقاط خوشه‌بندی‌نشده کمتر از مقدار Minpts باشند. اگر تعداد نقاط باقی‌مانده خوشه‌بندی‌نشده کمتر از Minpts شود، این نقاط برچسب نویز خواهند گرفت، زیرا در این حالت دیگر امکان تشکیل خوشه جدید نیست.

فرض کنید، r تعداد نقاط باقی‌مانده از مجموعه‌داده برای خوشه‌بندی بعد از فراخوانی چندین باره الگوریتم DBSCAN باشد. زمانی که $\text{Minpts} \leq r < k$ باشد، فاصله تا rامین نزدیک‌ترین همسایه از متراکم‌ترین نقطه خوشه‌بندی‌نشده به‌عنوان Eps جدید در نظر گرفته می‌شود. حال اگر $r < \text{Minpts}$ باشد، نقاط باقی‌مانده به‌عنوان نویز برچسب‌گذاری می‌شوند. مراحل اصلی الگوریتم پیشنهادی به‌طور خلاصه در شکل ۵ نشان داده شده است.



شکل ۵. مراحل اصلی الگوریتم پیشنهادی

یکی از خصوصیت‌های روش پیشنهادی این است که تعیین مقدار دقیق پارامترهای ورودی (پارامتر Eps به‌صورت خودکار تعیین می‌شود) ساده‌تر است. به‌طور کلی، الگوریتم ارائه‌شده با تعیین خودکار پارامتر Eps، حساسیت به پارامترهای ورودی را کاهش داده است. در نگاه نخست به نظر می‌رسد که الگوریتم سه پارامتر دارد و شاید چندان مناسب نباشد، اما بهترین مقدار برای پارامتر L همان مقدار Minpts است. یعنی برای داشتن نتایج ایده‌آل، مقدار پارامتر L را برابر با مقدار Minpts قرار داده و از این پارامتر صرف‌نظر می‌شود. بنابراین، برای این الگوریتم فقط تعیین مقدار دو پارامتر Minpts و k کافی است و برای فهم بهتر و دقیق الگوریتم است که پارامتر L به‌صورت یک پارامتر جداگانه در نظر گرفته شده است. در رابطه با پارامتر k نیز گفته می‌شود که اولاً همان‌گونه که اشاره شد، حد پایین این پارامتر مشخص است و این پارامتر باید حداقل به‌اندازه Minpts باشد، ثانیاً مقدار این پارامتر در یک طیف وسیع تغییرپذیر است (برخلاف پارامتر Eps)، بدون اینکه در نتایج الگوریتم تغییری حاصل شود و این به‌معنای کاهش حساسیت به پارامتر ورودی است. این کاهش حساسیت باعث انتخاب سریع‌تر و ساده‌تر مقدار دقیق پارامترها می‌شود، به‌گونه‌ای که در ارزیابی‌های انجام‌شده روی نتایج آزمایش‌ها، فقط با تعداد معدودی آزمون روی مجموعه‌داده‌ها نتایج مطلوبی به دست آمده است.

به‌علاوه، یکی از خصوصیت‌های بارز الگوریتم DBSCAN سادگی و درک‌پذیر بودن آن است. در الگوریتم پیشنهادشده، این سادگی و درک‌پذیر بودن به‌خوبی رعایت شده است، بنابراین، نقاط نویز ممکن است طی فراخوانی‌های بعدی الگوریتم DBSCAN برچسب خوشه بگیرند.

الگوریتم پیشنهادی با استفاده از شاخص مکانی KD-Tree پیاده‌سازی شده است. تابع K-نزدیک‌ترین همسایه[1] با گرفتن یک نقطه به‌عنوان ورودی، kامین نزدیک‌ترین همسایه آن نقطه را برمی‌گرداند. تابع Distance نیز فاصله بین دو شیء ورودی‌اش را محاسبه می‌کند. همان‌طور که اشاره شد، تعیین پارامترهای الگوریتم پیشنهادی در مقایسه با سایر الگوریتم‌ها ساده‌تر است. پارامتر Minpts که حداقل تعداد نقاط یک خوشه را مشخص می‌کند، بسته به مجموعه‌داده بررسی‌شده قابل تعیین است.

بدیهی است که پارامتر k حداقل باید به‌اندازه Minpts باشد، زیرا در غیر این صورت ممکن است خوشه‌ای یافت نشود. به بیان دیگر، پارامتر k باید به‌گونه‌ای باشد که در فاصله بین نقطه بررسی‌شده و kامین نزدیک‌ترین همسایه‌اش یک خوشه تشکیل شود، یعنی حداقل Minpts نقطه در این فاصله قرار گرفته باشد. بنابراین، اگر پارامتر k حداقل به‌اندازه Minpts باشد، این شرط برقرار خواهد بود. در رابطه با پارامتر L نیز همان‌گونه که در آزمایش‌ها نشان داده شده است، می‌توان این پارامتر را با مقدار ثابت Minpts تنظیم کرد.

ارزیابی نتایج آزمایش

در این بخش، ابتدا مجموعه‌داده آزمون تشریح شده، سپس، تنظیمات سخت‌افزاری و شبیه‌سازی آزمایش و در بخش نهایی، ارزیابی نتایج تشریح شده است. پیاده‌سازی الگوریتم پیشنهادی و الگوریتم‌های

1. K-nearest neighbors algorithm

مقایسه‌شده، روی رایانه شخصی با مشخصات پردازنده سه‌هسته‌ای اینتل، حافظه ۴ گیگابایت و سیستم عامل ویندوز ۱۰، و در محیط نرم‌افزاری MATLAB-R2020b انجام شده است. در شبیه‌سازی آزمایش‌ها، از ۴ مجموعه‌داده استاندارد استفاده شده است شامل Image segmentation با ۲۳۱۰ نمونه، ۷ کلاس و ۱۹ ویژگی، Pendigit با ۱۰۹۹۲ نمونه، ۴۴ کلاس و ۱۶ ویژگی، Letters با ۲۰۰۰۰ نمونه، ۴۰ کلاس و ۱۶ ویژگی و درنهایت، Shuttle control با ۵۸۰۰۰ نمونه، ۲ کلاس و ۱۵ ویژگی. این ۴ مجموعه‌داده از پایگاه داده‌های UCI[1] برداشته شده است.

در کلیه آزمایش‌ها، از روش اعتبارسنجی K-fold با K=10 استفاده شده و نتایج به‌صورت میانگین محاسبه و نشان داده شده است. برای ارزیابی سرعت از معیار ثانیه و برای دقت دسته‌بندی از معیار خطا MSE استفاده شده است که مطابق رابطه ۲ محاسبه شده است.

$$MSE = \frac{1}{n}\sum_{i=1}^{n} \left(Y_i - \widehat{Y_i}\right)^2 \quad \text{رابطه ۲)}$$

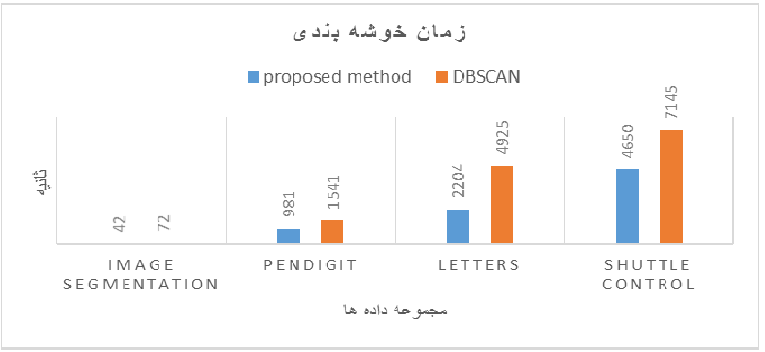
که درآن، پارامتر Y_i خروجی واقعی، $\widehat{Y_i}$ خروجی الگوریتم است و n تعداد نمونه‌ها است. نتایج تجربی آزمایش، به‌منظور ارزیابی عملکرد الگوریتم پیشنهادی و الگوریتم استاندارد DBSCAN مقایسه شده است. در واقع، در این مقایسه پس از خوشه‌بندی و برچسب‌گذاری داده‌ها در خوشه‌های مختلف، میزان صحیح برچسب‌گذاری‌شده‌ها با توجه به مجموعه‌داده‌ها مقایسه شده و میزان خطا محاسبه شده است. در جدول ۱، زمان اجرای خوشه‌بندی و میزان خطا در آزمایش روش پیشنهادی و الگوریتم استاندارد DBSCAN مقایسه شده است.

جدول ۱. مقایسه نتایج الگوریتم پیشنهادی

مجموعه‌داده	روش پیشنهادی DBSCAN		روش استاندارد DBSCAN	
	زمان (S)	درصدخطا (MSE)	زمان (S)	درصدخطا (MSE)
Image segmentation	۴۲	۰/۰۱۲	۷۲	۰/۰۴۶
Pendigit	۹۸۱	۰/۰۱۹	۱۵۴۱	۰/۰۷۹
Letters	۲۲۰۴	۰/۱۳۴	۴۹۲۵	۰/۳۰۱
Shuttle control	۴۶۵۰	۱/۰۸۴	۷۱۴۵	۱/۹۱۸

در شکل ۶، میزان زمان اجرای خوشه‌بندی برای مجموعه‌داده‌های آزمایش‌شده نشان داده شده است که نشان می‌دهد در خوشه‌بندی پیشنهادی، زمان اجرای خوشه‌بندی کاهش یافته و هرچه مجموعه‌داده حجیم‌تر باشد، کاهش زمان اجرای خوشه‌بندی محسوس‌تر است.

1. https://archive.ics.uci.edu/ml/index.php



شکل ۶. مقایسه زمان خوشه‌بندی

در شکل ۷، میزان خطای آزمایش برای مجموعه‌داده‌ها نشان داده شده است. در خوشه‌بندی پیشنهادی، میزان معیار خطا کاهش یافته، زیرا برچسب داده‌ها متناسب با کلاس داده‌ها، با تعداد بیشتر بوده است.

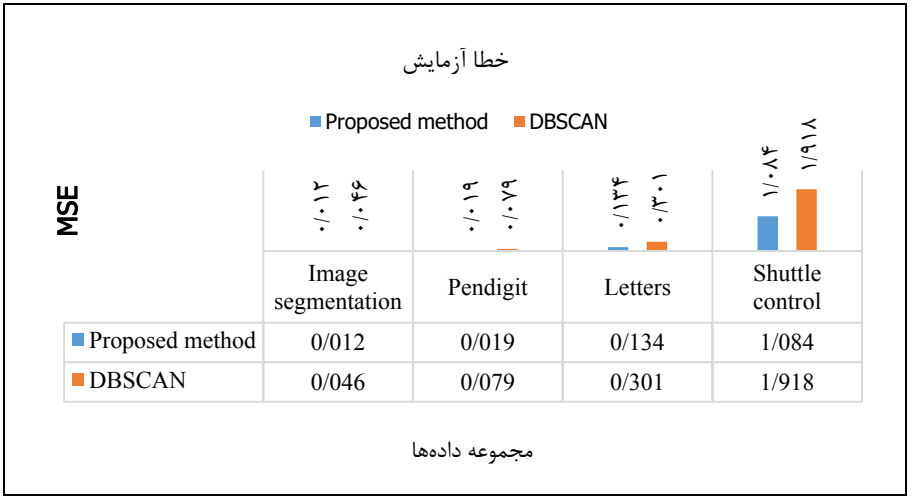


	Image segmentation	Pendigit	Letters	Shuttle control
Proposed method	0/012	0/019	0/134	1/084
DBSCAN	0/046	0/079	0/301	1/918

شکل ۷. مقایسه خطای آزمایش

مقایسه بین الگوریتم‌های پیشنهادی و بهبودیافته خوشه‌بندی DBSCAN شامل دو روش K-DBSCAN و H-DBSCAN که روش پیشنهادی هم برگرفته از آنها است، در جدول ۲ نشان می‌دهد که روش پیشنهادی در مجموعه‌داده‌های مشترک توانسته است که معیارهای زمان اجرای خوشه‌بندی و دقت مناسبی را به‌همراه داشته باشد.

جدول ۲. مقایسه نتایج الگوریتم پیشنهادی با سایر روش‌ها

مجموعه‌داده	روش پیشنهادی DBSCAN		روش K-DBSCAN		روش h-DBSCAN	
	زمان (S)	درصد خطا (MSE)	زمان (S)	درصد خطا (MSE)	زمان (S)	درصد خطا (MSE)
Image segmentation	۰/۰۱۲	۴۲	۰/۰۱۳	۴۱	۰/۰۱۱	۵۳
Pendigit	۰/۰۱۹	۹۸۱	۰/۰۲۵	۹۷۵	۰/۰۱۷	۱۰۸۱
Letters	۰/۱۳۴	۲۲۰۴	۰/۱۴۶	۲۱۱	۰/۱۲۹	۲۴۸۰
Shuttle control	۱/۰۷۱	۴۶۵۰	۱/۱۱۲	۲۴۸۰	۱/۰۷۱	۴۹۹۱

در جدول ۲، زمان اجرای خوشه‌بندی در روش پیشنهادی با دو روش K-DBSCAN و H-DBSCAN مقایسه شده است و نشان می‌دهد که خوشه‌بندی در روش K-DBSCAN با زمان کمتری در مقایسه با سایر روش‌های خوشه‌بندی انجام شده است. معیار خطای خوشه‌بندی در روش پیشنهادی با دو روش K-DBSCAN و H-DBSCAN مقایسه شده که نشان می‌دهد خوشه‌بندی در روش H-DBSCAN با خطای کمتری در مقایسه با سایر روش‌های دیگر خوشه‌بندی انجام شده است.

همان‌طور که از نتایج تجربی جدول ۲ مشخص است، روش پیشنهادی نتایج زمان اجرای خوشه‌بندی و خطای آزمایش کمتی از روش H-DBSCAN دارد. در واقع، با توجه به اینکه از خاصیت هر دو روش استفاده می‌کند، نتایج آن حد متوسطی دارد. در حالتی که هر دو معیار زمان اجرای خوشه‌بندی و دقت هم‌زمان مد نظر باشند، روش پیشنهادی می‌تواند روش مطلوب‌تری باشد.

نتیجه‌گیری و کارهای آتی

همان‌طور که در شبیه‌سازی‌های روش پیشنهادی مشاهده شد، این روش توانسته در مقایسه با خوشه‌بندی استاندارد DBSCAN به‌طور مطلوب‌تری عمل کند و خوشه‌بندی در زمان کمتر و دقت بالاتری در آزمایش‌ها داشته باشد. براساس ارزیابی نتایج آزمایش‌ها، روی مجموعه‌داده Image segmentation، الگوریتم پیشنهادی در ۴۲ ثانیه زمان خوشه‌بندی داشت، در صورتی که در مقایسه با خوشه‌بندی استاندارد DBSCAN میزان ۲۰ ثانیه کمتر و در مقایسه با الگوریتم خوشه‌بندی H-DBSCAN میزان ۱۱ ثانیه کمتر بود. به‌علاوه، خطای الگوریتم پیشنهادی به‌اندازه ۰/۰۳۴ در مقایسه با خوشه‌بندی استاندارد DBSCAN و در مقایسه با الگوریتم خوشه‌بندی K-DBSCAN میزان ۰/۰۰۱ کمتر بوده است. همچنین، روی مجموعه‌داده Pendigit، الگوریتم پیشنهادی نیاز به ۹۸۱ ثانیه خوشه‌بندی داشت که این در مقایسه با الگوریتم خوشه‌بندی H-DBSCAN میزان ۱۰۰ ثانیه کمتر بوده است.

روی مجموعه‌داده‌های Letters و Shuttle control که تعداد نمونه‌های بیشتری دارد، نتایج اختلاف بیشتری را نشان می‌دهد، به‌طوری که روی مجموعه‌داده Letters در الگوریتم پیشنهادی، زمان خوشه‌بندی ۲۲۰۴ ثانیه بود، اما در خوشه‌بندی استاندارد DBSCAN زمان بیشتر از دو برابر بوده و

همچنین، در مقایسه با الگوریتم خوشه‌بندی H-DBSCAN نیز زمان خوشه‌بندی کمتری داشته است. میزان خطا روی مجموعه‌داده Shuttle control در روش پیشنهادی حدود ۰/۸۳۴ کمتر از خوشه‌بندی استاندارد DBSCAN است. به‌علاوه، در الگوریتم پیشنهادی زمان لازم برای خوشه‌بندی ۴۶۵۰ ثانیه بود، اما در خوشه‌بندی استاندارد DBSCAN زمان ۷۱۴۵ ثانیه است. میزان خطا در این مجموعه‌داده در روش پیشنهادی حدود ۰/۰۲۸ در مقایسه با الگوریتم K-DBSCAN کمتر بوده است.

ضعف الگوریتم پیشنهادی، در زمان خوشه‌بندی در مقایسه با الگوریتم K-DBSCAN روی مجموعه‌داده Image segmentation، مقدار ۱ ثانیه، روی مجموعه‌دادهPendigit مقدار ۶ ثانیه، روی مجموعه‌داده Letters مقدار ۹۳ ثانیه و روی مجموعه‌داده Shuttle control مقدار ۴۸۵ ثانیه بود. به‌علاوه، خطای آزمایش در مقایسه با الگوریتم H-DBSCAN روی مجموعه‌داده Image segmentation مقدار ۰/۰۰۱، روی مجموعه‌داده Pendigit مقدار ۰/۰۰۲، روی مجموعه‌داده Letters مقدار ۰/۰۰۵ و روی مجموعه‌داده Shuttle control مقدار ۰/۰۱۳ بوده است. به‌طور کلی، می‌توان گفت که الگوریتم پیشنهادی، در معیار دقت در مقایسه با الگوریتم H-DBSCAN، ضعف کمتری دارد، اما از طرفی در مقایسه با آن، زمان خوشه‌بندی مناسب‌تری دارد.

فهرست منابع

Ankerst, M., Breunig, M. M., Kriegel, H.P., & Sander, J. (1999). OPTICS. *ACM SIGMOD Record,* 28(2), 49–60.

Bechini, A., Criscione, M., Ducange, P., Marcelloni, F., & Renda, A. (2020). FDBSCAN-APT: A Fuzzy Density-based Clustering Algorithm with Automatic Parameter Tuning. *2020 IEEE International Conference on Fuzzy Systems* (FUZZ-IEEE).

Botts, C. H. (2021). A Novel Metric for Detecting Anomalous Ship Behavior Using a Variation of the DBSCAN Clustering Algorithm. *SN Computer Science*, 2(5).

Chen, Y., Tang, S., Bouguila, N., Wang, C., Du, J., & Li, H. (2018). A fast clustering algorithm based on pruning unnecessary distance computations in DBSCAN for high-dimensional data. *Pattern Recognition*, 83, 375–387.

Chen, Y., Zhou, L., Bouguila, N., Wang, C., Chen, Y., & Du, J. (2020). BLOCK-DBSCAN: Fast Clustering For Large Scale Data. *Pattern Recognition*, 107624.

Chimwayi, K. B., & Anuradha, J. (2018). Clustering West Nile Virus Spatio-temporal data using ST-DBSCAN. *Procedia Computer Science*, 132, 1218–1227.

De Moura Ventorim, I., Luchi, D., Rodrigues, A. L., & Varejão, F. M. (2021). BIRCHSCAN: A sampling method for applying DBSCAN to large datasets. *Expert Systems with Applications*, 184, 115518.

Deng, D. (2020). DBSCAN Clustering Algorithm Based on Density. *2020 7th International Forum on Electrical Engineering and Automation* (IFEEA).

Galán, S. F. (2019). Comparative evaluation of region query strategies for DBSCAN clustering. *Information Sciences*, 502, 76–90.

Gholizadeh, N., Saadatfar, H., & Hanafi, N. (2021). K-DBSCAN: An improved DBSCAN algorithm for big data. *The Journal of supercomputing*, 77, 6214-6235.

Hartmann, S., Ma, H., Hameurlain, A., Pernul, G., & Wagner, R. R. (Eds.). (2018). *Database and Expert Systems Applications*. Lecture Notes in Computer Science.

Heidari, S., Alborzi, M., Radfar, R., Afsharkazemi, M. A., & Rajabzadeh Ghatari, A. (2019). Big data clustering with varied density based on MapReduce. *Journal of Big Data*, 6, 1-16.

Kim, J. H., Choi, J.H., Yoo, K. H., & Nasridinov, A. (2018). AA-DBSCAN: an approximate adaptive DBSCAN for finding clusters with varying densities. *The Journal of Supercomputing,* 75(1), 142- 169.

Kuo, F. Y., Wen, T.H., & Sabel, C. E. (2018). Characterizing Diffusion Dynamics of Disease Clustering: A Modified Space–Time DBSCAN (MST-DBSCAN) Algorithm. *Annals of the American Association of Geographers,* 108(4), 1168–1186.

Lai, W., Zhou, M., Hu, F., Bian, K., & Song, Q. (2019). A New DBSCAN Parameters Determination Method Based on Improved MVO. *IEEE Access*, 7, 104085–104095.

Li, S. (2020). An Improved DBSCAN Algorithm Based on the Neighbor Similarity and Fast Nearest Neighbor Query. *IEEE Access*, 1–1.

Louhichi, S., Gzara, M., & Abdallah, H. B. (2018). Skin Lesion Segmentation Using Multiple Density Clustering Algorithm MDCUT And Region Growing. *2018 IEEE/ACIS 17th International Conference on Computer and Information Science* (ICIS).

Louhichi, S., Gzara, M., & Ben-Abdallah, H. (2018). MDCUT2: a multi-density clustering algorithm with automatic detection of density variation in data with noise. *Distributed and Parallel Databases*, 37, 73-99.

Pavlis, M., Dolega, L., & Singleton, A. (2017). A Modified DBSCAN Clustering Method to Estimate Retail Center Extent. *Geographical Analysis,* 50(2), 141–161.

Sabor, K., Jougnot, D., Guerin, R., Steck, B., Henault, J.M., Apffel, L., & Vautrin, D. (2021). A data mining approach for improved interpretation of ERT inverted sections using the DBSCAN clustering algorithm. *Geophysical Journal International*, 225(2), 1304–1318.

Sharma, A., & Upadhyay, D. (2018). VDBSCAN clustering with map-reduce technique. In Recent Findings in Intelligent Computing Techniques: *Proceedings of the 5th ICACNI 2017,* Volume 2 (pp. 305-314). Springer Singapore.

Sheridan, K., Puranik, T. G., Mangortey, E., Pinon-Fischer, O. J., Kirby, M., & Mavris, D. N. (2020). An application of dbscan clustering for flight anomaly detection during the approach phase. *In AIAA Scitech 2020 Forum* (p. 1851).

Wang, Q., Wang, Z., Zhang, L., Liu, P., & Zhang, Z. (2020). A novel consistency evaluation method for series-connected battery systems based on real-world operation data. *IEEE Transactions on Transportation Electrification,* 7(2), 437-451.

Wang, Y., Gu, Y., & Shun, J. (2020, June). Theoretically-efficient and practical parallel DBSCAN. *In Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data* (pp. 2555-2571).

Weng, S., Gou, J. & Fan, Z. (2021). h-DBSCAN: A simple fast DBSCAN algorithm for big data. *Proceedings of The 13th Asian Conference on Machine Learning*, PMLR 157:81-96, 2021.

Wu, X., Cheng, C., Zurita-Milla, R., & Song, C. (2020). An overview of clustering methods for geo-referenced time series: from one-way clustering to co- and tri-clustering. *International Journal of Geographical Information Science*, 1–27.

Yu, X., Zeng, F., Mwakapesa, D. S., Nanehkaran, Y. A., Mao, Y. -M., Xu, K. -B., & Chen, Z. -G. (2021). DBWGIE-MR: A density-based clustering algorithm by using the weighted grid and information entropy based on MapReduce. *Journal of Intelligent & Fuzzy Systems,* 40(6), 10781–10796.

An Extended Density-based Clustering Algorithm in Big Data

Reza Ghaemi [*1]
Assistant Prof., Department of Computer Engineering, Quchan Branch, Islamic Azad University, Quchan, Iran

Yaghoob Arad
Ph.D. Candidate, Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Neyshabour, Iran

Fereshteh Hajghazi
Ph.D. Candidate, Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Neyshabour, Iran

Abstract
Today, data generation through smart equipment, including mobile phones, has faced a significant growth, and clustering is one of the most widely used knowledge discovery techniques in big data. Density-based clustering (DBSCAN) is one of the most efficient clustering algorithms in data mining, and despite having advantages, it also has problems, such as the difficulty in determining the input parameters, as well as not being able to detect clusters. with different densities. In the proposed algorithm of this article, it is inspired by the K-DBSCAN algorithm in grouping large data with the aim of reducing the clustering execution time.In addition, by using K-Means and H-DBSCAN algorithms, different densities of the data set were identified and an Eps radius was determined for each density, and then, the proposed density-based clustering algorithm was developed with parameters The matching is applied to the data, and in fact, the innovation of this article is the use of K Means clustering and the estimation of different densities in the DBSCAN clustering method. The proposed algorithm has been compared with the simple DBSCAN clustering algorithm and two developed K-DBSCAN and H-DBSCAN algorithms on four standard data sets: Image segmentation, Pendigit, Letters and Shuttle control. The results show that the proposed algorithm is superior to other algorithms when both time and accuracy are criteria in clustering.

Keywords: Big data, Clustering, DBSCAN, K-DBSCAN, H-DBSCAN, K-Means.

1. Corresponding Author: r.ghaemi@iauq.ac.ir