بهبود الگوریتم خوشه‌بندی مبتنی بر چگالی با استفاده از اصلاح تعاریف چگالی و پارامتر ورودی

علیرضا پهلوانزاده* و علی اکبر نیک نفس

بخش مهندسی کامپیوتر، دانشگاه شهید باهنر کرمان، کرمان، ایران

چکیده

خوشه‌بندی مبتنی بر چگالی یکی از روش‌های مورد توجه در داده‌کاوی و نمونه‌ای پرکاربرد از این روش است. DBSCAN علاوه‌بر مزایای خود معایبی نیز دارد. به‌عنوان نمونه، تعیین پارامترهای ورودی این الگوریتم توسط کاربر کار مشکلی است. در مقالهٔ حاضر سعی شده است، اصلاحاتی روی یکی از الگوریتم‌های مبتنی برچگالی به نام ISB-DBSCAN انجام شود. در روش پیشنهادی همانند ISB-DBSCAN از یک پارامتر ورودی k به‌عنوان تعداد نزدیک‌ترین همسایه استفاده شده است. از آنجا که تعیین پارامتر k ممکن است، برای کاربر مشکل باشد، یک روش پیشنهادی با الگوریتم ژنتیک برای تعیین خودکار k نیز ارائه شده است. برای ارزیابی روش‌های پیشنهادی آزمایش‌هایی روی یازده مجموعه دادهٔ استاندارد انجام شد و دقت خوشه‌بندی در روش‌ها مورد ارزیابی قرار گرفت. نتایج به‌دست‌آمده در مقایسه با دیگر روش‌های موجود نشان داد که روش پیشنهادی در مجموعه‌داده‌های مختلف، نتایج بهتری را کسب کرده است.

واژگان کلیدی: خوشه‌بندی مبتنی بر چگالی، پارامتر همسایگی، خوشه‌بندی با چگالی متفاوت.

Improvement of density-based clustering algorithm using modifying the density definitions and input parameter

Alireza Pahlevanzadeh* & Aliakbar Niknafs

Department of Computer Engineering, Shahid Bahonar University of Kerman, Kerman, Iran

Abstract

Clustering is one of the main tasks in data mining, which means grouping similar samples. In general, there is a wide variety of clustering algorithms. One of these categories is density-based clustering. Various algorithms have been proposed for this method; one of the most widely used algorithms called DBSCAN. DBSCAN can identify clusters of different shapes in the dataset and automatically identify the number of clusters. There are advantages and disadvantages in this algorithm. It is difficult to determine the input parameters of this algorithm by the user. Also, this algorithm is unable to detect clusters with different densities in the data set. ISB-DBSCAN algorithm is another example of density-based algorithms that eliminates the disadvantages of the DBSCAN algorithm. ISB-DBSCAN algorithm reduces the input parameters of DBSCAN algorithm and uses an input parameter k as the nearest neighbor's number. This method is also able to identify different density clusters, but according to the definition of the new core point, It is not able to identify some clusters in a different data set.

This paper presents a method for improving ISB-DBSCAN algorithm. A proposed approach, such as ISB-DBSCAN, uses an input parameter k as the number of nearest neighbors and provides a new definition for core point. This method performs clustering in three steps, with the difference that, unlike ISB-DBSCAN algorithm, it can create a new cluster in the final stage. In the proposed method, a new criterion, such as the number of dataset dimensions used to detect noise in the used data set. Since the determination of the k parameter in the proposed method may be difficult for the user, a new method with genetic algorithm is also

* Corresponding author

*نویسندهٔ عهده‌دار مکاتبات

● تاریخ ارسال: ۹۶/۱۱/۱۹ ● تاریخ پذیرش: ۹۸/۰۳/۲۹ ● نوع مطالعه: پژوهشی

proposed for the automatic estimation of the k parameter. To evaluate the proposed methods, tests were carried out on 11 standard data sets and the accuracy of clustering in the methods was evaluated. The results showe that the proposed method is able to achieve better results in different data sets compare to other available methods. In the proposed method, the automatic determination of k parameter also obtained acceptable results.

Keywords: Density-based clustering, neighborhood parameter, clustering with different density

۱-مقدمه

خوشه‌بندی یکی از مهم‌ترین مسائل در یادگیری بدون نظارت[1] است. در یادگیری بدون نظارت داده‌ها بدون برچسب هستند. هدف از خوشه‌بندی یافتن یک ساختار معنادار درون یک مجموعه‌دادهٔ بدون برچسب است. درواقع در خوشه‌بندی سعی می‌شود یک مجموعهٔ بدون برچسب، به گروه‌ها و دسته‌هایی مشابه تقسیم شود، به‌طوری‌که داده‌های موجود در یک دسته بیشینه شباهت و داده‌های موجود در دسته‌های متفاوت کمینه شباهت را داشته باشند.

خوشه‌بندی به‌طور گسترده در بسیاری از حوزه‌ها کاربرد دارد؛ ازجمله آنها می‌توان به کاربرد خوشه‌بندی در حوزه پردازش تصویر جهت قطعه‌بندی تصاویر[2]، در حوزه شبکه‌های حس‌گر بی‌سیم [3]جهت گروه‌بندی حس‌گرها به‌منظور افزایش طول عمر[4]، در علوم کامپیوتر[5] جهت وب‌کاوی[6]، در علم پزشکی جهت تشخیص بیماری‌ها مانند تشخیص پیوسته میزان استرس در طول رانندگی [14]، در اقتصاد جهت گروه‌بندی مشتریان و بسیاری کاربردهای دیگر نام برد.

روش‌های خوشه‌بندی معمول همچونk میانگین [10][7] که از قبل وجود دارند، مبتنی بر فاصله بوده و خوشه‌های دایروی ایجاد می‌کنند. این روش‌ها برای یافتن خوشه‌هایی با اشکال غیر از دایره، ممکن است به مشکل برخورده و دچار اشتباه شوند. روش‌های مبتنی بر چگالی همان‌طور که از عنوان آنها نیز مشخص است، براساس چگالی داده‌ها عمل می‌کنند. به عبارت دیگر تا زمانی که چگالی داده‌ها از یک حد مشخص بیشتر باشد، خوشه گسترش پیدا می‌کند. این بدین معناست که اگر داده یا داده‌هایی به‌صورت جداگانه و به دور از سایر داده‌ها باشند، به‌راحتی مشخص می‌شوند؛ لذا این روش برای پیداکردن و حذف داده‌های پرت مناسب است. در این روش‌ها هر داده با توجه به تعداد داده‌های موجود در همسایگی آن دسته‌بندی می‌شود.

الگوریتم DBSCAN [8][6] یک روش خوشه‌بندی مبتنی بر چگالی است. برتری این روش نسبت به روش‌های دیگر خوشه‌بندی مانند الگوریتم خوشه‌بندی K میانگین، این است که به شکل داده‌ها حساس نیست و می‌تواند اشکال غیر منظم را نیز در داده‌ها تشخیص دهد. اگرچه DBSCAN می‌تواند نمونه‌ها را با پارامترهای ورودی داده‌شده مانند Eps (بیشینه شعاع همسایگی) و MinPts (کمینه تعداد نقاط لازم در همسایگی) خوشه‌بندی کند، اما کاربران را به‌واسطهٔ مسؤلیت انتخاب مقادیر پارامترهایی که منجر به کشف خوشه‌های قابل قبول می‌شود، به زحمت می‌اندازد. این مشکل در بسیاری از الگوریتم‌های خوشه‌بندی وجود دارد. چنین پارامترهایی به‌طورمعمول به‌صورت تجربی تنظیم و به‌سختی تعیین می‌شوند. اغلب الگوریتم‌های خوشه‌بندی به مقادیر این پارامترها حساس هستند و تنظیمات متفاوتِ اندکی ممکن است، منجر به خوشه‌بندی متفاوتی از داده‌ها شود. به‌علاوه مجموعه داده‌های جهان واقعی و دارای بُعد بالا اغلب دارای توزیع‌های جهت‌داری هستند که ساختار خوشه‌بندی ذاتی آن‌ها ممکن است، به‌خوبی توسط یک مجموعهٔ واحد از پارامترهای چگالی عام مشخص نشود. همچنین، این روش خوشه‌بندی، توانایی شناخت خوشه‌هایی با چگالی متفاوت در مجموعهٔ داده را ندارد.

در این مقاله یک روش پیشنهادی برای بهبود الگوریتم ISB-DBSCAN [9][12] ارائه شده است. در روش ISB-DBSCAN از پارامتر ورودی k به‌عنوان تعداد نزدیک‌ترین همسایهٔ هر داده استفاده می‌شود. همچنین این روش در شناخت خوشه‌هایی با چگالی متفاوت کاراست؛ اما در مجموعه داده‌های مختلف به‌دلیل تعیین تعریف دادهٔ هستهٔ پیشنهادشده قادر به یافتن خوشه‌های درستی نیست.

روش پیشنهادی با استفاده از تعریف یک مقدار جدید برای تعیین دادهٔ هسته و همچنین تغییر در الگوریتم ISB-DBSCAN فعلی ایجاد شده است. همچنین از آنجا که در روش پیشنهای نیز از پارامتر ورودی k استفاده شده، ممکن

[1] Unsupervised learning
[2] Image segmentation
[3] Wireless sensor network
[4] Life time
[5] Computer science
[6] Web mining
[7] k-means

[8] Density-based spatial clustering of applications with noise
[9] DBSCAN based on Influence Space and Detection of border points

است، تعیین این پارامتر برای مجموعه داده‌های مختلفی که کاربر هیچ شناخت قبلی نسبت به آنها ندارد، دشوار باشد. برای حل این مشکل یک روش جدید برای تخمین پارامتر k براساس الگوریتم ژنتیک ارائه شده است. نتایج به‌دست‌آمده نشان‌دهندهٔ این است که روش پیشنهادی نسبت به روش‌های قبلی بهبود یافته است. همچنین در بخش تعیین خودکار پارامتر k نیز نتایج قابل قبولی به‌دست آمد.

بخش‌بندی مقاله در ادامه به این صورت است که در بخش دوم مروری برکارهای مرتبط، در بخش سوم مروری بر الگوریتم‌های خوشه‌بندی مبتنی بر چگالی، الگوریتم ژنتیک و درخت پوشای کمینه و در بخش چهارم، الگوریتم‌های پیشنهادی ارائه و در بخش آخر نتایج ارزیابی روش‌های پیشنهادی در مقایسه با دیگر روش‌ها ارائه شده است.

۲-کارهای مرتبط

در زمینهٔ پارامترهای ورودی الگوریتم DBSCANو الگوریتم‌های مبتنی بر نزدیک‌ترین همسایه کارهای متفاوتی انجام شده است که در زیر خلاصه‌ای از این روش‌ها ارائه می‌شود.

در الگوریتم DBSCANهر نمونه دادهٔ تصادفی که مشاهده نشده است، انتخاب و همسایگان آن در شعاع Eps استخراج می‌شوند.

اگر تعداد داده‌ها در شعاع Eps مساوی و یا بیشتر از MinPts باشد، یک خوشهٔ جدید ایجاد می‌شود و در غیر اینصورت دادهٔ بعدی که هنوز مشاهده نشده است، انتخاب می‌شود. این کار تا زمانی که همهٔ داده‌ها مشاهده شوند، ادامه پیدا می‌کند. داده‌ها در DBSCAN به دادهٔ هسته[1]، دادهٔ لبه[2] و دادهٔ نوفه[3] دسته‌بندی می‌شوند. اگر کمینه تعداد داده‌های MinPts در شعاع همسایگی Eps وجود داشته باشد، دادهٔ هسته ایجاد می‌شود[6].

در [12] یک روش برای سرعت بخشیدن به شناسایی داده‌ها در همسایگی هر داده ارائه شد. همچنین این روش در مقایسه با DBSCAN در مورد پارامترهای ورودی، زمان اجرایی الگوریتم و تشخیص خوشه‌ها در چگالی‌های متفاوت بهبود یافت.

در [5] پیشنهاد حذف شعاع همسایگی Eps و جایگزینی آن با پارامتر دیگر به نام p(نوفه در مجموعهٔ داده) پیشنهاد شد. در این روش تعداد پارامترهای ورودی الگوریتم DBSCAN کاهش پیدا نمی‌کند؛ بلکه انتخاب پارامتر p به‌طورمعمول خیلی ساده‌تر از شعاع همسایگی است؛ چون در برخی برنامه‌های کاربردی کاربر می‌داند که نوفه نسبت به داده‌ها بیشتر است.

در [2] یک الگوریتم خوشه‌بندی بدون پارامتر پیشنهاد شد که APSCAN[4] نام دارد و از انتشار همبستگی[5] برای تشخیص چگالی‌های محلی مجموعهٔ داده با ایجاد یک فهرست چگالی نرمال‌شده استفاده می‌کند. بعد از آن نخستین جفت از پارامترهای تراکم با هر جفت دیگر از پارامترهای تراکم در فهرست چگالی نرمال به‌عنوان پارامترهای ورودی DDBSCAN[6] استفاده می‌شوند که برای تولید مجموعه جواب‌های خوشه‌بندی ایجاد شده‌اند. به این ترتیب نتایج خوشه‌بندی‌های مختلف با پارامترهای مختلف تراکم از فهرست نرمال چگالی به‌دست می‌آید؛ سپس یک قانون به‌روزشده با پارامترهای مختلف و نتایج به‌دست‌آمده از اجرای DDBSCAN توسعه می‌یابد و بعد از آن ترکیب نتایج خوشه‌بندی به یک نتیجه خوشه‌بندی جامع ایجاد می‌شود.

در [15] یک الگوریتم جدید خوشه‌بندی معرفی شده است که داده‌ها را با توزیع‌ها و چگالی‌های متفاوت در مجاورت نوفه خوشه‌بندی می‌کند. این الگوریتم به‌طور خودکار تعداد حدآستانه برای مناطق متراکم مختلف را بدون هیچ دانش قبلی تشخیص می‌دهد؛ درحالی‌که الگوریتم‌های دیگر خوشه‌بندی همانند الگوریتم DBSCAN به تعیین بصری از یک حد آستانه برای تمایز دو منطقه متراکم نیاز دارند.

در [4]، DMDBSCAN[7] پیشنهاد شد. در این الگوریتم خوشه‌ها با شکل‌ها و اندازه‌های مختلف قابل تشخیص بودند. این الگوریتم با استفاده از رسم k-distچندین مقدار Eps را برای نواحی با چگالی‌های مختلف ایجاد می‌کند؛ با این تفاوت که DBSCAN یک مقدار Eps برای همهٔ داده‌ها در نظر گرفته تا تمام داده‌ها خوشه‌بندی شوند.

در [16]توسعهٔ الگوریتم DBSCAN با تمرکز بر روی یکی از مشکلات عمدهٔ این الگوریتم انجام شد که قادر به تشخیص خوشه‌هایی است که بدون خوشه‌های دیگر قابل شناسایی نیستند. در ابتدا Eps و MinPts توسط مقادیر کوچک مقداردهی شده و سپس بیت به بیت افزایش داده می‌شوند.

در [8] یک روش برای بهبود الگوریتم DBSCANارائه شده است. برای تعیین مقادیر خودکار مختلف Eps برای هر

[1] Core point
[2] Border point
[3] Noise point

[4] A parameter free algorithm for clustering
[5] Affinity Propagation
[6] Double-Density-Based SCAN
[7] Dynamic Method DBSCAN

داده، گراف k-dist ایجاد می‌شود. میانگین فاصلهٔ هر داده به k نزدیکترین همسایهٔ هر داده محاسبه می‌شود. برای تعیین MinPts تعداد داده‌ها در شعاع همسایگی برای هر داده در مجموعهٔ داده و به‌صورت یک‌به‌یک محاسبه و سپس امید ریاضی برای همین داده‌ها محاسبه و همین مقدار به‌عنوان MinPts انتخاب می‌شود.

در[1] یک روش جدید برای خوشه‌بندی ارائه شده است. این روش از k نزدیک‌ترین همسایه برای عمل خوشه‌بندی استفاده می‌کند. این روش قادر به شناسایی خوشه‌هایی با چگالی مختلف در مجموعهٔ داده‌ها است.

در [17] یک روش جدید برای تعیین Eps ارائه شده است که از مقدار K در چگالی‌های متفاوت در خوشه‌بندی استفاده می‌کند.

گراف k-dist برای تمام داده‌های مجموعهٔ داده رسم و میانگین فاصلهٔ هر داده به k نزدیک‌ترین همسایه محاسبه می‌شود. میانگین k فاصله به‌صورت چیدمان نزولی رسم می‌شود. حد آستانه هنگامی که تغییرات زیادی در رسم باشد، تعیین می‌شود.

در [9]روشی برای خودکارسازی پارامترهای ورودی DBSCAN ارائه شده است. در ابتدا هیستوگرام[1] بر روی دوبه‌دوی ماتریس شباهت داده‌های ورودی اعمال و بدون دخالت کاربر پارامتر های ورودی الگوریتم DBSCAN توسط خوشه‌بندی مجموعه‌های غالب[2] ایجاد می‌شود؛ بعد از آن خوشه‌بندی از مجموعه‌های غالب به DBSCAN گسترش یافته و پارامترهای ورودی توسط خوشه‌ها در مجموعه‌های غالب مشخص می‌شوند. همچنین این روش می‌تواند بر روی انواع مجموعهٔ داده تصاویر و همچنین خوشه‌هایی با انواع شکل اجرا شود.

در [11] یک روش خوشه‌بندی ترکیبی مؤثر و کارآمد به نام BDE-DBSCAN[3] ارائه شد که ترکیبی از الگوریتم تکاملی در فضای پیوسته و الگوریتم DBSCAN به‌صورت هم‌زمان است. این روش می‌تواند به‌سرعت و به‌صورت خودکار مقدار مناسب پارامترهای MinPts و Eps را مشخص کند.

در [18] یک الگوریم خوشه‌بندی مبتنی بر چگالی جدید به نام E-DBSCAN[4] معرفی شده است. در این روش مقادیر پارامترهای ورودی Eps و MinPts برای الگوریتم DBSCAN به‌طور خودکار تعیین می‌شود. در این روش ابتدا تعداد داده‌هایی که باید خوشه‌بندی شوند، به چندین ناحیه تقسیم می‌شوند و Eps و MinPts برای هر ناحیه محاسبه می‌شود. بعد از آن Eps و MinPts های جفت با هم ترکیب شده و بنابراین داده‌ها خوشه‌بندی می‌شوند.

۳-مروری بر DBSCAN، ISB-DBSCAN، ISDBSCAN الگوریتم ژنتیک و درخت پوشای کمینه

در این بخش از مقاله، مفاهیم اولیه در زمینهٔ الگوریتم‌های خوشه‌بندی مبتنی بر چگالی، تعاریف و توضیحاتی درخصوص الگوریتم ژنتیک و همچنین درخت پوشای کمینه ارائه شده است.

۳-۱- الگوریتم DBSCAN

در الگوریتم‌های خوشه‌بندی مبتنی بر چگالی، خوشه‌بندی با استفاده از تراکم داده‌ها در مجموعهٔ داده انجام می‌شود. در این روش‌ها هر داده با توجه به تعداد داده‌های موجود در همسایگی آن دسته‌بندی می‌شود. دو پارامتر زیر نقش اساسی در الگوریتم DBSCAN دارد و بایستی متناسب با مسأله انتخاب شوند. این دو پارامتر به‌صورت زیر تعریف می‌شوند:

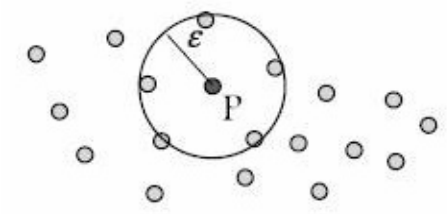
۱. پارامتر MinPts: کمینه تعداد داده‌های موجود در اطراف یک داده را مشخص می‌کند.

۲. پارامتر ε : شعاع همسایگی را مشخص می‌کند.

برای پیاده‌سازی روش‌های خوشه‌بندی مبتنی بر چگالی لازم است تا ابتدا اصطلاحاتی تعریف شوند:

تعریف ۱: اگر p را نقطهٔ هستهٔ یک همسایگی و ε شعاع همسایگی برای نقطهٔ p در نظر گرفته شود آن‌گاه همسایگی به شعاع ε برای نقطهٔ p همانند شکل (۱) به‌صورت زیر تعریف می‌شود:

$$N=(q \text{ in data set } D \mid dist(p,q) \leq \varepsilon)$$

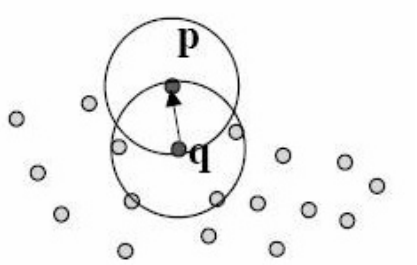


(شکل-۱): چگالی نقاط محلی در نقطهٔp به شعاعε [6]

(Figue-1): The density of local points at a point p to radius ε. [6]

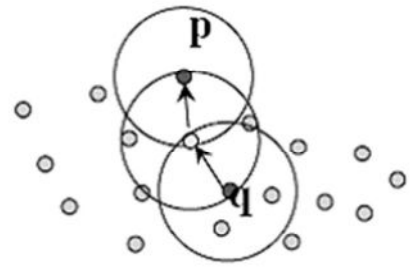
تعریف ۲: دادهٔ p را در دسترسِ مستقیم چگالی q گویند اگر p درون یک همسایگی به شعاع ε با هستهٔ q باشد(شکل(۲)).

[1] Histogram equalization
[2] Dominant sets
[3] Binary Differential Evolution DBSCAN
[4] Efficient density-based clustering algorithm



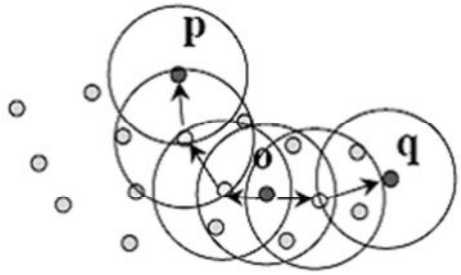
(شکل-۲): p در دسترس مستقیم چگالی q قرار دارد.[6]
(Figue-2): A point p is directly density reachable from a point q. [6]

تعریف ۳: دادهٔ p را در دسترسِ چگالی q گویند اگر داده‌ای وجود داشته باشد که هم درون یک همسایگی به شعاع ε با هستهٔ p و هم درون یک همسایگی به شعاع ε با هستهٔ q باشد(شکل(۳)).



(شکل-۳): دادهٔ p در دسترسِ چگالی دادهٔ q قرار دارد.[6]
(Figure-3): A point p is density reachable from a point q. [6]

تعریف ۴: دادهٔ p را متصلِ چگالی q گویند اگر داده‌ای مانند o وجود داشته باشد که هم در دسترسِ چگالی p و هم در دسترسِ چگالی q باشد(شکل(۴)).



(شکل-۴): دادهٔ pمتصلِ چگالی دادهٔ q است.[6]
(Figure-4): A point p is density connected from a point q. [6]

کاربرد الگوریتم DBSCAN در کشف خوشه‌هایی به‌شکل دلخواه در مجموعه داده‌هایی با داده‌های نوفه‌دار است. پس از برچسب‌دهی به داده‌ها، داده‌های هسته به‌عنوان نمایندهٔ خوشه انتخاب شده و یک خوشهٔ جدید تشکیل می‌دهند. داده‌های مرزی به خوشه‌ای تعلق می‌گیرند که به نمایندهٔ آن نزدیک‌تر باشند و داده‌های پرت نیز حذف خواهند شد.

۲-۳- الگوریتم ISB-DBSCAN

همان‌طور که درقبل نیز بیان شد، الگوریتم DBSCAN نیاز به دو پارامتر ورودی برای خوشه‌بندی دارد که تعیین آنها توسط کاربر مشکل است. الگوریتم ISB-DBSCAN تنها نیاز به یک پارامتر ورودی به‌عنوان k نزدیک‌ترین همسایه دارد. این روش نیز همانند DBSCAN مبتنی بر چگالی است؛ اما در مقایسه با DBSCAN از مزایایی برخوردار است. تعیین پارامتر k به‌طورمعمول ساده‌تر از پارامترهای DBSCAN است؛ همچنین این روش در تعیین خوشه‌هایی با چگالی‌های مختلف در انواع مجموعهٔ داده‌ها بهتر از DBSCAN عمل می‌کند. یکی دیگر از برتری این روش در مقایسه با DBSCAN تشخیص دادهٔ لبه از دادهٔ نوفه است. برای پیاده‌سازی این روش نیاز است تا ابتدا اصطلاحاتی تعریف شوند:

تعریف ۵: به ازای هر $x \in D$ ، فاصلهٔ k داده تا p به‌صورت $k_{dist}(p)$ نشان داده می‌شود.

تعریف ۶: k نزدیک‌ترین همسایهٔ دادهٔ p در مجموعه دادهٔ D به‌صورت $NN_k(x)$ تعریف می‌شود؛ به‌طوری‌که x داده در D وجود داشته باشد که $d(p,x) \le k_{dist}(p)$ باشد یا به‌عبارتی دیگر:
$NN_k(x)=\{X \in D \setminus \{p\} \mid d(p,x) \le k_{dist}(p)\}$

تعریف ۷: $RNN_k(p)$، معکوس k نزدیک‌ترین همسایه است که به‌صورت زیر تعریف می‌شود:
$RNN_k(p)=\{q \in D \mid q \in NN_k(p)\}$

تعریف ۸: $IS_k(p)$ یا فضای نفوذ k[1] به‌صورت زیر تعریف می‌شود:
$IS_k(p)= NN_k(x) \cap RNN_k(p)$

تعریف ۹: همسایگی نقطهٔ p به‌صورت زیر تعریف می‌شود:
$\{q \mid q \in D \cap q \in IS_k(p)\}$

تعریف ۱۰: اگر تعداد داده در $IS_k(p)$ بزرگ‌تر از $2k/3$ باشد آن‌گاه آن داده به‌عنوان دادهٔ هسته شناخته می‌شود.

تعریف ۱۱: دادهٔ p در دسترسِ مستقیمِ چگالی دادهٔ q قرار دارد اگر $p \in IS_k(q)$ باشد و $IS_k(q)$ بزرگ‌تر از $2k/3$ باشد.

تعریف ۱۲: دادهٔ p با توجه به k در دسترسِ چگالی هستهٔ q قرار دارد، اگر مجموعه‌ای از زنجیرهٔ داده‌ها مانند $p_1,\dots p_n$ وجود داشته باشد که $p_1=q$ و $p_n=p$ باشد و به‌طوری‌که p_{i-1} در دسترسِ مستقیمِ چگالی از p_i با توجه به k بوده و همچنین p_i دادهٔ هسته باشد.

تعریف ۱۳: دادهٔ p متصلِ چگالی به دادهٔ q با توجه به k است، اگر داده‌ای مانند o وجود داشته باشد که $o \in D$ باشد به‌طوری‌که هر دوی داده‌های pوq در دسترسِ چگالی هسته از o با توجه به k باشند.

[1] Influence space

الگوریتم ISB-DBSCAN همانند شکل (۵) عدد k را از ورودی دریافت کرده و طبق سه مرحله داده‌ها را خوشه‌بندی می‌کند. در ابتدا برای تمام داده‌ها IS_k محاسبه و در مرحلهٔ بعد داده‌های هسته مشخص می‌شوند. در مرحلهٔ آخر نیز داده‌های لبه و نوفه مشخص می‌شوند.

```
Require: D={x1,x2,…,xn}: the dataset.
K: the number of neighbors.
Ensure: C={ C1,C2,…,Ck }: set of clusters.
1: function ISB-DBSCAN D,k
2:   ClusterID=1
3:   mark all points xi∈D as "UNCLASSIFIED"
4:   calculate the influence space ISk(xi)  of each point xi∈D
5:   for all xi∈D do
6:     if xi is marked as "UNCLASSIFIED " then
7:       if ExpandCoreCluster xi, ClusterID then
8:         ClusterID++
9:       end if
10:    end if
11:   end for
12:  for all xi∈D do
13:    if xi is marked as "UNCLASSIFIED " then
14:      search each point yj∈ISk(xi) in the influence space
     of xi
15:        if points in ISk(xi) are all marked as "NOISE" or
     there is no point in ISk(xi)  then
16:          xi is labeled as "NOISE"
17:        else
18:          mark xi as ClusterID of the closest core point
     in ISk(xi)
19:        end if
20:    end if
21:  end for
22:end function
23:
24: function ExpandCoreCluster (xi,CLusterID)
25:  SeedList= ISk(xi)
26:  if |SeedList| >2k/3 then
27:    xi is labeled as ClusterID
28:  else
29:    false
30:  end if
31:  for all yj∈SeedList do
32:    if | ISk(yj) | > 2k/3 then
33:      yj is labeled as ClusterID
34:      for all zm∈| ISk(yj) | do
35:        if zm is labeled as "UNCLASSIFIED " or zm is
     labeled as "NOISE" then
36:          if zm is not in SeedList then
37:            add zm into SeedList
38:          end if
39:        end if
40:      end for
41:    end if
42:  end for
43:  true
44:end function
```

(شکل-۵): الگوریتم خوشه‌بندی ISB-DBSCAN[12]

(Figure-5): ISB-DBSCAN Clustering Algorithm [12]

۳-۳-الگوریتم ISDBSCAN

الگوریتم ISDBSCAN[1] همانند الگوریتم ISB-DBSCAN برای همهٔ داده‌ها IS_K را محاسبه می‌کند. در مرحلهٔ بعد یک داده به‌صورت تصادفی انتخاب شده و پیرامون آن خوشه ایجاد می‌شود. خوشه تا جایی گسترش پیدا می‌کند که به دادهٔ لبه و یا پرت برخورد کند. دادهٔ نوفه زمانی تشخیص داده می‌شود که تعداد داده‌ها در IS_K کمتر از حد تعریف شده باشد. این حد آستانه تعداد داده‌هایی با اندازهٔ IS_K کمتر از 2k/3 است. در شکل (۶) شبه‌کد الگوریتم ISDBSCAN نشان داده شده است.

```
ISDBSCAN(s,k)
Input
S:dataset of objects to be cluster;
K: number of neighbors;
Output
A set of clusters;
1: i=1;
2: while S≠∅
3:   p=Randomly select a point in S;
4:   Ci=MAKECLUSTER(S,p,k,i);
5:   S=S\Ci;
6:   if |Ci|>k then
7:     membership(p)=i;
8:     i=i+1;
9:   else
10:    membership(q)=noise;
11:  end if
12:  end while
13:  return{C1,…,Ci-1};

MAKECLUSTER(S,p,k.i)
Input
S: dataset of objects to be cluster;
p: starting point for cluster construction;
k: number of neighbor;
i: cluster index;
Output
A cluster or a set of noise
1:   if |ISK(p)|> 2k/3 then
2:     for each q∈ ISK(p) do
3:       if membership(q)=-1 then
4:         membership(q)=i;
5:         C=C∪{q};
6:         C=C∪ MAKECLUSTER(S,q,k,i);
7:       end if
8:     end for
9:   end if
10:  return C;
```

(شکل-۶): الگوریتم خوشه‌بندی ISDBSCAN[1]

(Figure-6): ISDBSCAN Clustering Algorithm [1]

۴-۳- الگوریتم ژنتیک

الگوریتم ژنتیک را می‌توان به‌طور ساده، یک روش جستجوگر نامید که بر پایهٔ مشاهدهٔ خصوصیات فرزندان نسل‌های متوالی، و انتخاب فرزندان بر اساس اصل (اثبات‌نشدهٔ) بقای بهترین، پایه‌ریزی شده است[13].

[1] Influence space DBSCAN

الگوریتم ژنتیک بر روی فرزندان یک نسل، (مجموعه جواب‌های مسأله در یک مرحله)، از قوانین موجود در علم ژنتیک تقلید کرده و با به‌کاربردن آنها، به تولید فرزندان با خصوصیات بهتر، (جواب‌های نزدیک‌تر به هدف مسأله) می‌پردازد و در هر نسل به‌کمک فرآیند انتخابی که براساس ارزش جواب‌ها است و تولید مثل فرزندان (جواب‌های) جدید، تقریب بهتری از جواب نهایی را به‌دست می‌آورد.

این فرآیند باعث می‌شود که نسل‌های جدید با شرایط مسئله سازگارتر باشند، این رقابت میان فرزندان (جواب‌ها) و پیروزشدن فرزند شایسته‌تر (جواب بهتر) و انتخاب شدنشان توسط الگوریتم ژنتیک برای تولید مثل بعدی و کناررفتن فرزندان یا همان ژن‌های مغلوب که درحقیقت جواب‌های دور از هدف مسأله هستند در نظر گرفته می‌شود.

۵-۳- درخت پوشای کمینه

درخت پوشای کمینه[1] یا درخت فراگیر کمینه در گراف‌های ارزش‌دار (وزن‌دار) ساخته می‌شود[7]. منظور از یک درخت پوشا از گراف، درختی است که شامل همه رئوس این گراف باشد؛ ولی فقط بعضی از یال‌های آن را در برگیرد. منظور از درخت پوشای کمینه (برای گراف همبند وزن‌دار) درختی است که بین درخت‌های پوشای آن گراف، مجموع وزن یال‌های آن، کمترین مقدار ممکن باشد. برای به‌دست‌آوردن درخت پوشای کمینه یک گراف جهت‌دار متصل می‌توان از الگوریتم‌های متفاوتی استفاده کرد. از معروف‌ترین الگوریتم‌های یافتن درخت پوشای کمینه، الگوریتم پریم است که به‌صورت خلاصه بیان می‌شود[3].

در این روش یک رأس انتخاب شده و کمترین یال (یال با کمترین وزن) که از آن می‌گذرد، انتخاب می‌شود. در مرحلهٔ بعد یالی انتخاب می‌شود که کمترین وزن را در بین یال‌هایی که از دو گره موجود می‌گذرد، داشته باشد. به همین ترتیب در مرحلهٔ بعد یالی انتخاب می‌شود که کمترین وزن را در بین یال‌هایی که از سه گره موجود می‌گذرد، داشته باشد.

این روال آن‌قدر تکرار شده تا درخت پوشای کمینه حاصل شود. باید توجه کرد که یال انتخابی در هر مرحله در صورتی انتخاب می‌شود که در گراف، دور ایجاد نکند.

۴- الگوریتم‌های پیشنهادی

همان‌طور که در قسمت‌های قبلی نیز بیان شد، الگوریتم DBSCAN دارای معایبی همچون انتخاب دو پارامتر ورودی توسط کاربر است که می‌تواند در انواع مختلف مجموعهٔ داده مشکل ایجاد کند. از دیگر مشکلات الگوریتم DBSCAN می‌توان به عدم پشتیبانی از خوشه‌هایی با چگالی‌های مختلف در مجموعهٔ داده اشاره کرد که با انتخاب یک Eps قادر به خوشه‌بندی تمام دادها نیست.

الگوریتم ISB-DBSCAN تمامی مشکلات بالا را به‌خوبی حل کرد اما با توجه به تعریف جدید دادهٔ هسته نمی‌تواند در تمامی مجموعهٔ داده‌ها به‌خوبی عمل کند.

در این مقاله تعریف جدیدی برای دادهٔ هسته و همچنین تعریف‌هایی برای در دسترسِ چگالی هسته، چگالی متصل و در دسترسِ مستقیمِ چگالی براساس دادهٔ هسته ارائه می‌شود؛ ضمن این‌که تعریف‌های ۵ تا ۹ طبق الگوریتم ISB-DBSCAN برای این روش همچون قبل وجود دارد. در روش پیشنهادی ساختار الگوریتم ISB-DBSCAN مورد تغییراتی قرار گرفته است تا بهترین جواب را در هر مجموعهٔ داده به‌دست آورد. ابتدا اصطلاحات زیر تعریف می‌شوند:

تعریف ۱۴: دادهٔ p دادهٔ هسته است، اگر تعداد داده در $IS_k(p)$ آن بزرگتر یا مساوی با k-1 باشد.

تعریف ۱۵: دادهٔ p در دسترسِ مستقیمِ چگالی q است، اگر دادهٔ $p \in ISk(q)$ و q یک دادهٔ هسته باشد.

تعریف ۱۶: دادهٔ p را در دسترسِ چگالی هستهٔ q گویند، اگر q یک دادهٔ هسته باشد و زنجیره‌ای از داده‌های هسته وجود داشته باشند که دادهٔ p دست‌کم در دسترسِ مستقیمِ چگالی یکی از این داده‌ها باشد.

تعریف ۱۷: دادهٔ p را متصلِ چگالی q گویند، اگر داده‌ای مانند o وجود داشته باشد که هم در دسترسِ چگالی هستهٔ p و هم در دسترسِ چگالی هستهٔ q باشد.

الگوریتم پیشنهادی یا MDD-ISB-DBSCAN[2] همچون الگوریتم ISB-DBSCAN یک پارامتر ورودی k را از ورودی دریافت کرده و عمل خوشه‌بندی را انجام می‌دهد. این الگوریتم همانند الگوریتم ISB-DBSCAN شامل سه بخش است. در بخش نخست تمام مجموعهٔ داده بررسی شده و IS_k برای تمام داده‌ها محاسبه می‌شود. در مرحلهٔ دوم تمام داده‌ها از نظر اینکه آیا دادهٔ هسته هستند یا خیر بررسی می‌شوند. برخلاف الگوریتم ISB-DBSCAN، در الگوریتم MDD-ISB-DBSCAN اگر دادهٔ بررسی‌شده دادهٔ هسته بود، تمامی نقاط همسایگی آن داده در IS_k با همان برچسب دادهٔ هسته برچسب زده می‌شوند. در این مرحله از الگوریتم اطمینان حاصل می‌شود که تمام نقاطی که دارای بیشترین چگالی هستند، برچسب زده و به‌عنوان یک خوشه مجزا در نظر گرفته شده‌اند.

[1] Minimal spinning tree

[2] Modify Density Definition of ISB-DBSCAN

همچنین این کار باعث می‌شود تا خوشه‌هایی که در مجاورت یکدیگر قرار دارند، به‌خوبی از یکدیگر جدا شوند.

بعد از پیداکردن تمام داده‌های هسته در بعضی مواقع ممکن است، داده‌هایی وجود داشته باشند که هم‌چنان عضو هیچ خوشه‌ای قرار نگرفته باشند. برای جلوگیری از این مشکل دو راه حل وجود دارد. در راه حل نخست می‌توان مقدار k را بزرگ در نظر گرفت تا تمام داده‌ها را دربرگیرد و تمام داده‌ها عضو خوشه‌ها باشند که در این‌صورت وجود نوفه در داده‌ها بی‌معنی است. در راه حل دوم همچون الگوریتم ISB-DBSCAN داده‌هایی را که برچسب ندارند انتخاب کرده و آنها را با نزدیک‌ترین دادهٔ هسته برچسب‌دهی و یا اینکه به‌عنوان نوفه تلقی می‌کنیم.

در الگوریتم MDD-ISB-DBSCAN برخلاف الگوریتم ISB-DBSCAN که در مرحلهٔ آخر فقط دادهٔ لبه و یا نوفه را برچسب‌دهی می‌کند، می‌تواند خوشهٔ جدید نیز تولید کند. این قابلیت به‌خاطر وجود خوشه‌هایی با چگالی پایین است که در مرحله دوم الگوریتم قادر به شناسایی نبوده‌اند. همچنین در الگوریتم ISB-DBSCAN برای تعیین نوفه در مجموعهٔ داده معیارهایی همچون خالی بودن RRN آن داده و یا خالی‌بودن IS_K آن داده درنظر گرفته می‌شود و یا درصورتی‌که تمام داده‌ها در همسایگی داده‌ای که برچسب ندارد نوفه باشند، آن داده به‌عنوان نوفه در نظر گرفته می‌شود. در روش MDD-ISB-DBSCAN علاوه‌بر خالی‌بودن IS_K، از بُعد مجموعهٔ داده برای تعیین نوفه استفاده شده است. اگر تعداد داده‌های IS_K در قسمت آخر الگوریتم MDD-ISB-DBSCAN کمتر از تعداد ابعاد مجموعهٔ داده باشد آن داده به‌عنوان نوفه در نظر گرفته می‌شود. مراحل الگوریتم MDD-ISB-DBSCAN به‌اختصار به شرح زیر است:

۱-تمامی داده‌ها را با ۱- برچسب‌گذاری کن.
۲-برای هر داده در مجموعهٔ داده IS_K را محاسبه کن.
۳-به‌ازای تمام داده‌ها در مجموعه داده کارهای زیر را انجام بده:
۳-۱-یک داده با برچسب ۱- از ابتدای مجموعهٔ داده انتخاب کن.
۳-۲-اگر تعداد IS_K دادهٔ انتخابی بزرگ‌تر یا مساوی k-1 بود به مرحلهٔ ۳-۳ برو در غیر این‌صورت مقدار false برگشت داده شده و دادهٔ بعدی انتخاب می‌شود.
۳-۳-دادهٔ انتخابی را با برچسب جدید برچسب‌گذاری کن. اگر تعداد IS_K دادهٔ انتخابی بزرگ‌تر یا مساوی k-1 بود داده‌های موجود در IS_K را به انتهای فهرست همسایگی اضافه کن.
۳-۴-از فهرست همسایگی دادهٔ بعدی را انتخاب کن و به مرحله ۳-۳ برو.
۳-۵-اگر تمام داده‌ها در فهرست همسایگی با برچسب جدید برچسب‌گذاری شده باشند به مرحلهٔ ۳-۶ برو.
۳-۶-اگر در مرحله ۳-۳ یک داده با برچسب جدید برچسب‌گذاری‌شده بود مقدار true برگشت داده می‌شود در غیر این‌صورت false برگشت داده می‌شود، اگر مقدار برگشتی true بود عدد برچسب خوشه‌بندی را یک واحد اضافه کن.
۴-تا زمانی که به انتهای مجموعهٔ داده نرسیدی به مرحلهٔ ۳-۱ برو.
۵-داده‌های که برچسب ۱- دارند را براساس بیشترین اندازهٔ IS_K مرتب کن.
۵-۱-تا زمانی که داده با برچسب ۱- در مجموعهٔ داده وجود داشته باشد، کارهای زیر را انجام بده:
۵-۲-داده‌ای که بزرگ‌ترین اندازه IS_K را دارد انتخاب کن.
۵-۳-اگر تمام داده‌ها در همسایگی دادهٔ انتخاب‌شده یک برچسب (غیر از نوفه) داشتند:
۵-۳-۱-اگر اندازه IS_K بزرگ‌تر از ابعاد مجموعهٔ داده بود آن‌گاه دادهٔ انتخاب‌شده نیز با برچسب داده‌های همسایه برچسب‌گذاری و در غیر این‌صورت به‌عنوان نوفه شناخته می‌شود.
۵-۴-اگر تعداد داده‌ها در همسایگی دادهٔ انتخاب‌شده برچسب داشته باشند (غیر از نوفه) اما بیش از یک برچسب برای آن داده‌ها وجود داشته باشد آن‌گاه:
۵-۴-۱-دادهٔ انتخاب شده را به نزدیک‌ترین دادهٔ هسته برچسب‌گذاری کن.
۵-۵-اگر در همسایگی دادهٔ انتخاب‌شده هیچ برچسبی وجود نداشت(همهٔ داده‌ها برچسب ۱- داشتند) آن‌گاه:
۵-۵-۱-اگر اندازه IS_K دادهٔ انتخاب‌شده بزرگ‌تر از ابعاد مجموعه‌داده بود آن‌گاه دادهٔ انتخاب‌شده را برچسب‌گذاری کن و IS_K را به فهرست همسایگی اضافه کن. در غیر این‌صورت آن داده را به‌عنوان نوفه در نظر بگیر.
۵-۵-۲-به‌ازای تمام فهرست همسایگی مراحل زیر را انجام بده:
۵-۵-۲-۱-اگر دادهٔ انتخاب‌شده در فهرست همسایگی دارای برچسب ۱- بود پس آن را برچسب‌گذاری کن.
۵-۵-۲-۲-اگر فهرست همسایگی دادهٔ انتخاب‌شده در مرحلهٔ ۵-۵-۱-۲ بزرگ‌تر از ابعاد مجموعهٔ داده بود، پس IS_K دادهٔ انتخاب‌شده را نیز به انتهای فهرست همسایگی اضافه کن.
۵-۵-۳-اگر به انتهای فهرست همسایگی نرسیدی به مرحلهٔ ۵-۵-۱-۲ برو.
۵-۵-۴-اگر دست‌کم یک داده در این مرحله با برچسب جدید برچسب‌گذاری شده باشد، مقدار true برگشت داده می‌شود درغیر این‌صورت false برگشت داده می‌شود.
۵-۶- اگر مقدار برگشتی true بود عدد برچسب خوشه‌بندی را یک واحد اضافه کن.
۵-۷-برو به مرحله ۵-۱.

خروجی الگوریتم تعداد خوشهٔ به‌دست‌آمده با توجه به عدد k ورودی است. اگر مقدار k به‌درستی توسط کاربر انتخاب شود، این روش خوشه‌بندی می‌تواند در انواع مجموعهٔ داده مفید واقع شود و خوشه‌بندی را به‌طور صحیح انجام دهد. در شکل(۷) شبه‌کد الگوریتم پیشنهادی ارائه شده است.

```
Require: d={x1,x2,…,xn}: the dataset.
K: the number of neighbors.
Ensure: C={ C1,C2,…,Ck }: set of clusters.

1: function IISB-DBSCAN D,k
2:   ClusterID=1
3:   mark all points xi∈D as -1
4:  calculate the influence space ISk(xi) of each point xi∈D
5:   for all xi∈D do
6:     if xi is marked as -1 then
7:       if expandCoreCluster(d,ClusterID,i,ISK) then
8:          ClusterID++
9:       end if
10:    end if
11:  end for
12: If threre was a point with -1 label then
13:    p=Sort points with ISK length in dataset
    14:    for all p in dataset
15:      i=select the largest length of ISK(i);
16:     BorderNoiseOrnewCluster(d,ClusterID,ISK,i);
17:    end
18: end
19:                                                    function
BorderNoiseOrnewCluster(d,ClusterID,ISK,i);
20:          seedList= ISK(i);
21:          If there was a label in the neighborhood of the
seedList for the selected point (other than noise and -1)
then
22:            If length(seedList)> Dataset Dimensions
then
23:              d(i)= neighborhood label;
24:            else
25:              d(i)=noise;
26:            end
27:      return d,false;
28:          end
29:          If there was more than one labels in the
neighborhood of the seedList for the selected point (other
than noise and -1) then
30:        If length(seedList)> Dataset Dimensions then
31:              d(i)= label of nearest core point;
32:            else
33:              d(i)=noise;
34:            end
35:        return d,false;
36:        end
37:      If there is no label (label -1) then
38:        If expandNewCluster(d,ClusterID,i) then
39:            ClusterID=ClusterID+1;
40:        end
41:     end
42:    return d,ClusterID;
43:    end

44: function expandNewCluster(d,ClusterID,i)
45:  seedList=ISK(i);
46:   If length(seedList)> Dataset Dimensions then
47:    result=true;
48:    d(i)=ClusterID;
```

```
49:    for all point in seedList
50:      if d(seedList(i))==-1
51:        d(seedList(i))=ClusterID;
52:      if length(ISK(seedList(i)))> Dataset Dimensions
53:         if ISK(seedList(i)) not in seedList then
54:           add ISK(seedList(i) to seedList ;
55:         end
56:      end
57:    end
58:  else
59:  result=false;
60:  d(i)=noise;
61:  end
62:  return d,result;
63: end

64: function expandCoreCluster(d,ClusterID,i,ISK)
65:  if length(ISK(i))>=(k-1)
66:     seedList= ISK(i);
67:     result=true;
68:     d(i)=ClusterID;
69:     i=1;
70:     for all point in seedList
71:       d(seedList(i))=ClusterID;
72:       if length(ISK(seedList(i)))>=(k-1)
73:          if ISK(seedList(i) )not in seedList then
74:             add ISK(seedList(i) to seedList ;
75:          end
76:       end
77:       i=i+1;
78:      end
79:    else
80:      result=false;
81:    end
82:  return result,d;
83: end
```

(شکل-۷): الگوریتم خوشه‌بندیMDD-ISB-DBSCAN

(Figure-7): MDD-ISB-DBSCAN Clustering Algorithm

۱-۴- تحلیل حساسیت پارامتر در الگوریتم پیشنهادی

همان‌طورکه در مبحث قبلی نیز بیان شد، الگوریتم MDD-ISB-DBSCAN به یک پارامتر وردی k نیاز دارد. پارامتر k می‌تواند به جای هر دو پارامترهای ورودی DBSCAN استفاده شده و خوشه‌های دقیق‌تری را تولید کند. درواقع با استفاده از این پارامتر و محاسبه مقدار IS_k می‌توان تعیین کرد که کدام یک از داده‌ها، دادهٔ هسته هستند. انتخاب این پارامتر نسب به ورودی‌های الگوریتم DBSCAN ساده‌تر بوده و جایگزین مناسبی برای آن پارامترها است. مشکل اساسی زمانی است که کاربر شناختی از مجموعهٔ داده نداشته و نسبت به آن هیچ دانش قبلی ندارد؛ پس انتخاب این پارامتر می‌تواند کار مشکلی باشد.

برای حل مشکل بالا یک روش پیشنهادی به نام PMDD-ISB-DBSCAN[1] در این قسمت ارائه می‌شود که می‌تواند مقدار خودکار پارامتر k را به‌طور تقریبی تعیین کند. برای این کار از الگوریتم ژنتیک استفاده می‌شود تا با تابع

[1] Parameter Modify Density Definition of ISB-DBSCAN

برازش پیشنهادی بهترین مقدار k را تخمین بزند. پارامترهای ورودی الگوریتم ژنتیک طبق جدول(۱) تعیین می‌شوند.

(جدول-۱): پارامترهای ورودی الگوریتم ژنتیک
(Table-1): Input Parameters of the Genetic Algorithm

پارامتر ورودی	مقدار
تعداد جمعیت اولیه	100
نرخ ترکیب	0.8
نرخ جهش	0.2
تعداد اجرا	100
تعداد متغییرهای کروموزوم	10

نوع کروموزوم تعریف‌شده در اینجا به‌صورت دودویی است و نوع ترکیب و جهش استفاده‌شده در روش PMDD-ISB-DBSCAN به‌صورت ترکیب یک نقطه و در جهش به‌صورت جهش وارونه‌سازی بیت است.

البته با توجه به نوع مسأله می‌توان از انواع جهش و ترکیب دیگر نیز استفاده کرد. برای انتخاب فرزندان در ترکیب از روش چرخ رولت و در جهش از روش تصادفی استفاده شده است. برای انتخاب فرزندان می‌توان از انواع روش‌های دیگر نیز استفاده کرد. باید توجه داشت در موارد خاصی که تمام جمعیت اولیه‌ای که به‌صورت تصادفی تولید شده‌اند بد باشد، تابع برازش مقدار صفر را برمی‌گرداند. برای حل این مشکل نیز از روش انتخاب فرزندان به‌صورت تصادفی استفاده می‌شود. حد پایین و بالا برای تعیین پارامتر k در روش PMDD-ISB-DBSCAN محدود شده و حد پایین برابر با ۳ و حد بالا برابر با تعداد داده‌های مجموعهٔ داده تقسیم بر ۲ است.

یکی از مهم‌ترین قسمت‌ها در الگوریتم ژنتیک تعریف تابع برازش برحسب مسأله مورد نظر است. استفاده از معیارهای ارزیابی خوشه‌بندی به‌عنوان تابع برازش می‌تواند به‌خوبی در خوشه‌های کروی جواب مورد نظر را بیابد؛ اما در خوشه‌بندی مبتنی بر چگالی از آنجا که شکل خوشه‌ها به‌انحصار کروی شکل نیست؛ لذا در بیشتر موارد ممکن است جواب حاصل توسط این معیارها دقیق نباشد.

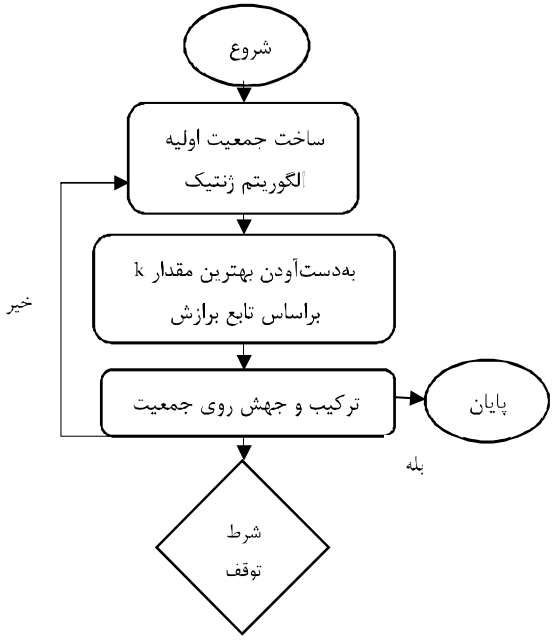
برای تشخیص اشکال گوناگون و دلخواه و همچنین فاصله بین داده‌های این خوشه‌ها از روش محاسبه فاصله درون خوشه‌ای با استفاده از درخت پوشای کمینه در فرمول (۱) استفاده شده است. درخت پوشای کمینه برای هر خوشه ممکن است، دارای یالی باشد که متعلق به آن خوشه نیست؛ درواقع داده‌ای که با یک یال به خوشهٔ مربوطه متصل شده است؛ اما فاصلهٔ آن داده تا بقیه داده‌ها زیاد است، پس باید از خوشه حذف شود. برای حذف این یال‌ها از فرمول (۲) استفاده می‌شود. برای این کار ابتدا خوشه‌بندی انجام شده و تعداد خوشه‌ها و داده‌های برچسب‌خورده به تابع برازش الگوریتم ژنتیک فرستاده می‌شوند؛ سپس برای هر خوشه یک درخت پوشای کمینه ایجاد شده و بزرگ‌ترین یال به‌دست می‌آید. حال اگر بزرگ‌ترین یال هر خوشه از دو برابر حاصل‌جمع میانگین یال‌های آن خوشه با انحراف معیار یال‌های آن خوشه کمتر باشد، پس مقدار یک و در غیر اینصورت مقدار صفر برگشت داده می‌شود. براساس عدد برگشت‌داده‌شده، تابع برازش مقدار برازندگی را محاسبه کرده و برای هر k برگشت داده می‌شود. شمای کلی روش PMDD-ISB-DBSCAN برای تعیین خودکار پارامتر ورودی k در شکل (۸) رسم شده است.

$$f(x)=\begin{cases}\sum_{i=1}^{C}(\text{mst}(C_i))-N & ,M=1\\ 0 & ,M=0\end{cases} \quad (۱)$$

-در فرمول (۱)، C تعداد خوشه، C_i خوشهٔ i ام، $\text{mst}(C_i)$ درخت پوشای کمینهٔ خوشهٔ C_i، N تعداد داده‌های نوفه در مجموعهٔ داده و M طبق فرمول (۲) تعریف می‌شود.

$$M=\begin{cases}0, & 2\times(\text{Avg}(C_i)+\text{SD}(C_i))\le B(C_i)\\ 1, & 2\times(\text{Avg}(C_i)+\text{SD}(C_i))>B(C_i)\end{cases} \quad (۲)$$

-در فرمول (۲)، $\text{Avg}(C_i)$ میانگین یال‌های خوشهٔ C_i، $\text{SD}(C_i)$ انحراف معیار یال‌های خوشهٔ C_i و $B(C_i)$ برابر با بزرگ‌ترین یال در خوشهٔ C_i است.



(شکل-۸): انتخاب خودکار پارامتر ورودی در الگوریتم PMDD-ISB-DBSCAN
(Figure-8): Automatic selection of the input parameter in the PMDD-ISB-DBSCAN algorithm

مقدار تابع برازش پیشنهادشده می‌تواند بین حد پایین صفر و حد بالای بهترین نتیجهٔ خوشه‌بندی براساس مجموعهٔ داده باشد. از آنجا که ممکن است، تابع برازش پیشنهادی تنها یک خوشه را تشخیص دهد(همهٔ مجموعهٔ داده را به‌عنوان یک خوشه در نظر بگیرد) لذا ممکن است با مقدار تابع برازش بالا جواب به‌طورکامل اشتباهی توسط الگوریتم برگشت داده شود. برای جلوگیری از این مشکل وجود یک شرط که دست‌کم تعداد خوشه را تعیین کند، ضرروی است؛ یعنی درالگوریتم PMDD-ISB-DBSCAN فرض بر این گرفته شده است که دست‌کم تعداد خوشه‌ها در مجموعهٔ داده ۲ باشد.

روش ارائه‌شده از نظر زمان اجرا مورد بررسی قرار نگرفته است؛ زیرا در بسیاری از موارد کاربردی داده‌کاوی عملیات خوشه‌بندی به‌صورت برون‌خط انجام شده و هدف مورد توجه تنها دقت و کیفیت خوشه‌بندی است. در چنین مواردی عامل زمان از اهمیت چندانی برخوردار نخواهد بود. به‌طور مثال در خوشه‌بندی جمعیت یک کشور برای مقاصد برنامه‌ریزی اقتصادی و اجتماعی سرعت محاسبات و مقایسهٔ زمان بر حسب ثانیه یا دقیقه و ... دارای اهمیت نیست؛ زیرا یک‌بار خوشه‌بندی انجام می‌شود و سال‌ها مورد استفاده قرار می‌گیرد.

۵-نتایج ارزیابی

در این بخش، الگوریتم MDD-ISB-DBSCAN به‌همراه روش تخمین پارامتر خودکار یا PMDD-ISB-DBSCAN پیاده‌سازی شده است. هر دو الگوریتم در نرم‌افزار متلب پیاده‌سازی شده‌اند. هر دو روش پیشنهادی ارائه‌شده روی دستگاه رایانه‌ای رومیزی و با مشخصات ۴ گیگابایت رم، سی پی یو core i5 2.8GHz اجرا شده‌اند. نتایج به‌دست‌آمده با دو الگوریتم ISB-DBSCAN [12] و ISDBSCAN [1] مقایسه شده‌اند. ارزیابی بر روی یازده مجموعهٔ داده استاندارد و در ابعاد مختلف بررسی شده است. مجموعه‌داده‌های استفاده‌شده در این بخش شامل مجموعه‌داده‌های استاندارد دوبُعدی و هم‌چنین مجموعه‌داده‌های جهان واقعی است که مجموعه‌داده‌های دوبُعدی در خوشه‌بندی‌های مبتنی بر چگالی استفاده می‌شوند[19,20]. توضیحات مختصری راجع به مجموعه‌داده‌ها در جدول (۲) ارائه شده است. تعداد خوشه و هم‌چنین برچسب هر داده در هر مجموعهٔ‌داده مشخص است.

در ابتدا الگوریتم MDD-ISB-DBSCAN را بر روی تمامی مجموعه‌داده‌های ارائه‌شده در جدول (۲) اجرا و نتایج را بررسی می‌کنیم. از آنجا که الگوریتم MDD-ISB-DBSCAN، الگوریتم ISB-DBSCAN و الگوریتم ISDBSCAN دارای پارامتر ورودی هستند و تغییر این پارامترها می‌تواند در جواب خوشه‌بندی مؤثر باشد، پس تنظیم این پارامترها به‌صورت دستی و توسط کاربر صورت گرفته و بهترین جواب انتخاب شده است.

(جدول-۲): تعریف مجموعه داده‌ها
(Table-2): Definition of datasets

بانک داده	تعداد نمونه	تعداد ویژگی	تعداد خوشه
Flame	240	2	2
Jain	373	2	2
R15	600	2	15
D31	3100	2	31
Spiral	312	2	3
Pathbased	300	2	3
Compound	399	2	6
Aggregation	788	2	7
Iris	150	4	3
Wine	178	13	3
Heart	270	13	2

برچسب هر داده در این آزمایش از قبل وجود دارد بر این اساس نرخ درستی تعیین برچسب خوشه‌بندی با فرمول (۳) اندازه گیری می‌شود.

(۳)

$$\text{Correct rate}=\frac{M}{N}\times 100\%$$

-در فرمول (۳) M تعداد صحیح داده‌های برچسب‌گذاری‌شده و N تعداد کل داده‌ها در مجموعهٔ داده است.

نتایج ارزیابی الگوریتم MDD-ISB-DBSCAN در جدول (۳) نشان داده شده است. همان‌طور که مشاهده می‌شود نتایج در روش MDD-ISB-DBSCAN نسبت به روش‌های موجود بهبود یافته است. یکی از عوامل مؤثر در تعیین بهبود روش MDD-ISB-DBSCAN، استفاده از تعداد IS_K در دادهٔ هسته است. این روش در مجموعه داده‌هایی که دارای خوشه‌های نزدیک به هم بوده و یا خوشه‌هایی که تنها با چند داده از یکدیگر جدا شده‌اند مؤثر است. همچنین این روش در شناسایی داده‌های لبه نیز می‌تواند مؤثر واقع شود.

همان‌طور که در شکل (۹) مشاهده می‌شود در مجموعه‌داده‌هایی همچون Flame روش MDD-ISB-DBSCAN، می‌تواند به‌خوبی خوشه‌هایی را که به هم متصل بوده‌اند، تشخیص دهد؛ اما در الگوریتم ISB-DBSCAN از آنجا که تعیین دادهٔ هسته با مقدار بزرگتر از $2k/3$ است و با توجه به این که داده‌ها به یکدیگر متصل بوده و دادهٔ هسته در

بیشتر k های ورودی توسط کاربر، تمام داده‌ها را شامل می‌شود؛ پس قادر به تشخیص صحیح این خوشه‌ها نیست. در مجموعه‌داده‌های جهان واقعی نیز الگوریتم MDD-ISB-DBSCAN به‌طورمعمول بهتر عمل می‌کند.

روش MDD-ISB-DBSCAN در مجموعهٔ دادهٔ D31 که دارای خوشه‌های نزدیک به هم بوده و همچنین داده‌ها دارای فشردگی بالایی هستند، عملیات خوشه‌بندی را با درصد خطای کم انجام داده است. در مجموعهٔ دادهٔ Aggregation الگوریتم ISB-DBSCAN توانسته است شش خوشه را تشخیص دهد؛ اما در روش MDD-ISB-DBSCAN هفت خوشه به‌درستی تشخیص داده شده‌اند. روش MDD-ISB-DBSCAN در مجموعه داده‌های مختلف در مقایسه با دیگر روش‌های موجود نیز بهتر عمل کرده است.

(جدول-۳): نرخ صحیح برچسب داده‌ها در ISB-DBSCAN، ISDBSCAN و روش MDD-ISB-DBSCAN

(Table-3): The correct rate of data label in ISB-DBSCAN, ISDBSCAN and the MDD-ISB-DBSCAN method

MDD-ISB-DBSCAN	IS DBSCAN	ISB-DBSCAN	بانک داده
99.58	63.33	64.58	Flame
100	**100**	**100**	Jain
99.66	99	**99.66**	R15
96.93	78	80	D31
100	**100**	**100**	Spiral
94.33	83	84.33	Pathbased
94.98	89.47	**94.98**	Compound
99.87	95.68	95.30	Aggregation
92.66	66.66	**99.78**	Iris
69.66	55.05	55.61	Wine
61.48	55.55	55.55	Heart
91.74	80.52	84.52	میانگین

در روش PMDD-ISB-DBSCAN طبق جدول (۴)، از هشت مجموعهٔ داده استفاده شده است. این روش به‌دلیل استفاده از معیار فاصله در مجموعه داده‌هایی با بُعد بالا ضعیف عمل می‌کند. به همین دلیل این مجموعه داده‌ها در این روش مورد استفاده قرار نمی‌گیرند.

همان‌طور که مشاهده می‌شود در این روش نرخ صحیح برچسب داده‌ها از روش‌های قبلی کمتر است. این روش در مجموعهٔ داده D31 نامناسب عمل کرده است. یکی از دلایلی که باعث وجود این مشکل شده وجود تابع برازش پیشنهادی است. از آنجا که تابع برازش پیشنهادی سعی در حذف بزرگ‌ترین یال دارد، باعث می‌شود تا خوشه‌هایی به‌وجود آیند که کم‌ترین فاصلهٔ درون‌خوشه‌ای را داشته باشند. به همین دلیل این روش باعث می‌شود تا تعداد زیادی خوشهٔ کوچک به تعداد کمتری خوشه با تعداد داده‌های بیشتر ترجیح داده شود.

روش PMDD-ISB-DBSCAN در مجموعه‌دادهٔ Jain دارای مقداری خطا است. این خطا به‌دلیل وجود یک یال بزرگ است که دو خوشه را به هم متصل کرده است. مجموع یال‌های این درخت در مقایسه با بزرگترین یال درخت باعث می‌شود تا در فرمول (۲) مقدار ۱ برگشت داده شود بنابراین خوشهٔ اشتباهی ایجاد می‌شود. در شکل (۱۰) نمودارهای هم‌گرایی الگوریتم ژنتیک در مجموعه‌داده‌های مختلف رسم شده است.

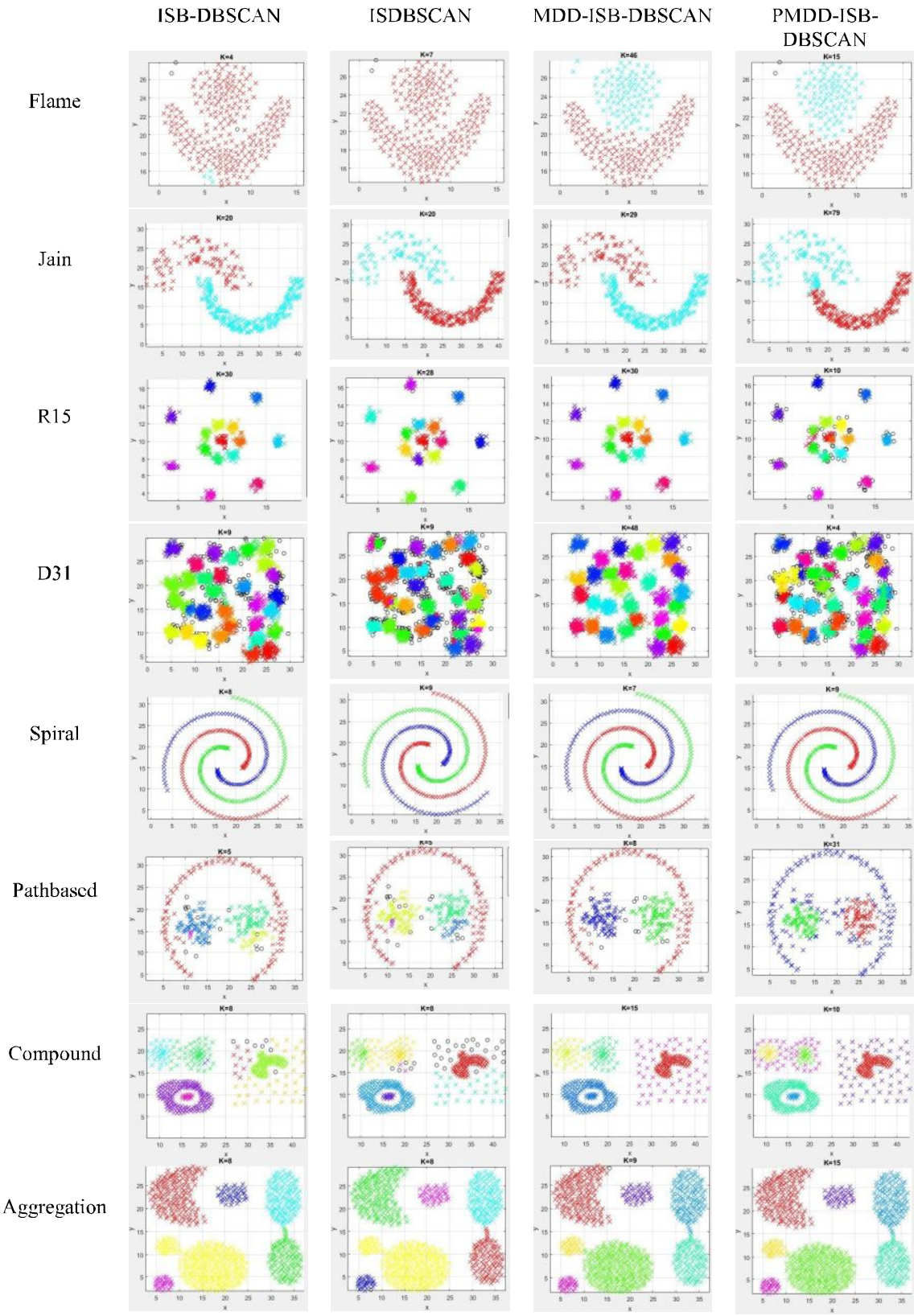
یکی از دلایلی که باعث کاهش نرخ صحیح برچسب داده‌ها شده نبود هیچ دانش قبلی از مجموعهٔ داده است؛ اما همان‌طور که در شکل (۹) مشاهده می‌شود، این روش توانسته است، خوشه‌ها را به‌طورتقریبی از یکدیگر جدا کند.

دلیل استفاده از این روش و برتری این روش نسب به روش‌های دیگر، خودکارسازی انجام خوشه‌بندی بدون دخالت کاربر است. ممکن است کاربری که قصد انجام خوشه‌بندی روی داده‌ها را داشته باشد، هیچ دانش قبلی از مجموعهٔ داده، تعداد خوشه‌ها، چگالی داده‌ها و ... نداشته باشد؛ در این زمان می‌توان با کمی درصد خطای بیشتر، از روش تنظیم خودکار پارامتر استفاده کرد و نتایج قابل قبولی را دریافت کرد. یکی دیگر از عواملی که باید در این روش به آن توجه کرد، ابعاد مجموعهٔ داده است. از آنجا که این روش از بزرگ‌ترین یال هر خوشه برای یافتن چگالی و همچنین فاصلهٔ درون‌خوشه‌ای کم استفاده می‌کند، پس استفاده از آن در مجموعه‌داده‌هایی با ابعاد دو مناسب‌تر است.

(جدول-۴): نرخ صحیح برچسب داده‌ها در ISB-DBSCAN، ISDBSCAN و روش PMDD-ISB-DBSCAN

(Table-4): The correct rate of data label in ISB-DBSCAN, ISDBSCAN and PMDD-ISB-DBSCAN method

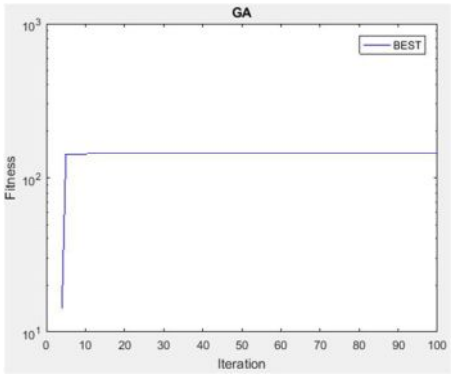
PMDD-ISB-DBSCAN	IS DBSCAN	ISB-DBSCAN	بانک داده
97.5	63.33	64.58	Flame
92.22	**100**	**100**	Jain
84.16	99	**99.66**	R15
35.48	78	**80**	D31
100	**100**	**100**	Spiral
83.66	83	**84.33**	Pathbased
89.97	89.47	**94.98**	Compound
99.74	95.68	95.30	Aggregation
85.34	88.56	**89.85**	میانگین



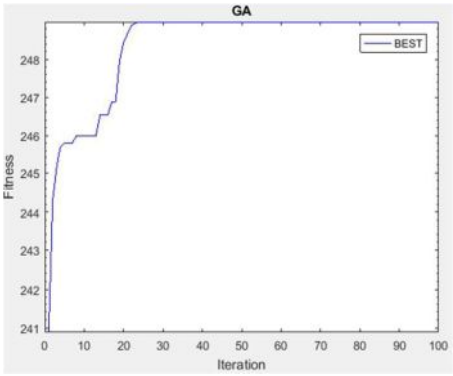
(شکل-۹): مقایسهٔ روش‌های مختلف در تشخیص صحیح خوشه‌ها

(Figure-9): Comparison of different methods for the correct identification of cluster

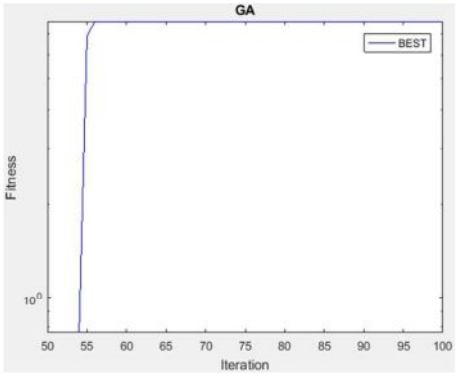
Flame



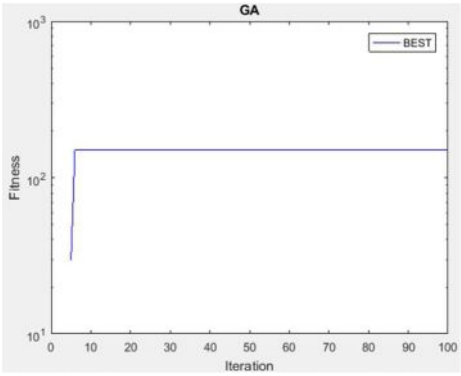
Jain



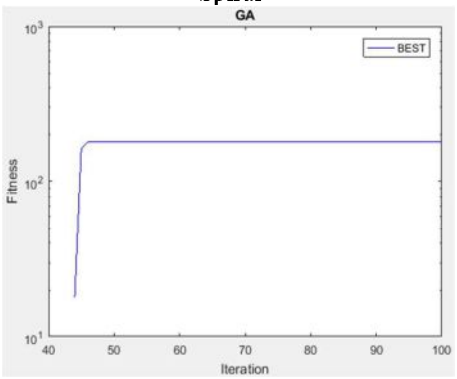
R15



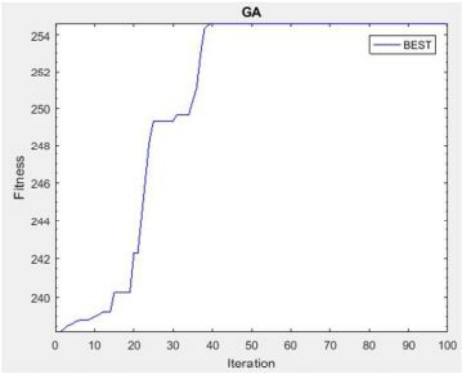
D31



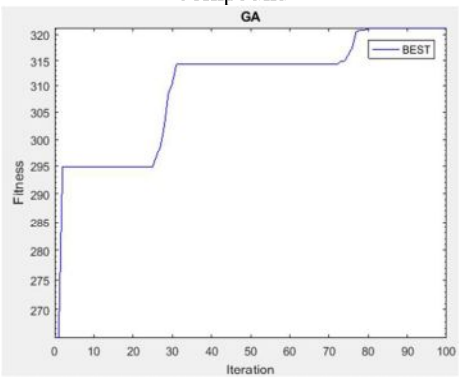
Spiral



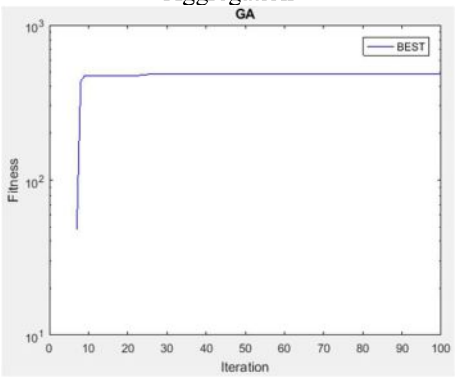
Pathbased



Compound



Aggregation



(شکل-۱۰): نمودارهای هم‌گرایی مجموعه داده‌های مختلف در الگوریتم ژنتیک

(Figure-10): Convergence diagrams of different dataset in the genetic algorithm

۶-نتیجه‌گیری

در این مقاله یک روش خوشه‌بندی مبتنی بر ISB-DBSCAN ارائه شد که توانست به‌طور قابل توجهٔ نتایج را بهبود بخشد. این روش مبتنی بر k نزدیک‌ترین همسایه عمل می‌کند و در سه مرحله انجام می‌شود. در مرحلهٔ نخست برای تمامی داده‌ها فضای نفوذ یا IS_K محاسبه می‌شود. در مرحلهٔ دوم تمامی داده‌های هسته‌ای که خوشه‌های اصلی را تشکیل می‌دهند، ایجاد می‌شوند. در مرحلهٔ سوم داده‌هایی که برچسب ۱- دارند، براساس فهرست همسایگی در IS_K خوشه‌بندی می‌شود. یکی از تفاوت‌های این روش با الگوریتم ISB-DBSCAN تولید خوشه جدید در مرحله سوم و هم‌چنین تعریف جدیدی از دادهٔ هسته است.

در قسمت دوم یک روش پیشنهادی برای تنظیم خودکار پارامتر ورودی ارائه شد. این روش می‌تواند به‌طور خودکار براساس مجموعهٔ داده بهترین پارامتر ورودی را با الگوریتم ژنتیک و تابع برازش پیشنهادی تشخیص دهد. یکی از مشکلات این روش درصد خطای بیشتر نسبت به دیگر روش‌ها است؛ اما از آنجا که ممکن است، کاربر در تعیین پارامتر k دچار مشکل شود، می‌تواند مفید واقع شود و با درصد خطای بیشتر نسبت به روش MDD-ISB-DBSCAN خوشه‌بندی مورد قبولی را ایجاد کند.

۷- مراجع 7- References

[1] C. Cassisi, A. Ferro, R. Giugno, G. Pigola, and A. Pulvirenti, "Enhancing density-based clustering: Parameter reduction and outlier detection," *Information Systems,* vol. 38, no. 3, pp. 317-330, 2013.

[2] X. Chen, W. Liu, H. Qiu, and J. Lai, "APSCAN: A parameter free algorithm for clustering," *Pattern Recognition Letters,* vol. 32, no. 7, pp. 973-986, 2011.

[3] H. Durnová, "Otakar Boruvka (1899-1995) and the Minimum Spanning Tree," 1998.

[4] M. T. Elbatta and W. M. Ashour" , A dynamic method for discovering density varied clusters," *International Journal of Signal Processing, Image Processing and Pattern Recognition,* vol. 6, no. 1, pp. 123-134, 2013.

[5] J. Esmaelnejad, J. Habibi, and S .H. Yeganeh, "A novel method to find appropriate ε for DBSCAN," in *Asian Conference on Intelligent Information and Database Systems*, Springer, 2010, pp. 93-102.

[6] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Kdd*, 1996, vol. 96, no. 34, pp. 226-231.

[7] R. L. Graham and P. Hell, "On the history of the minimum spanning tree problem," *Annals of the History of Computing,* vol. 7, no. 1, pp. 43-57, 1985.

[8]M. N. Gaonkar and K. Sawant, "AutoEpsDBSCAN: DBSCAN with Eps automatic for large dataset," *International Journal on Advanced Computer Theory and Engineering*, vol. 2, no. 2, pp. 11-16, 2013.

[9] J. Hou, H. Gao, and X. Li, "Dsets-dbscan: a parameter-free clustering algorithm," *IEEE Transactions on Image Processing,* vol. 25, no. 7, pp. 3182-3193, 2016.

[10] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100-108, 1979.

[11] A. Karami and R. Johansson, "Choosing dbscan parameters automatically using differential evolution," *International Journal of Computer Applications,* vol. 91, no. 7, 2014.

[12] Y. Lv *et al.*, "An efficient and scalable density-based clustering algorithm for datasets with complex structures," *Neurocomputing,* vol. 171, pp. 9-22, 2016.

[13] M. Mitchell, *An introduction to genetic algorithms*. MIT press, 1998.

[۱۴] پورمحمدی سارا، مالکی علی، "تشخیص پیوسته میزان استرس در طول رانندگی با استفاده از روش خوشه‌بندی Fuzzy c-means"، پردازش علائم و داده‌ها، ۱۴ (۴) :۱۴۲-۱۲۹، ۱۳۹۶.

[14] S. Pourmohammadi, A. Maleki, "A Fuzzy C-means Clustering Approach for Continuous Stress Detection during Driving", JSDP; 14 (4) :129-142,2018.

[15] S. Mitra and J. Nandy, "Kddclus: A simple method for multi-density clustering," *SKAD'11–Soft Computing Applications and Knowledge Discovery,* pp. 72, 2011.

[16] G. H. Shah, "An improved DBSCAN, a density based clustering algorithm with parameter selection for high dimensional data sets," *IEEE Engineering (NUiCONE), 2012 Nirma University International Conference on*, 2012, pp. 1-6.

[17] K. Sawant, "Adaptive methods for determining dbscan parameters," *International Journal of Innovative Science, Engineering & Technology,* vol. 1, no. 4, 2014.

[18] P. Sharma and Y. Rathi, "Efficient Density-Based Clustering Using Automatic Parameter Detection," in *Proceedings of the International Congress on Information and Communication Technology*, Springer, 2016, pp. 433-441.

[19] https://cs.joensuu.fi/sipu/datasets/

[20]http://archive.ics.uci.edu/ml/datasets/statlog+(heart)

علیرضا پهلوانزاده، فارغ‌التحصیل کارشناسی ارشد رشته هوش مصنوعی دانشگاه شهید باهنر کرمان است. زمینهٔ پژوهشی مورد علاقه ایشان داده‌کاوی است. نشانی رایانامه ایشان عبارت است از:
pahlevanzadeh@eng.uk.ac.ir

علی‌اکبر نیک‌نفس، عضو هیأت علمی و دانشیار بخش مهندسی کامپیوتر دانشگاه شهید باهنر کرمان، تحصیلات خود را در دانشگاه‌های شیراز، تربیت مدرس و شهید باهنر کرمان به پایان رسانده و در سال‌های اخیر در حوزه‌های پژوهشی مرتبط با داده‌کاوی و هوش تجاری فعالیت می‌کند.
نشانی رایانامه ایشان عبارت است از:
niknafs@uk.ac.ir