



دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

دانشکده مهندسی کامپیوuter و فناوری اطلاعات

پایان نامه کارشناسی ارشد

گرایش هوش مصنوعی

کشف داده‌های پرت محلی در کلان‌داده‌ها با استفاده از یک روش مبتنی  
بر چگالی

نگارش

سید احمد نقوی نوزاد

استاد راهنما

دکتر مریم امیرحائری

اسفندماه ۱۳۹۶



دانشگاه صنعتی امیرکبیر

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

پایان نامه کارشناسی ارشد

گرایش هوش مصنوعی

کشف داده‌های پرت محلی در کلان‌داده‌ها با استفاده از یک روش مبتنی بر چگالی

نام و نام خانوادگی: سید احمد نقوی نوزاد شماره دانشجویی: ۹۴۱۳۱۰۶۰ مقطع: کارشناسی ارشد

این پایان نامه توسط هیئت داوران زیر در تاریخ ۲۰ / ۱۲ / ۱۳۹۶ به تصویب رسیده است:

امضا:

استاد راهنما: دکتر مریم امیرحائزی

امضا:

داور داخلی: دکتر محمد رحمتی

امضا:

داور خارجی: دکتر بابک نجار اعرابی

به نام خدا

## تعهدنامه اصالت اثر

تاریخ:



اینجانب سید احمد نقوی نوزاد متعهد می‌شوم که مطالب مندرج در این پایان نامه حاصل کار پژوهشی اینجانب تحت نظارت و راهنمایی استادی دانشگاه صنعتی امیرکبیر بوده و به دستاوردهای دیگران که در این پژوهش از آنها استفاده شده است مطابق مقررات و روال متعارف ارجاع و در فهرست منابع و مأخذ ذکر گردیده است. این پایان نامه قبلاً برای احراز هیچ مدرک هم‌سطح یا بالاتر ارائه نگردیده است.

در صورت اثبات تخلف در هر زمان، مدرک تحصیلی صادر شده توسط دانشگاه از درجه اعتبار ساقط بوده و دانشگاه حق پیگیری قانونی خواهد داشت.

کلیه نتایج و حقوق حاصل از این پایان نامه متعلق به دانشگاه صنعتی امیرکبیر می‌باشد. هرگونه استفاده از نتایج علمی و عملی، واگذاری اطلاعات به دیگران یا چاپ و تکثیر، نسخه‌برداری، ترجمه و اقتباس از این پایان نامه بدون موافقت کتبی دانشگاه صنعتی امیرکبیر ممنوع است.  
نقل مطالب با ذکر مأخذ بلامانع است.

سید احمد نقوی نوزاد

امضا

تعدیم با بوسه بر دستان پدر م

به او که نمی دانم از بزرگی اش بکویم یا مردانگی، سخاوت، سکوت، مهربانی و....

تعدیم به مادر عزیزتر از جانم

مادرم هستی من ز هستی توست تا هستم و هستی دارست دوست

## چکیده

با توجه این که امروزه حجم داده‌هایی که به طُرُق مختلف جمع‌آوری شده و ذخیره می‌گردند، به طرز فزاینده‌ای رو به افزایش می‌باشد، لذا روش‌های متداول نرمافزاری جهت پردازش و مدیریت این حجم داده، کارائی لازم را نداشته و به همین سبب این نوع داده‌ها را در دسته‌بندی دیگری با نام کلان‌داده قرار می‌دهند. حجم و ابعاد بسیار بالای کلان‌داده‌ها، سبب می‌شود تا نیاز به روش‌ها و فناوری‌های نوینی که توانایی پردازش این گونه داده‌ها را دارند، بیشتر احساس شود. کشف داده‌های پرت نیز از جمله چالش‌های مطرح در زمینه‌ی کلان‌داده‌ها می‌باشد. کشف داده‌های پرت، معمولاً می‌تواند در قالب یک مرحله‌ی پیش‌پردازش داده‌ها در نظر گرفته شود که در آن تلاش می‌شود تا تعداد داده‌های اندکی که نسبت به سایر داده‌های معمول، رفتار قابل قبولی از خود بروز نمی‌دهند، شناسائی شوند.

در این پایان‌نامه قصد داریم تا روشی را جهت کشف داده‌های پرت محلی در کلان‌داده‌ها ارائه نمائیم که بنای آن بر خوشبندی مبتنی بر چگالی و مقیاس‌پذیر می‌باشد. با توجه به این که یک مجموعه‌داده‌ی کلان، قابلیت جاگرفتن در RAM را به یکباره دارا نیست، لذا ناچاریم تا آن را به صورت قطعه‌قطعه پردازش نمائیم، به گونه‌ای که هر قطعه از کلان‌داده در آن واحد، هم قابلیت جاگرفتن و هم پردازش‌شدن در حافظه‌ی اصلی را داشته باشد. سپس به ازای هر قطعه، اطلاعات مربوط به مدل خوشبندی را به روز می‌نمائیم. در تمام طول رویه‌ی خوشبندی، تلاش ما آن است تا داده‌های پرت در تشکیل و به‌روزرسانی خوش‌ها نقشی ایفا نکنند. در پایان امر خوشبندی مقیاس‌پذیر، ساختار خوش‌های نهائی را به دست می‌آوریم. در ادامه، با استفاده از یک معیار مناسب، به هر داده، امتیازی مبنی بر میزان پرت‌بودن نسبت خواهیم داد. نتایج آزمایشات بر روی مجموعه‌داده‌های واقعی و مصنوعی نشان می‌دهند که روش پیشنهادی ما، پیچیدگی زمانی خطی پائینی دارد و نسبت به سایر روش‌های سنتی که نیاز دارند تا مجموعه‌داده را به یکباره و نه به صورت قطعه‌قطعه مشاهده نمایند، هم اثربخش بوده و هم از بازدهی بالائی در مورد شرایط حاد برخوردار می‌باشد.

## واژه‌های کلیدی:

کلان‌داده، کشف داده‌های پرت، مقیاس‌پذیر، مبتنی بر چگالی.

## صفحه

## فهرست مطالب

<b>۱</b>	<b>فصل اول مقدمه</b>
۲	مسئله.....
۴	ضرورت انجام طرح.....
۴	چالش‌های موجود.....
۵	فرضیات.....
۶	هدف از اجراء.....
۶	نوآوری‌ها.....
۸	ساختار پایان‌نامه.....
<b>۹</b>	<b>فصل دوم مفاهیم پایه</b>
۱۰	داده‌ی پرت چیست؟.....
۱۲	پیدایش داده‌های پرت.....
۱۴	انواع داده‌های پرت.....
۱۴	داده‌های پرت سراسری.....
۱۵	داده‌های پرت حیطه‌ای (وابسته به قرائن) یا شرطی.....
۱۷	داده‌های پرت تجمعی.....
۱۸	تفاوت داده‌های پرت با داده‌های نوبزی.....
۲۲	انواع روش‌های کشف داده‌های پرت.....
۲۳	مدل‌های مبتنی بر مقادیر کرانی.....
۲۴	مدل‌های مبتنی بر احتمالات.....
۲۵	مدل‌های مبتنی بر خوشبندی.....
۲۶	مدل‌های مبتنی بر فاصله.....
۲۷	مدل‌های مبتنی بر چگالی.....
۲۸	مدل‌های مبتنی بر تئوری اطلاعات.....
۲۹	انواع خروجی روش‌های کشف داده‌های پرت.....
۲۹	اهمیت و ضرورت کشف داده‌های پرت.....
۳۲	شباهت رویه‌ی کشف داده‌های پرت با رویه‌ی کشف اقلام نوظهور.....
۳۳	چالش‌های موجود در زمینه‌ی کشف داده‌های پرت.....
<b>۳۶</b>	<b>فصل سوم مروری بر کارهای انجام‌شده</b>
۱,۳	کشف داده‌های پرت محلی با استفاده از یک روش مبتنی بر چگالی و معرفی معیار بنیادی LOF.....
۲,۳	ضریب داده‌ی پرت محلی با مقدار احتمالاتی مابین صفر و یک (LoOP).....
۴,۴	کشف داده‌های پرت محلی در جریان داده‌ها با استفاده از یک روش افزایشی.....

۴۵.....	۴.۳ جنگل جداسازی.....
۴۶.....	۵.۳ ماشین‌های بردار پشتیبان تک کلاسه.....
۴۷.....	۶.۳ کشف داده‌های پرت در مجموعه داده‌های نامی و با مقیاس بزرگ با استفاده از یک رویکرد مبتنی بر تئوری اطلاعات.....
۴۹.....	۱.۶.۳ شرح روش و پارامترها.....
۴۹.....	۲.۶.۳ آنتروپی و همبستگی تام.....
۵۱.....	۳.۶.۳ آنتروپی تام روی بردار تصادفی Y.....
۵۲.....	۴.۶.۳ وزن دار کردن ویژگی‌ها.....
۵۴.....	۵.۶.۳ آنتروپی تام وزن دار روی بردار تصادفی Y.....
۵۴.....	۶.۶.۳ یک تعریف رسمی از مسئله‌ی کشف داده‌های پرت.....
۵۵.....	۷.۶.۳ یک مفهوم جدید از ضریب داده‌ی پرت.....
۵۶.....	۱,۷,۶,۳ آنتروپی تام تفاضلی.....
۵۷.....	۲,۷,۶,۳ ضریب داده‌ی پرت.....
۵۹.....	۸.۶.۳ به روزرسانی ضریب داده‌ی پرت.....
۶۰.....	۹.۶.۳ تعیین یک حد بالا برای تعداد داده‌های پرت.....
۶۱.....	۱۰.۶.۳ معرفی الگوریتم‌های ITB-SP و ITB-SS.....
۶۳.....	۷.۳ کشف داده‌های پرت محلی در داده‌های با مقیاس بزرگ با استفاده از یک روش کاهش بُعد در ضمن حفظ چگالی داده‌ها.....
۶۴.....	۱.۷.۳ شرح روش و پارامترها.....
۶۴.....	۲.۷.۳ تحلیل مؤلفه‌ی اصلی (PCA).....
۶۵.....	۳.۷.۳ تصویرسازی تصادفی (RP).....
۶۶.....	۴.۷.۳ تصویرسازی تصادفی ضمن حفظ توزیع چگالی درونی داده‌ها.....
۶۷.....	۱,۴,۷,۳ محفوظماندن فاصله‌ی یک داده تا K-نزدیک‌ترین همسایه‌ی آن تحت تصویرسازی تصادفی.....
۶۸.....	۲,۴,۷,۳ محفوظماندن مجموعه‌های همسایگی هر داده در فضای کاهش بعديافتة با توجه به یک سری شرایط خاص.....
۶۹.....	۵.۷.۳ نزدیک‌ترین همسایگان حاصله از شاخص تصویرسازی (PINN).....
۷۲.....	۸.۳ کشف داده‌های پرت مکانی با استفاده از مدل یادگیری خودسازمان‌دهنده‌ی تکراری و تخمین فاصله‌ی مستحکم.....
۷۵.....	۱.۸.۳ شرح روش و پارامترها.....
۷۵.....	۲.۸.۳ تجمعی شبكه‌ی عصبی SOM با تخمین فاصله‌ی مستحکم جهت کشف داده‌های پرت مکانی.....
۷۸.....	۱,۲,۸,۳ فاصله‌ی ماهalanobis.....
۷۹.....	۲,۲,۸,۳ معیار فاصله‌ی مستحکم مبتنی بر کمینه‌ی دترمینان ماتریس کوواریانس.....
۸۱.....	۴ فصل چهارم روش پیشنهادی.....
۸۶.....	۱.۴ مقدمات و پیش‌زمینه‌های لازم.....
۸۶.....	۱.۱.۴ الگوریتم خوشبندی Kmeans.....

۸۸	معیار فاصله‌ی ماهالانوبیس.....	۲,۱,۴
۸۹	الگوریتم خوشه‌بندی مقیاس‌پذیر BFR	۳,۱,۴
۹۴	الگوریتم خوشه‌بندی DBSCAN	۴,۱,۴
۹۷	روش کاهش بعد تحلیل مؤلفه‌ی اصلی (PCA) .....	۵,۱,۴
۹۹	الگوریتم بهینه‌سازی انبوه ذرات (PSO) .....	۶,۱,۴
۱۰۱	روش پیشنهادی.....	۲,۴
۱۰۳	نمونه‌برداری.....	۱,۲,۴
۱۰۹	خوشه‌بندی مقیاس‌پذیر.....	۲,۲,۴
۱۰۹	۱,۲,۲,۴ بروزرسانی مدل خوشه‌بندی موقت با توجه به یک قطعه‌ی داده.....	
۱۱۰	۱,۱,۲,۲,۴ بررسی امکان تعلق داده‌های یک قطعه به خوشه‌های موقت.....	
۱۱۱	۲,۱,۲,۲,۴ بروزرسانی اطلاعات حیاتی خوشه‌های موقت.....	
۱۱۱	۳,۱,۲,۲,۴ بررسی امکان تعلق داده‌های معلق موجود در RAM به خوشه‌های بروزشده.....	
۱۱۲	۴,۱,۲,۲,۴ خوشه‌بندی داده‌های معلق موجود در RAM.....	
۱۱۶	۵,۱,۲,۲,۴ بررسی امکان تعلق داده‌های آخرین قطعه به خوشه‌های موقت.....	
۱۱۷	۲,۲,۲,۴ ساخت مدل خوشه‌بندی نهائی.....	
۱۲۲	۳,۲,۴ امتیازدهی.....	
۱۲۴	۴,۲,۴ پیچیدگی الگوریتم پیشنهادی.....	
۱۲۷	<b>۵ فصل پنجم نتایج آزمایشات انجام شده.....</b>	
۱۲۸	۱,۵ روش‌های رقیب و طرح کلی آزمایشات.....	
۱۲۹	۲,۵ آزمایش اثربخشی.....	
۱۲۹	۱,۲,۵ آزمایش بر روی مجموعه‌داده‌های واقعی.....	
۱۳۳	۲,۲,۵ آزمایش بر روی مجموعه‌داده‌های مصنوعی.....	
۱۳۵	۳,۵ آزمایش بازدهی.....	
۱۳۸	۴,۵ آزمایش پیچیدگی زمانی.....	
۱۴۰	۵,۵ آزمایش نرخ نمونه‌برداری.....	
۱۴۳	<b>۶ فصل ششم جمع‌بندی و نتیجه‌گیری.....</b>	
۱۴۴	۱,۶ جمع‌بندی و نتیجه‌گیری.....	
۱۴۶	<b>۷ منابع و مراجع.....</b>	

## صفحه

## فهرست اشکال

شکل ۱.۲	یک مثال ساده از داده‌های پرت در یک مجموعه‌داده‌ی دوبعدی.....	۱۱
شکل ۲.۲	یک مثال ساده از داده‌های پرت تجمعی.....	۱۷
شکل ۳.۲	تفاوت میان داده‌های نویزی و داده‌های پرت.....	۱۹
شکل ۴.۲	تفاوت میان داده‌های نرمال و داده‌های نویزی.....	۲۱
شکل ۵.۲	طیف امتیاز یک داده به لحاظ میزان پرتبودن از نرمال تا پرت.....	۲۲
شکل ۱.۳	فاصله‌ی دسترس پذیری داده‌های $p_1$ و $p_2$ با توجه به داده‌ی ۰.....	۳۸
شکل ۲.۳	نمودارتابع سیگموئید معکوس برای نسبت وزن ویژگی به میزان آنتروپی آن.....	۵۳
شکل ۳.۳	نمودارتابع $\delta(x)$ .....	۵۸
شکل ۴.۳	یک نمونه از تغییر در مجموعه‌ی همسایگی پس از تصویرسازی تصادفی.....	۷۰
شکل ۵.۳	مراحل الگوریتم PINN.....	۷۱
شکل ۱.۴	نمونه‌ای از خوشه‌های دارای توزیع نرمال در فضای دوبعدی.....	۸۳
شکل ۲.۴	فرض اولیه‌ی قوى الگوريتم BFR در مورد خوشه‌ها.....	۹۰
شکل ۳.۴	نمایشی از نقاط سه مجموعه‌ی نادیده گرفته شده، فشرده شده و نگهداری شده.....	۹۲
شکل ۴.۴	نمونه‌ای خوشبندی توسط الگوريتم DBSCAN.....	۹۶
شکل ۵.۴	اعمال PCA بر روی یک توزیع گاوین چندمتغیره.....	۹۸
شکل ۶.۴	نمائی از فاز نمونه‌برداری توسط روش پیشنهادی.....	۱۰۵
شکل ۷.۴	نمائی از منحنی‌های تراز ماهالانوبیس به ازای یک خوشه‌ی آلوهه.....	۱۰۸
شکل ۸.۴	نمونه‌ای از شکستن یک خوشه‌ی غیرمحدب به زیرخوشه‌های محدب.....	۱۱۳
شکل ۹.۴	نمونه‌ای از مدل خوشبندی موقت مربوط به مدل پیشنهادی.....	۱۱۷
شکل ۱۰.۴	نمونه‌ای از مدل خوشبندی نهائی مربوط به روش پیشنهادی.....	۱۲۰
شکل ۱۱.۴	نمونه‌ای از داده‌های بازسازی شده به ازای هر خوشه‌ی نهائی توسط روش پیشنهادی.....	۱۲۲
شکل ۱۲.۴	نمائی از منحنی‌های تراز ماهالانوبیس به ازای تعدادی خوشه‌ی نرمال.....	۱۲۳
شکل ۱.۵	نمودارهای هیستوگرام مربوط به مؤلفه‌های اصلی خوشه‌های یک مجموعه‌داده‌ی واقعی.....	۱۳۲
شکل ۲.۵	نتایج آزمایش بازدهی بر روی مجموعه‌داده‌های مصنوعی.....	۱۳۶
شکل ۳.۵	نمودار زمان مصرفی به ازای افزایش تعداد داده‌های یک مجموعه‌داده.....	۱۳۹
شکل ۴.۵	نمودار تغییرات $\det \mathbb{L}$ یک خوشه‌ی نمونه‌برداری شده به ازای نرخهای نمونه‌برداری متفاوت.....	۱۴۰
شکل ۵.۵	وضعیت منحنی‌های تراز ماهالانوبیس به ازای یک خوشه‌ی نمونه‌برداری شده.....	۱۴۱

صفحه

## فهرست جداول

جدول ۱.۵ نتایج AUC حاصل از آزمایش روش پیشنهادی و روش‌های رقیب..... ۱۳۰

## فهرست علائم

### علائم لاتین

مجموعه داده	D, X, Y
نقاط دلخواه در فضا	p, q, o
تابع فاصله	d
تعداد نزدیکترین همسایگان	k
ماتریس تبدیل	R, A
آنتروپی	H
اطلاعات دو طرفه	I
همبستگی تام	C
مجموعه همسایگی یک داده	S

### علائم یونانی

شعاع همسایگی	$\epsilon$
حداقل تعداد نقاط برای تشکیل یک ناحیه‌ی چگال	$\mu$
ماتریس کوواریانس	$\Sigma$
شعاع همسایگی ماهalanوبیس مجاز برای تعلق	$\alpha$
شعاع همسایگی ماهalanوبیس برای هرس کردن	$\beta$
ضریب انحراف از معیار	$\lambda$

۱

## فصل اول

### مقدمه

## ۱.۱ مسئله

امروزه، بسیاری از کاربردهای موجود، دارای داده‌هایی با حجم‌های بسیار انبوه هستند که به اصطلاح «کلان‌داده»<sup>۱</sup> نامیده می‌شوند و پردازش آن‌ها از اهمیت بسیار بالائی برخوردار است. کلان‌داده، یک واژه‌ی عمومی برای مجموعه‌داده‌های بسیار بزرگ و پیچیده‌ای است که برنامه‌های پردازش داده‌ی سنتی، برای تحلیل آن‌ها از شرایط کافی برخوردار نیستند.

کشف «داده‌های پرت»<sup>۲</sup> (یا همان کشف «ناهنجری»<sup>۳</sup>)، یک مسئله‌ی بنیادی در علم «داده‌کاوی»<sup>۴</sup> به حساب می‌آید. داده‌های پرت، به داده‌هایی اطلاق می‌گردد که از حالت نرمال منحرف گشته و کشف آن‌ها غالباً با «یافتن یک سوزن در انبار کاه» مقایسه می‌گردد. داده‌های پرت، روند آموزش اکثربیت مدل‌ها از روی داده‌ها را با خدشه مواجه کرده و در نتیجه، یافتن آن‌ها می‌تواند باعث افزایش دقت و کاهش سربار محاسباتی شود و بنابراین از اهمیت بسیار بالائی برخوردار می‌باشد. روش‌های کشف داده‌های پرت، علاوه بر قابلیت پیش‌پردازش داده‌ها، می‌توانند به عنوان روش‌های تشخیص ناهنجاری برای کاربردهایی نظیر کشف تقلب و کشف نفوذ هم به کار بردند [۱].

داده‌های پرت را می‌توان به طور کلی، به دو دسته‌ی «سراسری»<sup>۵</sup> و «محلي»<sup>۶</sup> تقسیم نمود. با توجه به اینکه داده‌های پرت محلی، لزوماً نسبت به کل داده‌ها رفتار غیر نرمال ندارند، لذا کشف آن‌ها با دشواری و سربار محاسباتی بیشتری مواجه می‌باشد [۱], [۲].

مدل‌هایی که تاکنون جهت کشف داده‌های پرت ارائه شده‌اند را می‌توان به شش دسته‌ی کلی تقسیم نمود: «مدل‌های مبتنی بر مقادیر کرانی»<sup>۷</sup>؛ «مدل‌های مبتنی بر احتمالات»<sup>۸</sup>؛ «مدل‌های مبتنی بر

<sup>1</sup> Big Data

<sup>2</sup> Outliers

<sup>3</sup> Anomaly

<sup>4</sup> Data mining

<sup>5</sup> Global

<sup>6</sup> Local

<sup>7</sup> Extreme Value Analysis

<sup>8</sup> Probabilistic Models

خوشبندی»<sup>۹</sup>؛ «مدل‌های مبتنی بر فاصله»<sup>۱۰</sup>؛ «مدل‌های مبتنی بر چگالی»<sup>۱۱</sup> و «مدل‌های مبتنی بر تئوری اطلاعات»<sup>۱۲</sup>.

روش‌های مبتنی بر مقدادیر کرانی، «دادگان»<sup>۱۳</sup> موجود را در قالب یک «توزیع احتمالاتی»<sup>۱۴</sup> در نظر گرفته و تنها نقاطی را که در دو طرف انتهائی این توزیع قرار دارند، به عنوان کاندید داده‌ی پرت در نظر می‌گیرند. در روش‌های مبتنی بر احتمالات، تصور می‌کنیم که داده‌های موجود، توسط یک «مدل مولد مخلوطی»<sup>۱۵</sup> تولید شده‌اند و از همین داده‌ها به عنوان عاملی برای تخمین پارامترهای مدل استفاده می‌شود. بعد از این‌که پارامترهای این مدل مولد، به درستی تشخیص داده شدن، داده‌های پرت همان داده‌هایی خواهند بود که احتمال و درستی این‌که توسط این مدل تولید شده باشند، بسیار پایین می‌باشد. روش‌های مبتنی بر خوشبندی، از یک تحلیل سراسری جهت شناسائی آن دسته از داده‌هایی که صلاحیت تشکیل یک خوشه را دارند، استفاده می‌نمایند. داده‌های پرت، آن داده‌هایی خواهند بود که نتوانسته‌اند به هیچ خوشه‌ای تعلق بگیرند. در روش‌های مبتنی بر فاصله، با توجه به این‌که داده‌های پرت، آن دسته از داده‌هایی می‌باشند که از نواحی چگال یا همان خوشه‌ها به اندازه‌ی کافی دور می‌باشند، از فاصله‌ی هر داده تا  $k$ -امین نزدیک‌ترین همسایه‌ی آن، به عنوان معیاری جهت تعیین میزان پرت‌بودن آن داده استفاده می‌شود. روش‌های مبتنی بر چگالی، با توجه به تحلیل چگالی توزیع داده‌ها عمل می‌نمایند تا یک امتیاز داده‌ی پرت محلی را برای هر نمونه داده، مبتنی بر چگالی همسایگی محلی برای آن داده مشخص نمایند. با توجه به این روش، آن نمونه‌داده‌هایی که امتیاز آن‌ها بالاتر است را می‌توان به عنوان داده‌ی پرت در نظر گرفت. روش‌های مبتنی بر تئوری اطلاعات، مبتنی بر این اصل عمومی هستند که تغییرات در اندازه‌ی یک مدل بررسی داده‌ی پرت را جهت توصیف میزان پرت‌بودن به ازای هر داده بررسی می‌نمایند [۳، ۴].

<sup>۹</sup> Clustering-based Models

<sup>۱۰</sup> Distance-based Models

<sup>۱۱</sup> Density-based Models

<sup>۱۲</sup> Information-Theoretic Models

<sup>۱۳</sup> Dataset (Database)

<sup>۱۴</sup> Probability distribution

<sup>۱۵</sup> Mixture-based Generative Model

## ۲.۱ ضرورت انجام طرح

از جمله کاربردهای حائز اهمیت تشخیص داده‌های پرت، پیش‌پردازش و یا همان پاکسازی داده‌هاست تا در نتیجه‌ی آن مدل‌هایی که از روی سیستم‌ها یاد گرفته می‌شوند، مدل‌های دقیق‌تر و کاراتری باشند. از آن‌جا که روش‌های معمول فعلی، جهت تشخیص رفتارهای غیر نرمال در مورد داده‌های با مقیاس بزرگ چندان کارائی ندارند، لذا یافتن راهی مناسب برای رفع این مشکل می‌تواند از ارزش بالائی برخوردار باشد. از دیگر کاربردهای شناسائی داده‌های پرت، استفاده در سیستم‌های کشف نفوذ و کشف تقلب است [۱]. این سیستم‌ها و امثال آن‌ها که امروزه جهت کشف ناهنجاری در یک مجموعه داده با هر ابعادی به کار می‌روند، به طور کلی یک رفتار نابهنجار و غیر نرمال را با توجه به شناختی که نسبت به کلیت داده‌های نرمال دارند شناسائی می‌کنند. در واقع در این گونه سیستم‌ها، تشخیص ناهنجاری، معادل با تشخیص داده‌ی پرت است و نکته‌ی قابل توجه، آن است که بیشتر این ناهنجاری‌ها به صورت محلی می‌باشند نه به صورت سراسری و برای آن که قادر به شناسائی این گونه ناهنجاری‌ها باشند نیازمند روش‌های تشخیص داده‌های پرت محلی‌اند. از سوی دیگر امروزه داده‌های ورودی این سیستم‌ها از حجم بسیار بالائی برخوردار بوده و بنابراین نیازمند به رویکردهای برخورد با کلان‌داده‌ها می‌باشند. به عنوان مثال، می‌توان داده‌های مربوط به تراکنش‌های مالی و تجاری بانک‌ها را در نظر گرفت که طبیعتاً از حجم بسیار بالائی برخوردار هستند و کشف رفتارهای غیر نرمال در این طیف داده‌ها که از جمله‌ی آن‌ها سرقت و سوء استفاده مالی می‌باشد، بسیار حیاتی بوده و از اهمیت بسیار بالایی برخوردار می‌باشد.

## ۳.۱ چالش‌های موجود

با توجه به این‌که روش‌های سنتی تشخیص داده‌های پرت، برای کلان‌داده‌ها به خاطر حجم انبوه داده‌ها قابل استفاده نیست، ارائه‌ی روش‌های متناسب با کلان‌داده‌ها برای تشخیص داده‌های پرت از اهمیت ویژه‌ای برخوردار است. در مورد روش‌های مبتنی بر چگالی نیز باید گفت که اگر چه از دقت بالائی برخوردار می‌باشند، اما به خاطر پیچیدگی محاسباتی فزاینده‌ای که دارند برای کلان‌داده‌ها به شکل فعلی قابل استفاده نخواهند بود. در نهایت در این پایان نامه قصد داریم تا بر روی روش‌های مبتنی بر چگالی جهت کشف ناهنجاری‌ها و رفتارهای غیر نرمال در داده‌های با مقیاس بزرگ متمرکز شویم.

## ۴.۱ فرضیات

از آن‌جا که حجم داده‌هایی که ما با آن‌ها روبرو هستیم بسیار بالاست، به حدی که در آن واحد، قابلیت ذخیره‌سازی در حافظه‌ی اصلی یا «RAM»<sup>۱۶</sup> را ندارند، مجاب خواهیم بود تا در هر بازه‌ی زمانی تنها یک «قطعه»<sup>۱۷</sup> از داده‌ها را در RAM جای داده و پس از پردازش کامل آن، نتایج حاصله را به صورت خلاصه‌شده ذخیره نمائیم، به گونه‌ای که قابلیت ترکیب با نتایج پردازش سایر بخش‌ها را دارا باشد و در نهایت بتوانیم از ترکیب کلیه‌ی این نتایج جزئی، یک استنتاج جامع و قبل قبول داشته باشیم.

دو نکته‌ی مورد توجه در مورد داده‌های پرت، قابل بیان است. اول این‌که نرخ رخداد آن‌ها در قیاس با نرخ رخداد داده‌های نرمال، بسیار کمتر می‌باشد و در نتیجه باید نسبت به وقوع آن‌ها هشیار بود. دوم آن‌که داده‌های پرت محلی، نسبت به کل داده‌ها پرت نبوده و به عبارتی رفتار ناهمنجر و غیر نرمال آن‌ها در قیاس با کل داده‌ها به سادگی قابل تشخیص نمی‌باشد. به عبارتی، این نوع داده‌ها تنها نسبت به یک همسایگی خاص رفتار ناهمنجر داشته و در نتیجه شناسائی و رفع آن‌ها متضمن دقت بیشتری می‌باشد [۵].

در این پایان‌نامه، فرض ما بر آن است که تعداد داده‌ها و تعداد ابعاد کلان‌داده‌ی مورد بررسی، ثابت بوده و به عبارتی از نوع جریان داده نمی‌باشد. روش پیشنهادی ما برای کشف داده‌های پرت، مبتنی بر خوشبندی می‌باشد و در تمام رویه‌ی خوشبندی، تلاش ما آن است تا داده‌های پرت در شکل‌گیری و به روزرسانی اطلاعات خوش‌ها نقشی ایفا نکنند. اما از آن‌جا که حجم مجموعه‌داده‌ی کلان مورد بررسی به حدی زیاد است که قابلیت جاده‌ی و پردازش در حافظه‌ی اصلی را در آن واحد دارا نمی‌باشد، لذا مجبور خواهیم تا در هر لحظه، تنها یک قطعه از آن را در حافظه قرار داده و پردازش کنیم. اندازه‌ی هر قطعه هم همان‌طور که پیش از این نیز اشاره شد، به گونه‌ای است که به طور همزمان، هم قابلیت قرارگرفتن و هم پردازش‌شدن در حافظه را دارد. پس از پردازش هر قطعه از داده‌ها، می‌بایست نتایج تقریبی حاصل از پردازش این قطعه را با نتایج حاصل از پردازش قطعه‌های پیشین ترکیب و تجمیع نمائیم، به گونه‌ای که پس از پایان پردازش همگی قطعه‌ها، نتیجه‌ی کلی حاصل شده بدین صورت

<sup>16</sup> Random Access Memory (RAM)

<sup>17</sup> Chunk

تدریجی با نتیجه‌ی حاصل از پردازش یکباره، برابری نماید. الگوریتمی را که بتواند به این صورت تدریجی با مجموعه‌داده رفتار کرده و یک نتیجه‌ی تقریبی را به دست آورد، از دیدگاه عملیاتی، یک الگوریتم مقیاس‌پذیر گویند [۶]. از دیدگاه الگوریتمی نیز مقیاس‌پذیربودن به معنی آن است که پیچیدگی مسئله با توجه به اندازه‌ی ورودی آن باید به اصطلاح «به طور تقریبی خطی»<sup>۱۸</sup> و یا «زیر خط»<sup>۱۹</sup> باشد [۷].

## ۵.۱ هدف از اجراء

روش‌های مبتنی بر چگالی، برای تشخیص داده‌های پرت از دقت بالایی در تشخیص این نوع داده‌ها برخوردار هستند ولی با توجه به سربار محاسباتی آن‌ها، به کارگیری آن‌ها برای کلان‌داده‌ها با چالش جدی روبروست. در این پایان‌نامه، قصد ما بر آن است تا روشی را جهت کشف داده‌های پرت محلی در کلان‌داده‌ها ارائه نمائیم که مبتنی بر یک «چارچوب خوشبندی مقیاس‌پذیر»<sup>۲۰</sup> می‌باشد و از روش‌های خوشبندی مبتنی بر چگالی، جهت کشف نواحی متراکم در هر مرحله استفاده می‌کند.

## ۶.۱ نوآوری‌ها

روش پیشنهادی ما، در واقع حالت توسعه‌یافته‌ای از الگوریتم BFR [۸] می‌باشد. اما مسئله آن است که الگوریتم BFR، مجموعه‌داده‌ی ورودی را عاری هر گونه نویز و داده‌ی پرت فرض کرده و هدف غائی آن، فقط خوشبندی می‌باشد. بدین ترتیب اگر مجموعه‌داده‌ی ورودی، دارای داده‌ی پرت از هر دو نوع ضعیف یا قوی باشد، این داده‌های نامطلوب نیز در نهایت به یک خوش نسبت داده شده و در شکل‌گیری ساختار آن خوش، نقش مؤثری ایفا می‌نمایند. علاوه بر این، فرض قوی اولیه‌ی الگوریتم BFR بر نرمال‌بودن توزیع خوش‌ها و نیز ناهمبسته‌بودن مقادیر ویژگی‌های هر یک از آن‌ها می‌باشد.

<sup>18</sup> Nearly linear

<sup>19</sup> Sublinear

<sup>20</sup> Scalable Clustering Framework

روش پیشنهادی ما، از سه مرحله‌ی اصلی تشکیل شده است. در مرحله‌ی اول، یک نمونه‌برداری اولیه از مجموعه‌داده انجام گرفته و از آن اطلاعات اولیه‌ی خوش‌ها و البته پارامترهای لازم جهت ادامه‌ی خوش‌بندی اخذ می‌گردد. در مرحله‌ی دوم، خوش‌بندی مقیاس‌پذیر انجام می‌شود. تلاش ما در طول این مرحله آن است تا داده‌های پرت در شکل‌گیری خوش‌ها نقشی نداشته باشند و در نتیجه در پایان این مرحله، تعداد اندکی از داده‌ها در حافظه‌ی اصلی یا همان RAM بلا تکلیف خواهند ماند. برخی از این داده‌ها، داده‌ی پرت می‌باشند که به درستی در حافظه به حالت معلق باقی مانده‌اند و برخی نیز داده‌های نرم‌ال می‌باشند که به دلیل محدودیت‌های اعمال‌شده، نتوانسته‌اند در تشکیل خوش‌ها نقش مؤثری ایفا نمایند. در نهایت، همگی این داده‌ها از حافظه‌ی اصلی پاک شده و اطلاعات ساختاری «خوش‌های موقت»<sup>۲۱</sup> در حافظه باقی می‌مانند. در این قسمت، باید با اعمال پارامترهای مناسب، خوش‌های موقت را با یکدیگر ترکیب کرده و ساختار «خوش‌های نهائی»<sup>۲۲</sup> را به دست آوریم. در مرحله‌ی سوم، یک بار دیگر تمامی داده‌های مجموعه‌داده را با توجه به خوش‌های نهائی حاصله از مرحله‌ی دوم، پردازش کرده و با توجه به یک معیار فاصله‌ی مناسب، به آن‌ها امتیازی مبنی بر میزان پرت‌بودن خواهیم داد. عمدتی نوآوری‌های ما در این الگوریتم پیشنهادی، به قرار زیر می‌باشد:

- برخلاف الگوریتم BFR، در ابتدای امر، نیازی به دانستن تعداد خوش‌های اصلی یا همان  $K$  نمی‌باشد.
- علاوه بر خوش‌های گاؤسین با مقادیر ویژگی ناهمبسته، قادر به شناسائی خوش‌های نرم‌ال با مقادیر ویژگی همبسته نیز می‌باشد.
- پیچیدگی زمانی آن، خطی و از مرتبه‌ی  $O(n)$  می‌باشد.
- در حالی که در یک رویه‌ی مقیاس‌پذیر و به صورت قطعه‌قطعه، اقدام به خوش‌بندی مبتنی بر چگالی و کشف داده‌های پرت می‌نماید و نیازی به دانستن اطلاعات کل داده‌ها در آن واحد و در ابتدای امر ندارد، اما دقیقاً آن با سایر روش‌هایی که دادگان را به یکباره دیده و در مورد آن تصمیم‌گیری می‌نمایند، رقابت می‌کند.

<sup>21</sup> Temporary clusters

<sup>22</sup> Final Clusters

## ۷.۱ ساختار پایان نامه

این پایان نامه در ادامه به این صورت تدوین شده است: در فصل دوم، به بررسی دقیق‌تر انواع داده‌های پرتو پرداخته و عمدۀ روش‌های کشف داده‌های پرتو معرفی می‌گردد. همین‌طور به برخی از کاربردهای کشف داده‌های پرتو نیز اشاره شده و چالش‌های موجود در این زمینه را به اختصار بیان خواهیم نمود. در فصل سوم، به بررسی مفصل برخی از روش‌های ارائه شده تاکنون جهت کشف داده‌های پرتو خواهیم پرداخت. در فصل چهارم، مدل پیشنهادی جهت کشف داده‌های پرتو را ارائه خواهیم نمود. در فصل پنجم، نتایج آزمایشات انجام‌شده بر روی مجموعه داده‌های واقعی و مصنوعی را به همراه تحلیل آن‌ها عرضه کرده‌ایم. فصل ششم نیز پایان نامه را با جمع‌بندی و نتیجه‌گیری پایان می‌دهد.

۲

## فصل دوم

### مفاهیم پایه

در این فصل، در ابتدا تعریفی از داده‌ی پرت ارائه شده و عمدی انواع آن معرفی می‌گردد. در ادامه، به تفاوت میان داده‌های پرت و «داده‌های نویزی»<sup>۲۳</sup> پرداخته و تعدادی از برجسته‌ترین مدل‌های موجود جهت کشف داده‌های پرت را به اختصار معرفی می‌نماییم. سپس از اهمیت و ضرورت کشف داده‌های پرت در کاربردهای گوناگون صحبت نموده و برخی از چالش‌های موجود در این زمینه را برخواهیم شمرد.

## ۱.۲ داده‌ی پرت چیست؟

داده‌های پرت، در واقع «الگوهای»<sup>۲۴</sup> خاصی در مجموعه‌داده هستند که با یک مفهوم مشخص از رفتار نرمال در آن مجموعه‌ی داده‌ی خاص هم خوانی ندارند [۱]. داده‌های پرت را می‌توان به لحاظ مفهومی مکمل خوش‌های نرمال موجود در مجموعه‌داده دانست. چرا که در خوش‌بندی به دنبال داده‌هایی هستیم که با یکدیگر شباهت بالاتری دارند، و در این میان داده‌های پرت آن دسته از «داده‌های منفرد»<sup>۲۵</sup> می‌باشند که نسبت به مابقی داده‌ها تفاوت قابل ملاحظه‌ای دارند [۴]. شکل ۱.۲ آرایشی از داده‌های پرت را در کنار «داده‌های نرمال»<sup>۲۶</sup> در یک فضای ساده‌ی دوبعدی نمایش می‌دهد. این مجموعه‌داده دارای دو محدوده‌ی نرمال  $N_1$  و  $N_2$  می‌باشد، و همان‌طور که پیداست، بیشتر داده‌ها نیز در این دو محدوده‌ی مشخص واقع شده‌اند. تجمع عمدی داده‌ها در این دو محدوده، خود گواهی بر نرمال‌بودن آن‌ها می‌باشد، چرا که معمولاً تعداد داده‌های پرت نسبت به سایر داده‌ها بسیار کمتر می‌باشد و در نتیجه در محدوده‌های چگال‌تر یافت نمی‌شوند. اما داده‌هایی که به طرز قابل توجهی خارج از نواحی با چگالی بالا قرار دارند، مانند نقاط  $o_1$  و  $o_2$  و البته نقاطی که در ناحیه‌ی نسبتاً چگال  $o_3$  قرار دارند، پرت به حساب می‌آیند [۱].

لازم به ذکر است که نقاطی که در ناحیه‌ی  $o_3$  قرار دارند، به دلیل چگالی اندکی که در این ناحیه وجود دارد، می‌توانند چالشی برای آن دسته از روش‌های کشف داده‌های پرت باشند که از «رویکردهای مبتنی

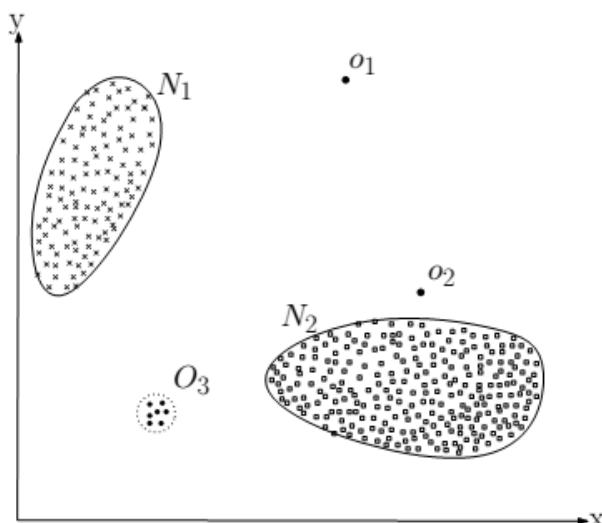
<sup>23</sup> Noisy data (Noise)

<sup>24</sup> Patterns

<sup>25</sup> Individual data points

<sup>26</sup> Inliers

بر چگالی»<sup>۲۷</sup> استفاده می‌نمایند. البته چنین خوش‌های کوچکی از داده‌های پرت معمولاً زمانی ایجاد می‌شوند که فرایند مربوط به تولید آن‌ها به تعداد محدودی تکرار گردد و این یک رویه‌ی کاملاً طبیعی در مورد داده‌های پرت به حساب آمده و خود جالب توجه و حائز اهمیت می‌باشد [۴].



شکل ۱.۲ یک مثال ساده از داده‌های پرت در یک مجموعه‌داده‌ی دوبعدی [۱]

تعریف دیگری از داده‌های پرت توسط «هاوکینز»<sup>۲۸</sup> بدین صورت می‌باشد:

«یک داده‌ی پرت، داده‌ای است که مقادیر ضبط شده برای ویژگی‌های آن به حدی نسبت به سایر داده‌ها متفاوت است که موجب بروز شک و تردید برای تحلیل‌گر می‌شود، به این مضمون که این داده می‌بایست توسط مکانیزم متفاوت از مکانیزم تولید داده‌های نرمال به وجود آمده باشد [۹].»

داده‌های پرت را در ادبیات داده‌کاوی و آماری تحت عناوین مختلفی از جمله «داده‌های غیرنرمال»<sup>۲۹</sup>، «داده‌های ناسازگار»<sup>۳۰</sup>، «داده‌های منحرف»<sup>۳۱</sup> و «داده‌های نابهنجار»<sup>۳۲</sup> نیز معرفی می‌نمایند. در بسیاری از مسائل مربوط به داده‌کاوی، داده‌هایی که تولید می‌شوند توسط یک یا تعداد بیشتری از فرآیندهای

<sup>27</sup> Density-based approaches

<sup>28</sup> Hawkins

<sup>29</sup> Abnormalities

<sup>30</sup> Discordants

<sup>31</sup> Deviants

<sup>32</sup> Anomalies

غیرملموس تولیدکننده‌ی داده به وجود می‌آیند. حال اگر این فرآیند تولیدکننده‌ی داده دچار اختلال شده و در یک رویه‌ی غیرمعمول عمل نماید، موجب تولید داده‌های نابهنجار یا همان داده‌های پرت خواهد شد. با توجه به این، یک داده‌ی پرت می‌تواند از این دیدگاه جذاب باشد که حاوی اطلاعات مفیدی در مورد خصیصه‌های غیرنرمال سیستم تولید داده و موجودیت‌ها یا همان ویژگی‌های ضبطشده است [۱۰].

## ۲.۲ پیدایش داده‌های پرت

می‌توان دلیل پیدایش داده‌های پرت را از دیدگاه آماری این‌گونه بیان نمود که در واقع دو مکانیزم بنیادی وجود دارند که می‌توانند سبب ایجاد داده‌های پرت گردند. تشخیص این که کدام یک از این مکانیزم‌ها سبب ایجاد داده‌های پرت گشته است بسیار مهم می‌باشد، چرا که بر روی برداشت و استنتاج «تحلیل گر داده»<sup>۳۳</sup> تأثیر جدی خواهد داشت.

- **مکانیزم اول:** مجموعه‌داده‌ی مربوطه توسط یک توزیع با «دباله‌ی سنگین»<sup>۳۴</sup> مانند «توزیع تی»<sup>۳۵</sup> به وجود آمده است. بدین سبب هر داده‌ای می‌تواند پرت و غیرنرمال به نظر برسد، مگر آن‌که یک «حد آستانه‌ی»<sup>۳۶</sup> مشخص جهت تعیین پرت‌بودن معین گردد.

با توجه به مدل اول تولید داده‌های پرت، می‌توان خانواده‌ی توزیع‌های آماری را به دو دسته‌ی اصلی تقسیم نمود. یک دسته توزیع‌هایی هستند که متمایل به دارابودن داده‌های پرت هستند و به اختصار آن‌ها را «متتمایل به پرت»<sup>۳۷</sup> می‌نامیم؛ و دسته‌ی دیگر توزیع‌هایی هستند که در برابر حضور داده‌های پرت از خود مقاومت نشان می‌دهند که به اختصار آن‌ها را «مقاوم در برابر پرت»<sup>۳۸</sup> نام می‌نیم. خانواده‌ی توزیع‌های متمایل به پرت، دارای دنباله‌ای طولانی هستند که به آرامی به سمت صفر میل می‌کند و خود

<sup>33</sup> Data analyst

<sup>34</sup> Heavy-tailed distribution

<sup>35</sup> “Student’s t-distribution” OR simply the “t-distribution”

<sup>36</sup> Threshold

<sup>37</sup> Outlier-prone

<sup>38</sup> Outlier-resistant

به دو زیردسته‌ی مطلقاً متمایل به پرت و نسبتاً متمایل به پرت تقسیم می‌شوند. خانواده‌ی توزیع‌های مقاوم در برابر پرت نیز به آن دسته از توزیع‌هایی اطلاق می‌گردد که متمایل به پرت نباشند.

- **مکانیزم دوم:** داده‌های موجود برخاسته از دو توزیع متفاوت می‌باشند. یکی از این توزیع‌ها که «توزیع بنیادی»<sup>۳۹</sup> نامیده می‌شود، داده‌های به اصطلاح «خوب»<sup>۴۰</sup> را تولید می‌نماید، در حالی که توزیع دوم که «توزیع آلوده‌کننده»<sup>۴۱</sup> نامیده می‌شود، داده‌های به اصطلاح «آلوده»<sup>۴۲</sup> را تولید می‌نماید. اگر توزیع آلوده‌کننده دنباله‌های سنگین‌تری نسبت به توزیع بنیادی داشته باشد، در آن صورت گرایش بیشتری به این مطلب وجود خواهد داشت که داده‌های آلوده را همان داده‌های پرت تصور نماییم، که البته با این کار به صورت کاملاً آشکارا داده‌های حاصل از توزیع بنیادی را داده‌های نرمال و تماماً منفک از داده‌های حاصل از توزیع دوم تصور نموده‌ایم.

اگر مکانیزم دوم در مورد یک مجموعه‌داده صحت داشته باشد، باز هم مانند آن‌چه در مورد مکانیزم اول بیان شد، می‌توان به دو صورت مطلق و نسبی، داده‌های حاصل از دو توزیع را از یکدیگر سوا نمود. حالت اول آن است که به ضرس قاطع بگوئیم در یک مجموعه‌داده با  $n$  داده، تعداد  $k$  داده برخاسته از توزیع بنیادی و به عبارتی نرمال می‌باشند، و به دنبال آن تعداد  $k$  داده نیز حاصل از توزیع آلوده‌کننده و به عبارت دیگر پرت می‌باشند. حالت دوم که از قطعیت به دور است نیز آن است که به صورت نسبی و احتمالاتی بیان کنیم که هر داده‌ی موجود، با احتمال  $p$  برخاسته از توزیع آلوده‌کننده و با احتمال  $1-p$  حاصل از توزیع بنیادی می‌باشد. در اینجا مقدار پارامتر  $k$  یک متغیر تصادفی می‌باشد که خود از «توزیع دوجمله‌ای»<sup>۴۳</sup> پیروی می‌نماید. در ضمن در بسیاری از موارد مقدار احتمالاتی  $p$  یک مقدار نزدیک به صفر در نظر گرفته می‌شود، حتی در مورد مسائلی که مقدار دقیق آن مشخص نمی‌باشد [۹].

<sup>39</sup> Basic distribution

<sup>40</sup> Good observations

<sup>41</sup> Contaminating distribution

<sup>42</sup> Contaminants

<sup>43</sup> Binomial distribution

## ۳.۲ انواع داده‌های پرت

به طور کلی می‌توان داده‌های پرت را در سه دسته‌ی «سراسری»<sup>۴۴</sup>، «حیطه‌ای (وابسته به قرائن)» یا «شرطی»<sup>۴۵</sup> و «تجمعی»<sup>۴۶</sup> ردیبندی نمود، که در ادامه به شرح مختصری از هر کدام بسنده خواهیم نمود.

### ۱.۳.۲ داده‌های پرت سراسری

داده‌های پرت سراسری، به آن دسته از داده‌ها در مجموعه‌داده اطلاق می‌گردد که نسبت به همگی داده‌های دیگر موجود رفتار غیر نرمالی را از خود بروز می‌دهند و به همین سبب در برخی موارد به آن‌ها «ناهنجری‌های برجسته»<sup>۴۷</sup> نیز اطلاق می‌شود. کشف این‌گونه داده‌های پرت بسیار ساده می‌باشد و البته بسیاری از روش‌های موجود هم مربوط به یافتن همین نوع ناهنجاری‌های برجسته می‌باشند، چرا که با ساده‌ترین ایده‌ها نیز می‌توان با دقت بالائی آن‌ها را کشف نمود. به عنوان مثال اگر به همان شکل ۱.۲ توجه نمائیم، مشاهده می‌کنیم که داده‌ی **۰۱** نسبت به سایر داده‌های موجود رفتار متفاوت و غیر نرمالی از خود بروز داده و البته از توزیع سایر داده‌ها نیز پیروی نمی‌نماید و به همین دلیل به عنوان داده‌ی پرت سراسری شناخته می‌شود.

نکته‌ی مهم در مورد داده‌های پرت سراسری این است که باید برای کشف آن‌ها ضابطه و معیار معین و صحیحی از انحراف را تعریف نمود و البته که این معیار می‌باشد با توجه به نوع کاربرد و مجموعه‌داده‌ای که مورد بررسی است مشخص شود. با توجه به معیارهای متنوعی که برای کشف داده‌های پرت سراسری تعریف می‌شود، این روش‌ها را در دسته‌های متفاوتی قرار داده و با توجه به نیاز مورد استفاده قرار می‌دهند.

به عنوان مثال در زمینه‌ی کشف نفوذ در شبکه‌های کامپیوتری، اگر رفتار ارتباطی یک سیستم خاص به طرز قابل توجهی نسبت به الگوی رفتاری نرمال و معمول شبکه متفاوت باشد (مثلا حجم زیادی از

<sup>44</sup> Global

<sup>45</sup> Contextual (or Conditional)

<sup>46</sup> Collective

<sup>47</sup> Point anomalies

اطلاعات را در عرض مدت کوتاهی به بیرون مخابره نماید)، این طرز رفتار می‌تواند به عنوان یک داده‌ی پرت سراسری در نظر گرفته شود و سیستم کامپیووتری مربوطه نیز به عنوان یک مورد مشکوک به هکشدن شناخته شده و مورد بررسی واقع خواهد شد [۲].

## ۲.۳.۲ داده‌های پرت حیطه‌ای (وابسته به قرائن) یا شرطی

در یک مجموعه‌داده، یک داده‌ی خاص در صورتی داده‌ی پرت حیطه‌ای و یا وابسته به قرائن به حساب می‌آید که رفتار غیر نرمالی را با توجه به ویژگی‌های خاصی از داده نسبت به اطرافیانش از خود بروز دهد. به عبارتی به ازای هر زیرمجموعه‌ی خاصی از ویژگی‌های یک داده، می‌توان برای آن مجموعه‌ی همسایگی متفاوت و به بیان ادبی مجموعه‌ی قرائن و همنشینان متفاوتی را تعریف نمود و سپس در این حیطه بررسی نمائیم که آیا داده‌ی مربوطه نسبت به اطرافیانش از خود رفتار نابهنجاری بروز می‌دهد یا خیر. داده‌های پرت حیطه‌ای را گاهی داده‌های پرت شرطی نیز می‌نامند، چرا که پرت‌بودن آن‌ها مشروط به نوع همسایگی آن‌ها می‌باشد. بدین ترتیب در مورد داده‌های پرت حیطه‌ای می‌بایست حیطه و قرائن مربوطه به درستی مشخص گردد تا در محاسبات دچار خطأ نشویم [۱۰, ۲].

به طور عمومی در مورد کشف این گونه داده‌های پرت، لازم است تا ویژگی‌های مجموعه‌داده مورد بررسی را به دو گروه زیر تقسیم نمائیم:

- **ویژگی‌های قرائني:** آن دسته از ویژگی‌های یک داده که به سبب آن‌ها می‌توان همسایگی مشروط آن داده را تعیین نمود، ویژگی‌های قرائني نامند. به عنوان مثال اگر بگوئیم که «دمای فعلی ۲۱ درجه‌ی سانتی‌گراد است و حال آیا / این وضعیت یک وضعیت استثنائی است یا نه؟!»، می‌بایست بررسی نمود که مکان و فصلی از سال که این جمله در آن قید شده چه فصلی بوده است. اگر مکان مربوطه شهر تورنتو و فصل مربوطه نیز زمستان بوده پس آری! این وضعیت یک داده‌ی پرت به حساب می‌آید؛ اما اگر مکان شهر تورنتو و فصل نیز تابستان بوده است نه! این یک داده‌ی نرمال محسوب می‌شود. با این تفاسیر ویژگی‌های قرائني در این مثال مکان و تاریخ مربوط به دمای قیدشده می‌باشد که می‌توان به سبب آن‌ها برای این داده یک همسایگی مناسب تعریف نموده و میزان پرت‌بودن آن را بررسی نمود.

- **ویژگی‌های رفتاري:** ویژگی‌های رفتاري به آن دسته از ویژگی‌های یک داده اطلاق می‌گردد که با استفاده از آن‌ها بررسی می‌گردد که آیا داده‌ی مربوطه در حیطه‌ای که برای آن با استفاده از ویژگی‌های قرائني تعریف شده است، یک داده‌ی پرت به حساب می‌آید یا نه. مثلاً در مثالی که پیش از در مورد ویژگی‌های قرائني قید شده، ویژگی‌های رفتاري را می‌توان دمای فعلی، میزان رطوبت و فشار هوا درنظر گرفت.

همان طور که مشاهده می‌شود برخلاف داده‌های پرت سراسری که تنها ویژگی‌های رفتاری یک داده را نسبت به تمامی مابقی مجموعه داده مورد بررسی قرار می‌دادیم، در مورد داده‌های پرت حیطه‌ای می‌باشد هر دوی ویژگی‌های رفتاری و البته ویژگی‌های قرائی را در نظر بگیریم. یک زیرمجموعه از مقادیر ویژگی‌های رفتاری می‌توانند با توجه یک حیطه‌ی خاص سبب شوند تا داده‌ی مربوطه به عنوان داده‌ی پرت شناسائی شود (به عنوان مثال دمای ۲۸ درجه در زمستان شهر تورنتو)، در حالی که در حیطه‌ای دیگر سبب می‌شوند تا داده‌ی مربوطه نرمال شناسائی گردد (به عنوان مثال دمای ۲۸ درجه در تابستان شهر تورنتو).

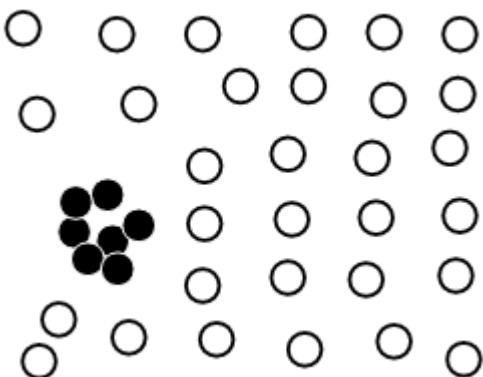
در مورد داده‌های پرت سراسری نیز می‌توان گفت که در واقع شکل خاصی از داده‌های پرت حیطه‌ای می‌باشد که در آن زیرمجموعه‌ی ویژگی‌های قرائی تهی می‌باشد و در نتیجه محدودیتی برای انتخاب همسایگی وجود ندارد. به بیان دیگر رویه‌ی کشف داده‌های پرت سراسری تمامی داده‌های مجموعه داده را به عنوان یک حیطه‌ی مورد بررسی برای سنجش میزان پرت‌بودن هر داده در نظر می‌گیرد. از سوی دیگر، رویه‌ی کشف داده‌های پرت حیطه‌ای یک نوع انعطاف‌پذیری را برای کاربر ایجاد می‌نماید تا به دلخواه خود پرت‌بودن هر داده را در یک حیطه‌ی خاص مورد بررسی قرار دهد، که در زمینه‌های متعددی کاربرد فراوان دارد.

در اینجا باید قيد گردد که داده‌های پرت حیطه‌ای یا شرطی، شکل تکامل‌یافته‌تری از همان داده‌های پرت محلی می‌باشند. محلی‌بودن مفهومی است که با توجه به رویکردهای کشف داده‌ی پرت مبتنی بر چگالی بهتر می‌توان آن را درک نمود. با توجه به این رویکردها، به ازای هر داده یک همسایگی مشخص تعریف شده و در آن همسایگی برای داده‌ی مربوطه یک مقدار چگالی محلی (مثلاً با توجه به فاصله‌ی آن از نزدیک‌ترین همسایگانش) به دست می‌آید و در نهایت بر مبنای چگالی‌های حاصله، یک ضریب داده‌ی پرت محلی به معنای میزان پرت محلی‌بودن برای هر داده تعریف می‌شود. به عبارتی اگر چگالی محلی یک داده در حیطه‌ی مورد بررسی نسبت به چگالی محلی همسایگان آن داده، به میزان قابل توجهی کمتر باشد، آن‌گاه داده‌ی مربوطه امتیاز بسیار بیشتری را از منظر پرت محلی‌بودن کسب خواهد نمود [۲].

### ۳.۳.۲ داده‌های پرت تجمعی

در یک مجموعه‌داده‌ی خاص، اگر زیرمجموعه‌ای از داده‌ها به صورت تجمعی رفتار غیر معمولی نسبت به سایر داده‌های مجموعه از خود نشان دهنده، به گونه‌ای که هر کدام از آن داده‌ها به صورت انفرادی چنین رفتاری را از خود بروز نمی‌دهند، آن‌گاه زیرمجموعه‌ی مربوطه را یک داده‌ی پرت تجمعی نامند. به عنوان مثال در شکل ۲.۲ مشاهده می‌کنیم که داده‌های مشخص شده با رنگ مشکی با توجه به این که چگالی و تراکم آن‌ها نسبت به سایر داده‌های مجموعه شدیداً بیشتر می‌باشد، لذا جمعاً یک داده‌ی پرت تجمعی به حساب می‌آیند. ولی همان‌طور که پیداست هر کدام از این داده‌ها به تنها نسبت به مابقی داده‌ها، پرت به حساب نمی‌آید [۲].

از جمله کاربردهای کشف داده‌های پرت تجمعی در زمینه‌ی کشف خرابکاری در بازار بورس می‌باشد. به این ترتیب که اگر بین دو گروه از بازار بورس، به تنها نی یک تراکنش مربوط به یک سهام خاص اتفاق بیفتند نرمال درنظر گرفته خواهد شد، اما اگر از همین تراکنش مربوط به همان سهام خاص به تعداد زیاد و در عرض مدت زمان کوتاهی اتفاق بیفتند، به عنوان یک داده‌ی پرت تجمعی در نظر گرفته خواهد شد، چرا که می‌تواند دلیل بر آن باشد که تعدادی از افراد عضو در حال تغییر و دستکاری بازار بورس بوده و به عبارتی به دنبال خرابکاری و یا نوعی بهره‌وری غیر قانونی می‌باشند [۱۰، ۲].



شکل ۲.۲ داده‌های مشخص شده با رنگ مشکی جمعاً یک داده‌ی پرت تجمعی را تشکیل می‌دهند [۲]

مثال دیگر در مورد داده‌های پرت تجمعی، رویه‌ی مدیریت زنجیره‌ی تأمین یک شرکت به خصوص می‌باشد. اگر سفارشات موجود و رویه‌ی انتقال محموله‌ها را در نظر بگیریم، رخداد تأخیر به ازای یک محموله‌ی تصادفی را می‌توان نادیده گرفت، چرا که تأخیر در انتقال کالاهای تجاری هر از چندگاهی

اتفاق می‌افتد. اما اگر تعداد مثلاً ۱۰۰ سفارش تنها در یک روز دچار تأخیر در انتقال گردند، در آن صورت باید به این مسئله توجه خاص شده و همگی آن ۱۰۰ سفارش تأخیری را به صورت جمعی به عنوان تنها یک داده‌ی پرت به حساب آورد. در حالی که هر کدام از این سفارشات ممکن است به تنهاei به عنوان یک داده‌ی پرت محسوب نگردد. به عبارت دیگر می‌بایست به همگی این ۱۰۰ سفارش جمعاً با یک نگاه نزدیک‌تر و دقیق‌تر نگریسته شود تا مشکل موجود در انتقال محموله‌ها مشخص و مرتفع گردد [۲].

برخلاف روش‌های کشف داده‌های پرت سراسری و البته حیطه‌ای، در مورد کشف داده‌های پرت تجمعی می‌بایست نه تنها ویژگی‌های رفتاری تک‌تک داده‌ها بلکه ویژگی‌های رفتاری گروه‌های مختلفی از داده‌ها را پیوسته مورد بررسی قرار دهیم، که این مهم مستلزم داشتن آگاهی پیشین نسبت به ارتباطات میان داده‌ها اعم از فاصله و میزان شباهت میان آن‌ها می‌باشد [۲], [۱۰].

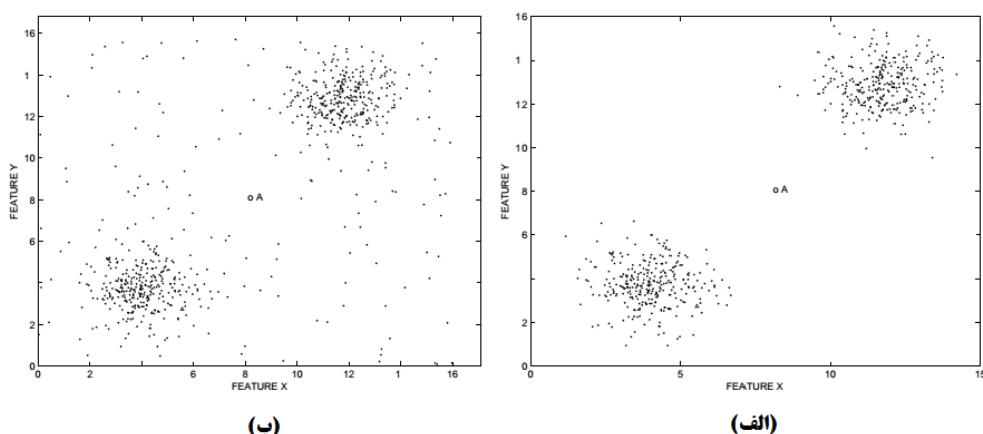
در پایان معرفی انواع داده‌های پرت باید گفت که یک مجموعه داده می‌تواند شامل انواع داده‌های پرت باشد و یا حتی یک داده‌ی خاص می‌تواند به انواع مختلفی از داده‌های پرت نسبت داده شود.

## ۴.۲ تفاوت داده‌های پرت با داده‌های نویزی

لازم به ذکر است که داده‌ی پرت با داده‌ی نویزی به طور جدی متفاوت می‌باشد. چرا که نویز یک خطای تصادفی و یا هم یک نوع «واریانس»<sup>۴۸</sup> در مقادیر ویژگی‌ها یا همان متغیرهای اندازه‌گیری شده به حساب می‌آید و به همین سبب کشف داده‌های نویزی در علم داده‌کاوی از اهمیت کمتری نسبت به کشف داده‌های پرت برخوردار است [۲]. نویز را می‌توان به عنوان یک پدیده‌ی ناخواسته در مجموعه‌داده در نظر گرفت که خارج از توجه و علاقه‌ی تحلیل‌گر داده می‌باشد، اما می‌تواند در قالب یک مانع بر سر راه تحلیل و تجزیه‌ی داده‌ها عمل نماید [۱]. به عنوان مثال در مورد مثال کشف تقلب در مورد کارت‌های اعتباری، رویه‌ی خرید یک مشتری را می‌توان به صورت یک متغیر تصادفی مدل‌سازی نمود. به این ترتیب که یک مشتری در حالت معمول اقلام خاصی را با استفاده از کارت اعتباری خویش خریداری می‌نماید و حال اگر در بعضی موارد مثلاً وعده‌ی غذائی گران‌تری را خریداری نموده و یا هم از یک نوع نوشیدنی خاص، مقدار بیشتری را خارج از نرم سفارش دهد، در آن صورت در صورت حساب کارت

<sup>48</sup> Variance

اعتباری خوبیش تعدادی به اصطلاح «تراکنش نویزی»<sup>۴۹</sup> به وجود آورده است که می‌توان آن‌ها را به عنوان «خطای تصادفی»<sup>۵۰</sup> و یا هم واریانس در ویژگی‌های مدنظر مربوط به آن مدل خرید کاربر در نظر گرفت. لذا این‌گونه تراکنش‌ها را نمی‌توان به عنوان داده‌های پرت در نظر گرفت، چرا که شناسائی آن‌ها نه تنها موجب هزینه‌های سنگین برای شرکت صادرکننده‌ی کارت اعتباری خواهد بود، بلکه حتی ارسال چنین «هشدارهای غلطی»<sup>۵۱</sup> به مشتریان موجب رنجش خاطر آنان می‌شود، چرا که مشتری مربوطه آگاهانه چنین خریدهایی را گاه و بی‌گاه انجام داده و در نتیجه از دید او چنین پیغام‌هایی از سوی شرکت صادرکننده‌ی کارت، مزاحم تلقی شده و همگی این مسائل در نهایت به ضرر شرکت مربوطه ختم خواهد شد [۲]. حذف داده‌های نویزی از آن جهت حائز اهمیت است که می‌بایست داده‌های ناخواسته را قبل از رویه‌ی کشف داده‌های پرت و البته هر رقم رویه‌ی تحلیل و تجزیه‌ی داده‌ها از مجموعه‌داده حذف نمود [۱, [۲].



شکل ۳.۲ تفاوت میان داده‌های نویزی و داده‌های پرت [۱۰]

در اینجا باید گفت که تعیین این که چه میزان انحراف از نرم به اصطلاح «کافی»<sup>۵۲</sup> خواهد بود تا یک داده به عنوان یک داده‌ی پرت شناسائی گردد، بسته به «قضاؤت شخصی»<sup>۵۳</sup> داشته و در میان

<sup>49</sup> Noisy transaction

<sup>50</sup> Random error

<sup>51</sup> False alarms

<sup>52</sup> Sufficient

<sup>53</sup> Subjective judgement

مجموعه‌داده‌های متفاوت و با توجه به علاقه و تشخیص تحلیل‌گر داده متمایز خواهد بود. در موارد دنیای واقعی، یک مجموعه‌داده ممکن است شامل مقادیر زیادی داده‌ی نویزی باشد که عمدتاً خارج از علاقه و توجه تحلیل‌گر داده می‌باشد و معمولاً آن دسته از داده‌هایی مورد توجه تحلیل‌گر می‌باشند که «به طرز قابل ملاحظه‌ای انحراف جالب توجه»<sup>۵۴</sup> داشته باشند. جهت نمایش این نکته می‌توان به مثال‌های (الف) و (ب) موجود در شکل ۳.۲ توجه نمود. پر واضح است که الگوهای اصلی (همان خوشها) در هر دو مثال کاملاً یکسان می‌باشند، اما تفاوت قابل ملاحظه‌ی اصلی در نواحی بیرونی این خوشها می‌باشد. در مورد مثال (الف)، داده‌ی نام‌گذاری شده با A تفاوت قابل توجهی با مابقی داده‌ها داشته و به همین سبب به عنوان یک داده‌ی پرت یا نابهنجار شناخته می‌شود. اما همین مسئله را در مورد مثال (ب) نمی‌توان با قاطعیت بیان نموده و بیشتر به قضایت شخصی تحلیل‌گر وابسته است. چرا که داده‌ی A در مثال (ب) در یک «ناحیه‌ی تُنک»<sup>۵۵</sup> از مجموعه‌داده واقع شده است و به همین سبب نمی‌توان به ضرس قاطع بیان نمود که انحراف چشم‌گیری را به نسبت سایر داده‌ها از خود نشان می‌دهد. همین‌طور با توجه به قربت معنائی که این داده با سایر داده‌های نویزی موجود دارد، می‌توان گفت که داده‌ی A نیز یک نویز به حساب آمده و چندان مورد توجه تحلیل‌گر داده نخواهد بود.

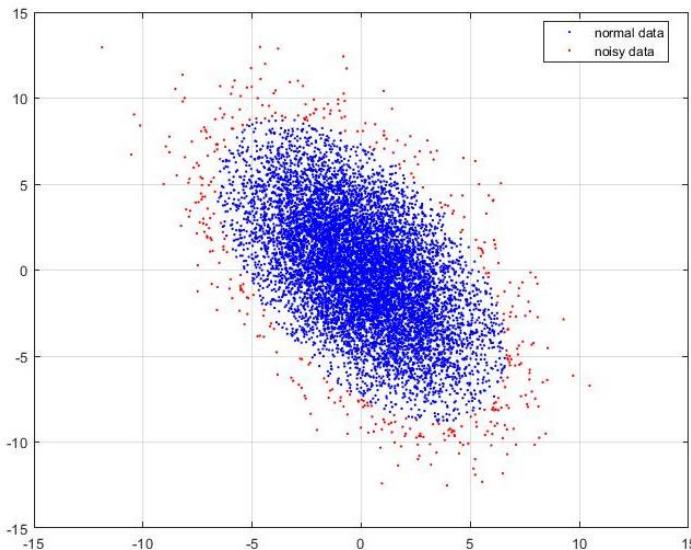
داده‌های نویزی در واقع مرز معنائی میان داده‌های نرمال و داده‌های پرت به حساب می‌آیند و به عبارتی نویز را به عنوان یک داده‌ی پرت ضعیف مدل‌سازی می‌نمایند. داده‌های نویزی حائز شرایط لازم جهت این‌که از دید تحلیل‌گر داده جالب توجه و به اندازه‌ی کافی نابهنجار به حساب آیند نمی‌باشند. به عنوان مثال می‌توان داده‌هایی را که در نواحی تُنک حاشیه‌ی یک خوشه قرار دارند به عنوان داده‌های نویزی معرفی نمود. شکل ۴.۲ یک خوشه را نشان می‌دهد که نواحی حاشیه‌ای آن با رنگ متمایزی نشان داده شده‌اند و به نوعی نمایانگر داده‌های نویزی می‌باشند. با توجه به آن‌چه در مورد قربت معنائی داده‌های نویزی با داده‌های پرت مطرح شد، اکثریت قریب به اتفاق روش‌های کشف داده‌های پرت نیز از این‌که در پایان امر یک «تصمیم سخت و قطعی»<sup>۵۶</sup> را در مورد پرت‌بودن و یا نبودن داده‌ها ارائه نمایند، اجتناب

<sup>۵۴</sup> Significantly interesting deviation

<sup>۵۵</sup> Sparse region

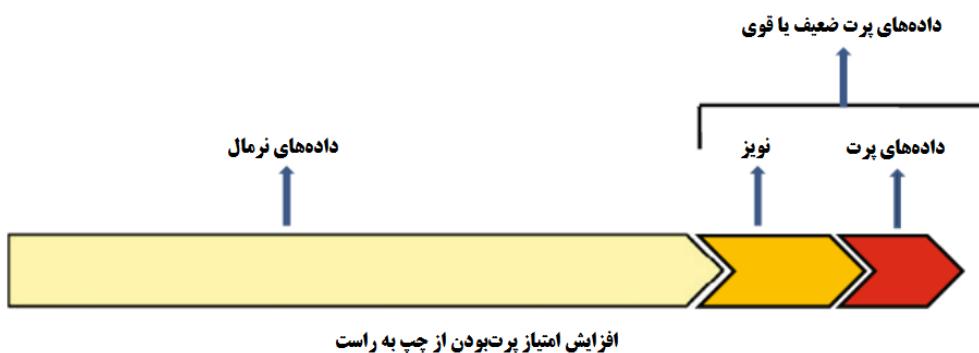
<sup>۵۶</sup> Crisp decision

کرده و در عوض به ازای هر داده، یک امتیاز با مقدار حقیقی را به عنوان میزان پرتبودن آن داده معرفی می‌نمایند.



شکل ۴.۲ تفاوت میان داده‌های نرمال و داده‌های نویزی

با توجه به شکل ۵.۲ امتیاز هر داده به لحاظ میزان پرتبودن بر روی یک طیف پیوسته قرار می‌گیرد که بیشتر قسمت ابتدائی آن را داده‌های نرمال به خود اختصاص داده و سپس قسمت اندکی هم از آن داده‌های نویزی بوده و در نهایت قسمت پایانی را داده‌های پرت کسب می‌کنند. مرز میان نواحی مختلف بر روی این طیف به صورت کاملاً آشکار و دقیق قابل تشخیص نبوده و بسته به کاربردهای متفاوت و با توجه رأی و قضاوت تحلیل‌گر داده انتخاب می‌شود. علاوه بر این برخی داده‌های نویزی می‌توانند با توجه به مدل خاص کشف داده‌ی پرت، امتیاز بالائی را کسب نموده و به عنوان داده‌ی پرت معرفی گردند. اما آن‌چه در مورد داده‌های پرت قطعی است آن است که همواره امتیاز به نسبت بالاتری را در مقایسه با داده‌های نویزی کسب می‌کنند. اما این مسئله خود به تنها نمی‌تواند مرزی را میان داده‌های نویزی و داده‌های پرت تعریف کند و در نهایت این سلیقه‌ی تحلیل‌گر داده خواهد بود که تمایز میان نویز و ناهنجاری را تنظیم می‌نماید.



شکل ۵.۲ طیف امتیاز یک داده به لحاظ میزان پرت‌بودن از نرمال تا پرت [۱۰]

برخی جهت این که میان نویز و ناهنجاری به لحاظ نام‌گذاری تمایز قائل شوند، به ترتیب از واژه‌های «داده‌های پرت ضعیف»<sup>۵۷</sup> و «داده‌های پرت قوی»<sup>۵۸</sup> استفاده می‌کنند. حذف داده‌های نویز به تنها کابردۀای ارزشمندی از جمله پاک‌سازی مجموعه داده دارد تا بتوان بعد از آن به سایر عملیات داده‌کاوی پرداخت. اگرچه که این داده‌ها خود به تنها ای برای تحلیل‌گر ارزشی ندارند، اما کشف و پاک‌سازی آن‌ها پیش از اعمال هر الگوریتم دیگری، کما کان یک مسئله‌ی مهم به حساب می‌آید [۱۰].

## ۵.۲ انواع روش‌های کشف داده‌های پرت

برای این که بتوانیم به هر کدام از داده‌های یک مجموعه داده امتیازی مبنی بر میزان پرت‌بودن نسبت دهیم و در نهایت با تعیین یک مقدار حد آستانه با قطعیت در مورد پرت‌بودن یا نبودن آن‌ها تصمیم بگیریم، نیازمند یک مدل تحلیل داده‌های پرت می‌باشیم که بتواند الگوهای نرمال را در مجموعه داده با دقیق بالائی تشخیص داده و مرزی را میان الگوهای نرمال و غیر نرمال مانند شکل ۵.۲ قائل شود. در ادامه، تعدادی از مدل‌های کلیدی جهت تحلیل داده‌های پرت را به اختصار توضیح خواهیم داد.

<sup>57</sup> Weak outliers

<sup>58</sup> Strong outliers

## ۱.۵.۲ مدل‌های مبتنی بر مقادیر کرانی

اگر توزیع احتمال داده‌های یک مجموعه‌داده را در نظر بگیریم، «مقادیر کرانی»<sup>۵۹</sup> آن نقاطی هستند که در نواحی انتهائی دو طرف این توزیع قرار گرفته‌اند. آن‌چه گفته شد در مورد یک توزیع تکبعدی بود، اما در مورد توزیع‌های چندبعدی نیز می‌توان همین مطلب را تعمیم داد. در واقع این روش داده‌های پرت را مطابق با «دبایله‌های آماری»<sup>۶۰</sup> توزیع‌های احتمال در نظر می‌گیرد. در اینجا ذکر این نکته مهم می‌باشد که مقادیر کرانی نوع خاصی از داده‌های پرت به حساب می‌آیند؛ به عبارت دیگر، همه‌ی مقادیر کرانی داده‌های پرت به حساب می‌آیند، اما عکس آن ممکن است صحت نداشته باشد. به عنوان مثال، مجموعه‌داده‌ی تکبعدی {1,3,3,3,50,97,97,97,100} را در نظر بگیرید. در این مجموعه‌داده، مقادیر ۱ و ۱۰۰ را شاید بتوانیم به عنوان مقادیر کرانی در نظر بگیریم، اما مقدار ۵۰ از آن‌جا که میانگین مجموعه‌داده به حساب می‌آید، یک مقدار کرانی نخواهد بود. این در حالی است که مقدار ۵۰ «منفردترین نقطه»<sup>۶۱</sup> در مجموعه‌داده به حساب می‌آید و بنابراین از دیدگاهی غیر از دیدگاه کرانی، خود مستحق آن است که یک داده‌ی پرت محسوب گردد.

تحلیل داده‌های پرت با استفاده از مقادیر کرانی که به اختصار آن را «تحلیل مقادیر کرانی» می‌نامیم، خود به تنها کاربردهای مهمی در زمینه‌ی کشف داده‌های پرت دارد و بنابراین نقش لاینفکی را در این زمینه بازی می‌کند. یکی از این کاربردهای مهم، زمانی است که بخواهیم امتیازهای پرت‌بودن با مقادیر حقیقی حاصله از یک مدل کشف داده‌ی پرت را به مقادیر دودوئی تبدیل نمائیم. به این ترتیب که آن دسته از امتیازهایی که مقادیر کرانی به حساب می‌آیند را ارزش یک داده و مابقی امتیازات را با مقدار صفر ارزش‌گذاری نمائیم. مقدار یک به معنای پرت‌بودن و مقدار صفر نیز به معنای نرمال‌بودن خواهد بود. تحلیل مقادیر کرانی چند متغیره نیز با تعمیم همان حالت تکمتغیره می‌تواند در مواردی کاربردی باشد که می‌خواهیم امتیازات حاصله از چند مدل تحلیل داده‌ی پرت را در ابتدا به یک امتیاز نهائی به ازای هر داده تبدیل کرده و سپس مانند قبل آن‌ها را به مقادیر دودوئی مبدل نمائیم. به عنوان مثال، اگر یک سیستم هواسنجی را در نظر بگیریم که امتیازهای پرت‌بودن به ازای هر موقعیت مکانی را با توجه به دو

<sup>59</sup> Extreme values

<sup>60</sup> Statistical tails

<sup>61</sup> Most isolated point

عامل دما و فشار به صورت جداگانه تولید نموده است، در نتیجه دو سری امتیاز به ازای هر داده (در اینجا هر موقعیت مکانی) خواهیم داشت، که می‌بایست این امتیازات را یک پارچه نموده و به یک امتیاز واحد تبدیل نمائیم و یا هم در نهایت امر مانند قبل، از این امتیازات واحد، یک مقدار دودوئی به معنای پرت‌بودن و یا نبودن استخراج نمائیم [۴].

## ۲.۵.۲ مدل‌های مبتنی بر احتمالات

مدل‌های مبتنی بر احتمالات، خود حالت تعمیم‌یافته‌ای از روش‌های تحلیل مقادیر کرانی چندمتغیره یا همان چندبعدی می‌باشند. در واقع، روش تحلیل مقادیر کرانی چندبعدی را که خود مبتنی بر «فاصله‌ی ماهalanobis»<sup>۶۲</sup> می‌باشد، می‌توان به عنوان یک «مدل مخلوط گاوین»<sup>۶۳</sup> با تنها یک جزء منفرد در مخلوط تصور نمود. با تعمیم این مدل به اجزاء مخلوط چندگانه، نه تنها می‌توان مقادیر کرانی چندبعدی را شناسائی نمود، بلکه داده‌های پرت سراسری نیز با این رویه کشف خواهند شد.

اصل اساسی «مدل‌های مولّد مخلوطی» این است که تصور نمائیم که داده‌های یک مجموعه‌داده توسط یک مخلوط با تعداد  $k$  توزیع مختلف و با توابع توزیع احتمال  $g_k, \dots, g_1$ ، طبق رویه‌ی زیر پدید آمده‌اند:

- ۱) یک جزء مخلوط را با «احتمال اولیه»<sup>۶۴</sup>  $\alpha_i$  به طوری که  $i \in \{1, \dots, k\}$  انتخاب می‌نمائیم. تصور می‌کنیم که جزء  $i$  انتخاب شده است.
- ۲) یک داده را با توجه به توزیع  $g_r$  تولید می‌نمائیم.

این مدل مولّد را با  $\mathcal{M}$  نام‌گذاری می‌نمائیم و هر داده در مجموعه‌داده توسط این مدل تولید می‌شود. در واقع، در مدل مبتنی بر احتمالات، از مجموعه‌داده‌ی  $\mathcal{D}$  به عنوان یک عامل تخمین پارامترهای این مدل استفاده می‌شود. در بیشتر موارد، در مدل‌های مبتنی بر احتمالات، از توزیع نرمال یا گاوین استفاده می‌شود، اما می‌توان از سایر توزیع‌ها نیز استفاده نمود.

<sup>62</sup> Mahalanobis distance

<sup>63</sup> Gaussian Mixture Model (GMM)

<sup>64</sup> Prior probability

بعد از این که پارامترهای مدل  $\mathcal{M}$  به درستی تشخیص داده شدند، داده‌های پرت همان داده‌های خواهند بود که احتمال و درستی این که توسط این مدل تولید شده باشند، بسیار پایین می‌باشد. آن‌چه مطرح شد، در واقع بازتاب همان تعریف هاوکینز از داده‌های پرت می‌باشد که در ابتدای این فصل به آن اشاره گردید [۴].

### ۳.۵.۲ مدل‌های مبتنی بر خوشبندی

از آن‌چه در مورد مدل‌های مبتنی بر احتمالات در قسمت قبل بیان شد، می‌توان به ارتباط میان رویه‌ی خوشبندی و رویه‌ی کشف داده‌های پرت پی‌برد. در خوشبندی، به دنبال «اجتماعات»<sup>۶۵</sup> داده‌ها هستیم، در حالی که در مورد تحلیل داده‌های پرت، به دنبال داده‌هایی هستیم که از همین اجتماعات، بسیار دور می‌باشند. بدین ترتیب، خوشبندی و کشف داده‌های پرت، یک ارتباط تکمیلی با یکدیگر خواهند داشت. با یک نگاه ساده می‌توان گفت که در یک مجموعه‌داده، هر داده یا به یک خوشی مشخص تعلق دارد و یا هم به عنوان یک داده‌ی پرت شناسائی می‌گردد. الگوریتم‌های خوشبندی، عمدتاً یک خاصیت «مدیریت داده‌های پرت»<sup>۶۶</sup> را در خود دارند تا داده‌هایی که در حوالی و یا نقاط دوری از خوشها قرار دارند، به خوشها نسبت ندهند.

روش‌های مبتنی بر خوشبندی، از یک تحلیل سراسری جهت شناسائی آن دسته از داده‌هایی که صلاحیت تشکیل یک خوش را دارند، استفاده می‌نمایند. به عنوان مثال، در شکل ۱.۲ داده‌هایی که در ناحیه‌ی  $0_3$  قرار دارند، مانند سایر داده‌های نرمالی در که در نواحی  $N_1$  و  $N_2$  تشکیل خوش داده‌اند، از کثرت و البته چگالی کافی برخوردار نیستند و در نتیجه به عنوان یک خوشی نرمال شناسائی نخواهند شد. همان‌طور که پیش از این نیز توضیح داده شد، این داده‌ها، داده‌های پرتی هستند که بر اثر تکرار رویه‌ی تولید مربوطه، به تعداد محدودی ایجاد شده‌اند.

از جمله مشکلاتی که در مورد روشهای مبتنی بر خوشبندی وجود دارد، آن است که این روشهای گاهی قادر به این نمی‌باشند که داده‌های نویزی حواشی خوشها (داده‌های پرت ضعیف) را به درستی از

<sup>65</sup> Crowds

<sup>66</sup> Outlier handling

داده‌های واقعاً منزوی (داده‌های پرت قوی) تمیز دهند. پر واضح است که مورد دوم از مورد اول، اهمیت و حساسیت بیشتری داشته و در نتیجه این یک ضعف جدی در مورد روش‌های مبتنی بر خوشبندی می‌باشد [۴].

#### ۴.۵.۲ مدل‌های مبتنی بر فاصله

در روش‌های مبتنی بر فاصله جهت کشف داده‌های پرت، با توجه به این نکته که داده‌های پرت در واقع آن دسته از داده‌هایی هستند که از «توابع متراکم»<sup>۶۷</sup> یا همان خوشبندی به اندازه‌ی کافی دور می‌باشند، یک راه طبیعی و البته «بسته به مورد»<sup>۶۸</sup> جهت انتساب یک امتیاز پرت‌بودن به هر داده به صورت زیر می‌باشد:

امتیاز پرت‌بودن مبتنی بر فاصله به ازای یک داده  $O$ ، فاصله‌ی آن داده از  $k^{\text{امین نزدیک‌ترین همسایه‌اش}}_k$  می‌باشد.

تعریفی که قید شد، از فاصله‌ی یک داده تا  $k^{\text{امین نزدیک‌ترین همسایه‌اش}}_k$  استفاده کرده و به عنوان عمومی‌ترین نوع تعریف برای مدل‌های مبتنی بر فاصله به حساب می‌آید. انواع دیگری از تعاریف، مانند میانگین فواصل یک داده تا  $k^{\text{-نزدیک‌ترین همسایگان}}_k$  آن نیز به کار می‌روند.

در مقایسه با روش‌های مبتنی بر خوشبندی، روش‌های مبتنی بر فاصله، از یک تحلیل عمیق‌تر و البته جزئی‌تر جهت تمیزدادن میان داده‌های نویزی و داده‌های نابهنجار (که مورد توجه تحلیل‌گر داده هستند) استفاده می‌نمایند. این مسئله به خاطر آن است که داده‌های نویز که در حاشیه‌ی خوش قرار دارند، به نسبت داده‌های نابهنجار، فاصله‌ی کمتری تا  $k^{\text{امین نزدیک‌ترین همسایه‌ی خود}}_k$  دارند و در نتیجه، امتیاز کمتری مبنی بر میزان پرت‌بودن کسب خواهند نمود.

اما همین تحلیل جزئی‌تر در مورد روش‌های مبتنی بر فاصله، سبب پیچیدگی محاسباتی بالائی می‌شود. چرا که می‌بایست جهت یافتن نزدیک‌ترین همسایگان هر داده، فاصله‌ی آن داده را با تمامی داده‌های

<sup>67</sup> Crowded regions

<sup>68</sup> Instance-specific

دیگر در مجموعه‌داده محاسبه نمائیم که هزینه‌ی زمانی آن از مرتبه‌ی  $O(n^2)$  خواهد بود و  $n$  همان تعداد داده‌ها می‌باشد [۴].

## ۵.۵.۲ مدل‌های مبتنی بر چگالی

روش‌های مبتنی بر چگالی، همچون روش‌های مبتنی بر فاصله، از همان فواصل هر داده تا نزدیک‌ترین همسایگان آن داده، جهت تشخیص نواحی تُنک در مجموعه‌داده استفاده می‌نمایند. در واقع همان‌طور که پیش از این قید گردید، داده‌های پرت در نواحی پرجمعیت یا همان اجتماعات یافت نمی‌شوند، بلکه در نواحی با چگالی کمتر یا همان تُنک می‌باشد آن‌ها را جست و جو نمود [۴]. به همین منظور، در روش‌های مبتنی بر چگالی، در ابتدا باید یک نوع همسایگی خاص جهت بررسی به ازای هر داده تعریف نمائیم. این همسایگی، می‌تواند مانند آن‌چه در مورد روش‌های مبتنی بر فاصله مطرح شد، شامل  $k$ -نزدیک‌ترین همسایه‌ی هر داده باشد. در برخی موارد، این همسایگی را گسترش داده و داده‌هایی را که با داده‌ی مورد بررسی، در  $k$ -نزدیک‌ترین همسایگان‌اشتارک دارند نیز به این همسایگی اضافه می‌کنند. به این داده‌ها، «نزدیک‌ترین همسایگان مشترک»<sup>۶۹</sup> گویند. همین‌طور گاهی نیز، داده‌هایی را که داده‌ی مورد بررسی، یکی از  $k$ -نزدیک‌ترین همسایگان آن داده‌ها می‌باشد، به این مجموعه‌ی همسایگی اضافه می‌نمایند. این داده‌ها را نیز «نزدیک‌ترین همسایگان معکوس»<sup>۷۰</sup> نامند. پس از تعریف مجموعه‌ی همسایگی مورد نظر، می‌باشد فاصله‌ی هر داده را با همسایگان آن محاسبه نموده و به نوعی با استفاده از آن‌ها برای هر داده، یک مقدار «چگالی احتمال محلی»<sup>۷۱</sup> یا همان «میزان توزیع چگالی در محل یک داده»<sup>۷۲</sup> را تعریف نمائیم. در نهایت، جهت تعریف یک امتیاز به معنای میزان پرت‌بودن به ازای هر داده، می‌باشد مجموع مقادیر چگالی احتمال محلی به ازای همسایگان یک داده را بر مقدار چگالی احتمال محلی آن داده تقسیم نموده و حاصل را با یک عبارت «نرمال‌سازی»<sup>۷۳</sup>، تعدیل نمائیم. این عبارت نرمال‌سازی، به ازای هر داده می‌تواند متفاوت بوده و معمولاً همان تعداد داده‌های موجود در مجموعه‌ی

<sup>69</sup> Shared Nearest Neighbors (SNN)

<sup>70</sup> Reverse Nearest Neighbors (RNN)

<sup>71</sup> Local Probability Density

<sup>72</sup> Density distribution at the location of an object

<sup>73</sup> Normalization

همسایگی آن داده می‌باشد. در پایان، می‌توان لیست امتیازات حاصله را به صورت نزولی مرتب نموده و تعداد داده‌های پرت درخواستی کاربر را از بالای این لیست انتخاب نماییم و به عنوان خروجی ارائه دهیم. آن‌چه گفته شد، بنای عمدۀ روش‌های مبتنی بر چگالی می‌باشد که همان‌طور که مشهود بود، به نوعی مبتنی بر فاصله نیز می‌باشند [۳، [۱۱].

## ۶.۵.۲ مدل‌های مبتنی بر تئوری اطلاعات

داده‌های پرت، داده‌هایی هستند که به طور طبیعی و ذاتی، با توزیع ماقبی مجموعه‌داده همخوانی ندارند. بدین ترتیب، اگر بخواهیم توزیع احتمالاتی یک مجموعه‌داده را با توجه به الگوهای نرمال موجود در آن به نوعی فشرده‌سازی نمائیم، داده‌های پرت موجب خواهند شد تا «کمینه‌ی طول کد»<sup>۷۴</sup> لازم جهت بازنمائی این توزیع، افزایش یابد. به عنوان مثال، به دو رشته‌ی زیر توجه نمائید:

ABABABABABABABABABABABABABAB  
ABABA**C**ABABABABABABABABABABABAB

همان‌طور که پیداست، هر دو رشته از طول یکسان برخوردارند، اما رشته‌ی دوم تنها در یک موقعیت که شامل نماد یکتای C می‌شود، متفاوت است. رشته‌ی اول را می‌توان به صورت مختصر «AB ۱۷ بار» توصیف نمود. اما رشته‌ی دوم به دلیل دارابودن موقعیت یکتای مطابق با نماد C، نمی‌تواند به همین صورت مختصر توصیف شده و می‌بایست با کدی طولانی‌تر بازنمائی گردد. به عبارت دیگر، حضور نماد C در رشته‌ی دوم، سبب می‌شود تا «کمینه‌ی طول توصیف»<sup>۷۵</sup> آن رشته افزایش یابد. در اینجا به راحتی می‌توان گفت که نماد C در رشته‌ی دوم، مطابق با یک داده‌ی پرت می‌باشد.

با توجه به مثالی که مطرح شد، می‌توان این‌گونه بیان نمود که روش‌های مبتنی بر تئوری اطلاعات نیز مبتنی بر همین اصل عمومی هستند که تغییرات در اندازه‌ی یک مدل بررسی داده‌ی پرت (مانند کمینه‌ی طول توصیف و یا «آنتروپی»<sup>۷۶</sup>) را جهت توصیف میزان پرت‌بودن به ازای هر داده بررسی می‌نمایند [۴، [۱۲].

<sup>۷۴</sup> Minimum Code Length

<sup>۷۵</sup> Minimum Description Length

<sup>۷۶</sup> Entropy

## ۶.۲ انواع خروجی روش‌های کشف داده‌های پرت

هر مدل کشف داده‌ی پرت، در پایان امر، یک خروجی را به ازای هر داده مبنی بر «میزان پرت‌بودن»<sup>۷۷</sup> ارائه می‌نماید. این خروجی به ازای هر داده به دو صورت زیر می‌تواند ارائه شود:

- امتیاز پرت‌بودن با مقدار حقیقی: چنین امتیازی، در واقع گرایش به این که داده‌ی مربوطه به چه میزان پرت می‌باشد را به صورت کمی بیان می‌کند. مقادیر بالای این امتیاز، درستنمایی این که داده‌ی مربوطه پرت باشد را افزایش (یا در برخی موارد، کاهش) می‌دهد. برخی الگوریتم‌ها نیز ممکن است که یک مقدار احتمالاتی مابین صفر و یک را به عنوان امتیاز پرت‌بودن گزارش دهند [۴]. نوع امتیاز با مقدار حقیقی، بسیار مرسوم می‌باشد، چرا که همگی اطلاعات حاصله از الگوریتم مربوطه را به همراه دارد، اما نمی‌توان با استفاده از آن به تعداد مشخص و البته اندک داده‌های پرت پی برد [۱۰].
- برچسب دودوئی: یک مقدار دودوئی به عنوان خروجی الگوریتم ارائه می‌شود. بدین معنی که داده‌ی مربوطه پرت می‌باشد یا نه. این نوع خروجی، اطلاعات کمتری را به نسبت نوع اول در بر می‌گیرد، چرا که با اعمال یک حد آستانه بر روی نوع اول، می‌توان آن را به صورت دودوئی درآورد. هر چند که عکس این عمل ممکن نمی‌باشد. علی‌رغم این که ارائه‌ی یک مقدار حقیقی به عنوان خروجی الگوریتم معتبرتر می‌باشد، اما در مورد اکثریت کاربردهای خاص، می‌بایست یک مقدار دودوئی را به عنوان یک تصمیم سخت و قطعی گزارش نمود [۱۰, ۴].

## ۷.۲ اهمیّت و ضرورت کشف داده‌های پرت

ضرورت و اهمیت کشف داده‌های پرت، با توجه به این مهم می‌باشد که داده‌های پرت در مجموعه‌داده، معمولاً به اطلاعاتی پرمعنا (و غالباً حیاتی) تفسیر می‌شوند که حتی در برخی موارد قابلیت پیگرد قانونی خواهند داشت. این اطلاعات در مورد دامنه‌های کاربردی متفاوت، متنوع بوده و در هر مورد، تعبیری مختص به خود را دارند [۱]. برخی از موارد کاربردی کشف داده‌های پرت به قرار زیر می‌باشند:

- پاکسازی مجموعه‌داده<sup>۷۸</sup>: از آنجا که تعداد داده‌های پرت قوی که مورد توجه تحلیل‌گر داده هستند، به نسبت داده‌های پرت ضعیف که همان نویز به حساب می‌آیند، اندک می‌باشد، لذا رویه‌ی کشف داده‌های پرت،

<sup>77</sup> Outlierness degree

<sup>78</sup> Data cleaning

- می‌تواند به حذف این داده‌های نویزی مزاحم کمک نماید. نویز می‌تواند ناشی از خطا در فرآیند جمع‌آوری داده بوده و در رویه‌ی داده‌کاوی می‌تواند نقش مضری ایفا نماید [۴].
- **پردازش درخواست‌های وام<sup>۷۹</sup>:** در موارد بانک‌ها و مؤسسه‌تای که وام به مشتریان خود اعطاء می‌کنند، کشف ناهنجاری می‌تواند اطلاق به یافتن سیستم‌های کلامبردار و یا افراد بالقوه مشکل‌سازی باشد که قصد سوء استفاده از آن مؤسسه را داشته و می‌خواهند خلاف ضوابط قانونی، اخذ وام نمایند [۵].
  - **عملکرد شبکه<sup>۸۰</sup>:** با نظرات بر عملکرد کامپیوترهای یک شبکه، می‌توان به برخی خواص پنهان آن مانند گلوگاه‌های شبکه پی برد [۵].
  - **سیستم‌های کشف نفوذ<sup>۸۱</sup>:** در بسیاری از سیستم‌های کامپیوترا مبتنی بر میزبان یا شبکه شده، انواع مختلف داده جمع‌آوری می‌شود. این داده‌ها مشتمل بر فراغویی‌های سیستم عامل، میزان ترافیک شبکه و یا فعالیت‌های دیگر در شبکه می‌شود. این داده‌ها ممکن است رفتار غیرمعمولی را به جهت فعالیت بدخواهانه از خود نشان دهند. کشف چنین فعالیت‌هایی را کشف نفوذ در شبکه می‌خوانند [۱۰].
  - **کلامبرداری در مورد کارت‌های اعتباری<sup>۸۲</sup>:** کلامبرداری در زمینه‌ی کارت‌های اعتباری، یک امر بسیار شایع می‌باشد. چرا که در این مورد خاص، اطلاعات حساسی چون شماره‌ی کارت اعتباری و یا سایر شماره‌های خاص موجود بر روی کارت، می‌تواند به سادگی در اختیار افراد سودجو قرار بگیرد [۱۰]. بدین ترتیب، فعالیت‌های مشکوک و غیر معمول کارت‌های اعتباری، می‌تواند نشانی از به سرقت‌رفتن و سوء استفاده از کارت اعتباری باشد. با توجه به این‌که الگوهای غیر نرم‌الی چون این، به نسبت الگوهای نرم‌الی که همان استفاده‌ی معمول صاحب کارت می‌باشد، شدیداً به ندرت اتفاق می‌افتد، می‌توان آن‌ها را به عنوان داده‌های پرت شناسائی نمود [۴].
  - **رخدادهای جالب توجه حسگرهای<sup>۸۳</sup>:** از حسگرهای غالباً در مورد ردیابی و اندازه‌گیری پارامترهای محیطی و مکانی در بسیاری از کاربردهای دنیای واقعی استفاده می‌شود. تغییرات ناگهانی در الگوهای اصلی و زیربنایی مربوط به اطلاعات حاصله، می‌تواند نشان‌گر رخدادهای جالب توجه باشد. کشف رخداد، خود یکی از کاربردهای آغازین و جذاب در زمینه‌ی شبکه‌های حسگری می‌باشد [۱۰].

<sup>79</sup> Loan application processing

<sup>80</sup> Network performance

<sup>81</sup> Intrusion Detection Systems

<sup>82</sup> Credit card fraud

<sup>83</sup> Interesting sensor events

- عیب‌شناسی پزشکی<sup>۸۴</sup>: در بسیاری از کاربردهای پزشکی، دادگان لازم از طرق گوناگونی مانند تصاویر MRI تصاویر PET و یا سری‌های زمانی ECG جمع‌آوری می‌شوند. الگوهای غیرنرمال در چنین داده‌هایی مانند ناموزونی در نوار قلبی، معمولاً نشان از وجود یک بیماری و یا یک نارسائی می‌باشد [۵، ۱۰].
- اجرای احکام قضائی<sup>۸۵</sup>: کشف داده‌های پرت در زمینه اجرای احکام قضائی کاربردهای فراوانی دارد. کشف کلاهبرداری در مواردی مانند تراکنش‌های مالی، فعالیت‌های تجاری و یا حتی ادعاهای اخذ بیمه، معمولاً نیازمند آن است که الگوهای غیر نرمال در مجموعه‌داده ضبط شده را که توسط یک موجودیت مجرم ایجاد شده‌اند، شناسائی نمائیم. طبیعی است که این الگوهای غیر نرمال در گذر زمان و توسط فعالیت‌های گوناگون و نامحسوس یک موجودیت ایجاد شده و کشف آن‌ها از ارزش بالائی به لحاظ قانونی و قضائی برخوردار می‌باشد [۱۰].
- دانش زمین‌شناسی<sup>۸۶</sup>: مقدار قابل توجهی از دادگان «قضائی-زمانی»<sup>۸۷</sup> در مورد الگوهای آب و هوایی، تغییرات اقلیمی و یا الگوهای پوشش زمینی از طریق مکانیزم‌های گوناگونی مانند ماهواره‌ها و یا «حس‌کردن از راه دور»<sup>۸۸</sup> جمع‌آوری می‌شوند. کشف داده‌های نابهنجار در چنین دادگانی، بینش‌های ارزشمندی را در زمینه گرایش‌های انسانی و یا محیطی موجب می‌شوند که در واقع همین مسائل سبب ایجاد چنین نابهنجاری‌ها و ناسازگاری‌هایی در مجموعه‌داده شده‌اند [۱۰].
- تحقیقات داروئی<sup>۸۹</sup>: در زمینه تحقیقات مرتبط با دارو، همواره به دنبال تولید داروهایی نوین و مؤثر و البته با عوارض کمتر جهت درمان بیماری‌های رو به رشدی مانند سلطان هستیم. این مهم مستلزم آن است که ساختارهای مولکولی نوین موجود در عامل بیماری را با دقت شناسائی نموده و به دنبال ساخت داروئی جدید با ساختاری مؤثر باشیم [۵].
- کشف داده‌های به اشتباه برچسب‌خورده<sup>۹۰</sup>: در کاربردهایی که نیاز است تا از یک مجموعه‌داده به جهت آموزش یک مدل یادگیری مانند یک شبکه‌ی عصبی استفاده نماییم، ممکن است برخی از داده‌ها متعلق به یک کلاس خاص باشند، اما برچسبی که به آن‌ها تعلق گرفته است مربوط به کلاس دیگری باشد. کشف این‌گونه داده‌ها در چنین کاربردهایی، می‌تواند در بهبود رویه‌ی یادگیری نقش مؤثری ایفا نماید [۵].

<sup>84</sup> Medical diagnosis

<sup>85</sup> Law enforcement

<sup>86</sup> Earth science

<sup>87</sup> Spatiotemporal

<sup>88</sup> Remote sensing

<sup>89</sup> Pharmaceutical research

<sup>90</sup> Mislabeled data detecting

- عیب‌شناسی خرابی<sup>۹۱</sup>: شناسائی علت خرابی یا عملکرد دور از انتظار ادوات فیزیکی یک سیستم، مانند یک موتور محرك، مولد برق، خطوط لوله کشی نفت یا گاز و امثال آن و یا ابزارآلات فضایی مربوط به یک فضاپیما از جمله مسائل مهم و حیاتی‌ای می‌باشد که به صورت کلی به عنوان عیب‌شناسی خرابی از آن یاد می‌شود. با نظارت و دیده‌بانی فرآیندهای این ابزارآلات، می‌توان چنین نقص‌های نادری را شناسائی نمود که در نهایت منجر به بهبود عملکرد کلی سیستم مربوطه خواهد شد [۵].

## ۸.۲ شباهت رویه‌ی کشف داده‌های پرت با رویه‌ی کشف اقلام نوظهور

مبحث دیگری که به مبحث کشف داده‌های پرت شباهت بالائی دارد و به نوعی بنای هر دوی آن‌ها یکسان می‌باشد، مبحث «کشف اقلام نوظهور»<sup>۹۲</sup> نام دارد. در کشف اقلام نوظهور، به دنبال یافتن الگوهایی تاکنون مشاهده‌نشده (یا «تازه‌پدیدارشده»<sup>۹۳</sup> و یا «نوین»<sup>۹۴</sup>) در مجموعه‌داده هستیم که به نوعی مستلزم آن است که دادگان ما در حال رشد و از نوع «جريان‌داده»<sup>۹۵</sup> باشد. به عنوان مثال، می‌توان به کشف موضوع‌های مورد بحث جدید در یک گروه خبری اشاره نمود [۱]. مثال دیگر، یک سایت اینترنتی وابسته به یک رسانه‌ی اجتماعی است که در آن به مرور زمان، محتواهای جدیدی ظهر کرده و رویه‌ی کشف اقلام نوظهور می‌تواند به شناسائی موضوع‌ها و گرایش‌های جدید در یک روند وابسته به زمان کمک نماید [۲]. نکته این جاست که موضوعات جدید در یک مجموعه‌داده‌ی رو به رشد، در ابتدا به عنوان داده‌ی پرت شناسائی می‌شوند. تا به اینجای کار، هر دوی رویه‌های کشف داده‌های پرت و کشف اقلام نوظهور، شباهت‌های بسیاری به لحاظ روش‌های مدل‌سازی و کشف با یکدیگر دارند. هر چند، تفاوت مهم میان این دو رویه، آن است که در رویه‌ی کشف اقلام نوظهور، زمانی که موضوعات جدید پس از کشف‌شدن، مورد تأیید واقع می‌شوند، دیگر به عنوان داده‌ی نابهنجار به آن‌ها نگریسته نشده و معمولاً به عنوان یک مدل نرمال، به مجموعه‌ی مدل‌های نرمال موجود اضافه می‌شوند و البته داده‌هایی که از

<sup>91</sup> Fault diagnosis

<sup>92</sup> Novelty detection

<sup>93</sup> Emergent

<sup>94</sup> Novel

<sup>95</sup> Stream data

این قسم بعداً به مجموعه‌داده اضافه می‌گرددند نیز دیگر به عنوان داده‌ی پرت با آن‌ها برخورد نخواهد شد [۱، ۲].

## ۹.۲ چالش‌های موجود در زمینه‌ی کشف داده‌های پرت

کشف داده‌های پرت از هر نوعی، از اهمیت ویژه‌ای برخوردار می‌باشد، ولی این مسئله نیز مانند بسیاری از مسائل دیگر در زمینه‌ی داده‌کاوی، از چالش‌ها و مصائب خاص خود برخوردار می‌باشد. در ادامه به ذکر برخی از این چالش‌ها بسنده می‌نمائیم:

- **مدل‌سازی داده‌های نرمال و غیرنرمال به شکل مؤثر و سودمند:** کیفیت عملیات کشف داده‌های پرت از هر نوعی شدیداً به نحوی مدل‌سازی داده‌های نرمال و غیرنرمال وابسته است. غالباً ساختن یک مدل جامع که توانایی تشخیص داده‌های نرمال را از غیرنرمال، با دقت نسبتاً بالائی در هر مجموعه‌داده‌ای داشته باشد، اگر غیرممکن نباشد، شدیداً دشوار و طاقت‌فرساست. علت این امر را نیز می‌توان در این مهم جستجو نمود که شمارش تمامی حالت‌های نرمال ممکن برای یک کاربرد و یا مسئله‌ای خاص، بسیار سخت و حتی گاهی غیرممکن می‌باشد. علاوه بر این مرز بین نرمال‌بودن و غیرنرمال‌بودن داده‌ها همیشه به طور کامل، آشکار و قطعی نمی‌باشد و حتی گاهی محدوده‌ی وسیعی از شک و تردید یا همان «ناحیه‌ی خاکستری»<sup>۹۶</sup> بین این دو ناحیه وجود دارد [۱، ۲].
- **وابستگی رویه‌ی کشف داده‌های پرت به نوع کاربرد مورد نظر:** اگر بخواهیم به لحاظ فنی، روش‌های کشف داده‌های پرت را بررسی نمائیم، انتخاب معیار شباهت و یا فاصله جهت مدل‌سازی صحیح داده‌های مجموعه و توصیف آن‌ها، خود یک مسئله‌ی چالش‌برانگیز و جدی می‌باشد. متأسفانه این مسئله‌ی مهم، جنبه‌ی عمومی نداشته و بسته به کاربرد مربوطه، متفاوت می‌باشد. به بیان دیگر، به ازای هر کاربرد مشخص، اقتضایات و نیازمندی‌های مسئله متفاوت خواهد بود. به عنوان مثال، در مورد مسائلی مانند تحلیل داده‌های پزشکی، تحلیل داده‌های یک نیروگاه اتمی و یا تحلیل و ارزیابی داده‌های مربوط به پرتاپ یک فضاییما، اندک انحرافی از نرم، شدیداً مخرب و خطرناک خواهد بود. اما در مورد داده‌های مربوط به مسئله تحلیل بازاریابی و ارزش سهام، میزان نوسانات در حیطه‌ی داده‌های نرمال بسیار بیشتر از مواردی که ذکر گردید می‌باشد و در نتیجه جهت انتساب یک داده به داده‌های پرت، به میزان انحراف بیشتری از حالت نرمال نیاز می‌باشد. با توجه به وابستگی شدید روش‌های کشف داده‌های پرت به نوع مسئله و کاربرد مربوطه، امکان این که بتوان یک روش سراسری را جهتِ اعمال به انواع مسائل به وجود آوریم، به نوعی غیرممکن بوده و می‌بایست هر روش را بنا به یک کاربرد خاص توسعه دهیم [۱، ۲].
- **مدیریت داده‌های نویزی در کشف داده‌های پرت:** همان‌طور که پیش از این نیز قید گردید، داده‌های پرت با داده‌های نویزی متفاوتند و در نتیجه مجموعه‌داده‌ای که قرار است بر روی آن پردازشی از جمله کشف داده‌های پرت صورت گیرد، بهتر آن است که پیش از شروع عملیات، پاکسازی شده و از هرگونه داده‌ی نویزی

<sup>۹۶</sup> Gray area

عاری باشد. اما نکته این جاست که نویز در هر گونه مجموعه‌داده به دلایل متعددی از جمله خطاهای محاسباتی، خطاهای مربوط به حسگرهای ضبط‌کننده و امثال آن می‌تواند در قالب یک انحراف در مقادیر ویژگی‌ها و یا حتی در قالب «مقادیر مفقودی»<sup>۹۷</sup> وجود داشته باشد. عواملی چون کیفیت پایین مجموعه‌داده، به لحاظ صحت و سُقُم مقادیر ویژگی‌ها و البته حضور داده‌های نویزی در آن، سبب مشکلات جدی در امر کشف داده‌های پرت می‌گردد و می‌توانند سبب بهم ریختگی نظم مجموعه‌داده شده و در نتیجه‌ی آن تشخیص داده‌های نرمال از واقعاً غیرنرمال شدیداً دشوار می‌گردند. علاوه بر این حضور داده‌های نویزی می‌تواند موجب شود که یک داده‌ی پرت در قالب یک داده‌ی نویزی به اصطلاح «مخفی»<sup>۹۸</sup> شده و روش کشف داده‌ی پرت قادر به تمیزداندن آن از نویز نباشد، و حتی گاهی ممکن است یک داده‌ی نویزی به اصطلاح «تغییر ظاهر»<sup>۹۹</sup> داده و به اشتباه به عنوان یک داده‌ی پرت شناسائی گردد. پیش از این قید گردید که رویه‌ی کشف داده‌های پرت، می‌تواند در بسیاری موارد، موجب یافتن داده‌های نویزی و در نهایت حذف آن‌ها از مجموعه‌داده نیز گردد. اما آن‌چه حائز اهمیت می‌باشد، آن است که تحلیل‌گر داده نیاز دارد تا داده‌های واقعاً نابهنجار یا همان داده‌های پرت قوی را جهت بررسی فرآیندی که آن‌ها را به وجود آورده است، مورد تحقیق و بررسی قرار دهد. این داده‌های پرت قوی هستند که در نهایت، موجب کسب بیش‌هایی در مورد ساختار نرمال درونی مجموعه‌داده و ساختارهای مخرب جانبی می‌شوند [۱, ۲].

- **تغییر ظاهر داده‌های پرت به صورت نرمال در گذر زمان:** از آن‌جا که داده‌های پرت، نتیجه‌ی فعالیت‌های بدخواهانه‌ی جانبی می‌باشند، فرآیندهای متخاصمی که از روی عناد، به دنبال تولید داده‌ی پرت و برهم‌زنن نظم نرمال موجود هستند، در گذر زمان، همواره در تلاشند تا داده‌های غیر نرمال تولیدی را با ظاهری نرمال‌تر جلوه دهند. بدین صورت، کشف این داده‌های مزاحم، به مرور زمان دشوارتر خواهد شد و نیازمند آن است که روش‌های موجود جهت کشف و ضبط آن‌ها نیز توانمندتر گرددن [۱].
- **تکامل مفهوم نرمال‌بودن در مورد جریان داده‌ها:** در بسیاری از دامنه‌های کاربردی، با مجموعه‌داده‌های روبرو هستیم که در گذر زمان رشد کرده و از ثبات داده‌های آفلاین برخوردار نمی‌باشند. در این‌گونه مجموعه‌داده‌ها، مفهوم نرمال‌بودن نیز با ورود داده‌های جدید، ثبات خود را از دست داده و تکامل پیدا می‌کند. بدین ترتیب، ممکن است یک مفهوم نرمال فعلی، در آینده‌ای نه چندان دور، دیگر به اندازه‌ی کافی نمایان گر نرمال‌بودن و البته قابل استناد نباشد [۱].
- **قابل فهم‌بودن علت انتساب عنوان پرت‌بودن به یک داده:** در بسیاری موارد نه تنها نیاز است تا داده‌های پرت یک مجموعه‌داده شناسائی گرددند، بلکه علت این‌که یک داده به عنوان داده‌ی پرت شناسائی می‌گردد نیز می‌بایست مشخص گردد. مثلاً در مورد داده‌های پزشکی، می‌بایست مشخص گردد که بنا به کدام ویژگی‌های ضبط‌شده برای یک فرد بیمار، آن فرد به عنوان عنصری استثنائی (مثلاً کسی که مبتلا به یک نوع سرطان نادر است) شناسائی شده است. برای این امر، می‌توان از روش‌های آماری بهره برد که در آن‌ها به هر داده مورد بررسی یک مقدار امتیاز مبنی بر میزان داده‌ی پرت‌بودن اطلاق می‌گردد و البته در کنار آن مشخص می‌گردد که بنا به کدام ویژگی‌ها، این داده به عنوان یک داده‌ی پرت شناسائی شده است [۲, ۱۳].

<sup>97</sup> Missing values

<sup>98</sup> Hide

<sup>99</sup> Disguise

- در دسترس بودن داده‌های برچسب‌خورده جهت آموزش و ارزیابی مدل‌های بانظارت: مدل‌های کشف داده‌های پرت، مشتمل بر سه نوع «بانظارت»<sup>100</sup>، «نیمه‌نظرارتی»<sup>101</sup> و «بدون نظرارت»<sup>102</sup> می‌باشند. در دو مورد اول، حضور داده‌های برچسب‌خورده به ترتیب به دو صورت کلی و جزئی ضروری می‌باشد. اما در مورد سوم، می‌بایست مدلی را طراحی نمائیم که بنا به شباهت‌های ذاتی میان داده‌ها و بدون استفاده از هر گونه برچسبی، در مورد پرت‌بودن یا نبودن آن‌ها تصمیم‌گیری نماید. در دسترس بودن داده‌های برچسب‌خورده، جهت آموزش و ارزیابی مدل‌های بانظارت و نیمه‌نظرارتی خود مسئله‌ی بزرگ و مهمی است که متوجه روش‌های جمع‌آوری داده می‌شود [۱، ۲].

<sup>100</sup> Supervised

<sup>101</sup> Semi-Supervised

<sup>102</sup> Unsupervised

۳

## فصل سوم

### مروری بر کارهای انجام شده

تشخیص داده‌های پرت به دلیل اهمیت و ضرورت آن در کاربردهای مختلف، تاکنون بسیار مورد توجه محققان بوده و روش‌های گوناگونی جهت کشف آن‌ها ارائه شده است. بیشتر روش‌های مطرح در زمینه‌ی کشف ناهنجاری را می‌توان در ۵ رویکرد عمده دسته‌بندی نمود: روش‌های مبتنی بر توزیع و احتمالات؛ روش‌های مبتنی بر فاصله؛ روش‌های مبتنی بر چگالی؛ روش‌های مبتنی بر خوشبندی؛ و روش‌های مبتنی بر تئوری اطلاعات. در این فصل نیز قصد داریم تا تعدادی از روش‌های محبوب در این زمینه را، برخی به تفصیل و برخی هم به طور مختصر شرح دهیم.

## ۱.۳ کشف داده‌های پرت محلی با استفاده از یک روش مبتنی بر چگالی و معرفی معیار بنیادی LOF

«برونیگ»<sup>۱۰۳</sup> و همکاران [۳]، روشی را جهت کشف داده‌های پرت محلی در «فضای اقلیدسی»<sup>۱۰۴</sup> ارائه داده‌اند که در آن در نهایت به هر داده، امتیازی مبنی بر میزان منفردبودن آن داده در ناحیه‌ی مربوط به خودش داده خواهد شد. به عبارتی، هر چه داده‌ی مربوطه در ناحیه‌ای چگال‌تر قرار داشته باشد، امتیاز کمتری به معنای میزان پرتبودن دریافت خواهد کرد، و بالعکس، هر چه این ناحیه تُنک‌تر باشد، امتیاز مربوطه بیشتر خواهد بود. چگالی در اینجا به معنای تابع توزیع چگالی احتمال و یا مانند آن نمی‌باشد، بلکه به معنای همان مفهوم عام چگالی، یعنی میزان تراکم داده‌ها حول یکدیگر می‌باشد. خلاصه‌ی این رویه آن است که هر چه چگالی اطراف یک داده نسبت به چگالی اطراف نزدیک‌ترین همسایگان آن داده کمتر باشد، در آن صورت مقدار ضریب داده‌ی پرت محلی یا همان LOF<sup>۱۰۵</sup> برای آن داده به مراتب بیشتر بوده و احتمال پرت محلی بودن آن نیز بیشتر خواهد بود.

حال در اینجا لازم است تا در ابتدا چند مفهوم پایه را جهت تعیین معیار LOF معرفی نمائیم. اگر مجموعه‌داده‌ی  $D$  را در فضای  $m$ -بعدی در نظر بگیریم و یک داده‌ی مورد بررسی را با  $p$  نشان دهیم، داریم:

<sup>103</sup> Markus M. Breunig

<sup>104</sup> Euclidean space

<sup>105</sup> Local Outlier Factor

(۱) تعریف اول: ( $k$ -فاصله‌ی داده‌ی  $p$ )<sup>۱۰۶</sup>

به ازای هر مقدار صحیح  $k$ , مقدار عددی  $k$ -فاصله‌ی داده‌ی  $p$  که به اختصار آن را به صورت  $k\text{-distance}(p)$  نشان می‌دهیم، در قالب فاصله‌ی  $d(p, o)$  یا همان فاصله‌ی داده‌ی  $p$  تا داده‌ی  $o \in D$  تعریف می‌شود، به گونه‌ای که:

i. به ازای حداقل تعداد  $k$  داده‌ی  $o' \in D \setminus \{p\}$ , این رابطه برقرار است:  $d(p, o') \leq d(p, o)$ ;

ii. به ازای حداکثر تعداد  $k-1$  داده‌ی  $o' \in D \setminus \{p\}$ , این رابطه برقرار است:  $d(p, o') < d(p, o)$ .

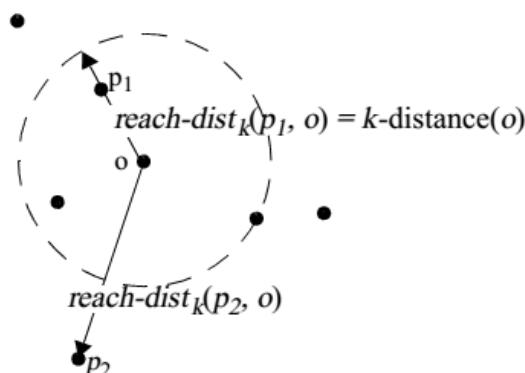
(توجه: منظور از  $D \setminus \{p\}$ , همان مجموعه‌داده‌ی  $D$  بدون احتساب نقطه‌ی  $p$  می‌باشد).

(۲) تعریف دوم: (مجموعه‌ی همسایگی  $k$ -فاصله‌ی داده‌ی  $p$ )<sup>۱۰۷</sup>

با توجه به مقدار  $k$ -فاصله‌ی داده‌ی  $p$ , مجموعه‌ی همسایگی  $k$ -فاصله‌ی داده‌ی  $p$  شامل تمامی داده‌هایی می‌شود که فاصله‌ی آن‌ها از  $p$ , بزرگتر از مقدار  $k$ -فاصله‌ی  $p$  نباشد. این مسئله را می‌توان به صورت زیر نشان داد:

$$N_{k\text{-distance}(p)}(p) = \{q \in D \setminus \{p\} \mid d(p, q) \leq k - \text{distance}(p)\} \quad (1.3)$$

این داده‌ها که با نماد  $q$  نشان داده شده‌اند را  $k$ -نزدیک‌ترین همسایگان داده‌ی  $p$  می‌نامند. از آن‌جا که ممکن است تعدادی از داده‌ها دارای دو نسخه در مجموعه‌داده باشند و نیز بر روی شعاع همسایگی  $k$ -فاصله‌ی یک داده، بیشتر از یک داده حضور داشته باشد، لذا اندازه‌ی مجموعه‌ی  $N_{k\text{-distance}(p)}(p)$  می‌تواند در برخی موارد بیشتر از  $k$  باشد. در ادامه، به جای استفاده از  $N_k(p)$ , به اختصار از  $N_{k\text{-distance}(p)}(p)$  استفاده خواهیم کرد.



شکل ۱.۳ فاصله‌ی دسترس‌پذیری داده‌های  $p_1$  و  $p_2$  با توجه به داده‌ی  $o$  [۳]

(۳) تعریف سوم: (فاصله‌ی دسترس‌پذیری داده‌ی  $p$  با توجه به داده‌ی  $o$ )<sup>۱۰۸</sup>

<sup>106</sup>  $k$ -distance of an object  $p$

<sup>107</sup>  $k$ -distance neighborhood of an object  $p$

اگر  $k$  را یک عدد طبیعی فرض کنیم، آن‌گاه فاصله‌ی دسترس‌پذیری داده‌ی  $p$  با توجه به داده‌ی  $o$ ، به صورت زیر تعریف می‌گردد:

$$\text{reachDist}_k(p, o) = \max\{k - \text{distance}(o), d(p, o)\} \quad (2.3)$$

شکل ۱.۳ مفهوم فاصله‌ی دسترس‌پذیری را با توجه به مقدار  $k=4$  نشان می‌دهد. همان‌طور که پیداست، هر دو داده‌ی  $p_1$  و  $p_2$  به اندازه‌ی کافی به داده‌ی  $o$  نزدیک هستند، اما با توجه به شاعع همسایگی تعریف شده، مقدار فاصله‌ی دسترس‌پذیری داده‌ی  $p_2$  از داده‌ی  $p_1$  بیشتر می‌باشد.

۴) تعریف چهارم: (چگالی دسترس‌پذیری محلی داده‌ی  $p$ )<sup>109</sup>

مقدار چگالی دسترس‌پذیری محلی به ازای داده‌ی  $p$ ، به صورت زیر تعریف می‌شود:

$$\text{lrD}_k(p) = \left( \frac{\sum_{o \in N_k(p)} \text{reachDist}_k(p, o)}{|N_k(p)|} \right)^{-1} \quad (3.3)$$

به طور شهودی پیداست که چگالی دسترس‌پذیری محلی برای داده‌ی  $p$ ، در واقع همان معکوس مقدار میانگین فواصل دسترس‌پذیری به ازای داده‌های موجود در مجموعه‌ی همسایگی آن داده می‌باشد.

۵) تعریف نهانی: (ضریب داده‌ی پرت (محلی) برای داده‌ی  $p$ )<sup>110</sup>

ضریب داده‌ی پرت محلی به ازای داده‌ی  $p$  به صورت زیر تعریف می‌شود:

$$\text{LOF}_k(p) = \frac{\sum_{o \in N_k(p)} \frac{\text{lrD}_k(o)}{\text{lrD}_k(p)}}{|N_k(p)|} \quad (4.3)$$

در نهایت، می‌بینیم که ضریب داده‌ی پرت محلی برای داده‌ی  $p$ ، به نوعی میزان منفردبودن آن را در محدوده‌ی همسایگی خویش نشان می‌دهد. این مقدار برابر با میانگین مقادیر نسبت چگالی دسترس‌پذیری محلی همسایگان داده‌ی  $p$  به چگالی دسترس‌پذیری خود  $p$  می‌باشد. پر واضح است که هر چه چگالی دسترس‌پذیری محلی داده‌ی  $p$  کمتر باشد، و در عین حال، چگالی دسترس‌پذیری محلی به ازای همسایگان آن بیشتر باشد، در آن صورت مقدار امتیاز LOF برای آن داده بیشتر خواهد بود.

<sup>108</sup> Reachability distance of an object  $p$  w.r.t object  $o$

<sup>109</sup> Local reachability density of an object  $p$

<sup>110</sup> (Local) Outlier Factor ( $LOF$ ) of an object  $p$

## ۲.۳ ضریب داده‌ی پرت محلی با مقدار احتمالاتی مابین صفر و یک (LoOP)

«کریگل»<sup>۱۱۱</sup> و همکاران [۱۴]، روشی را جهت کشف داده‌های پرت محلی ارائه داده‌اند که بنای آن همان LOF می‌باشد که پیش از این مطرح شد. اما فرق این روش در آن است که در نهایت به هر داده، امتیازی مطلقاً در بازه‌ی [۰,۱] اختصاص می‌یابد. امتیاز تعلق‌گرفته به هر داده به مفهوم میزان پرت‌بودن در این روش را به اختصار LoOP<sup>۱۱۲</sup> نامند. مزیت این روش در آن است که امتیاز حاصله از آن به صورت احتمالاتی قابل تفسیر می‌باشد. به عنوان مثال، می‌توان گفت که داده‌ی  $p$  با احتمال ۰,۸۵ پرت می‌باشد. این مسئله در حالی است که در مورد سایر روش‌های کشف داده‌های پرت که در نهایت به هر داده، امتیازی مبنی بر میزان پرت‌بودن نسبت می‌دهند، بسته به روش مورد استفاده، و یا بسته به مجموعه‌داده‌ی مربوطه و حتی ناحیه‌ای که داده در آن قرار گرفته است، مقدار این امتیاز می‌تواند تفاسیر متعددی داشته باشد. به عنوان مثال، در یک روش خاص، هر چه این امتیاز بالاتر باشد، داده‌ی مربوطه به هیچ عنوان پرت محسوب نخواهد شد. بالعکس در یک روش خاص دیگر، امتیاز بالاتر به منزله‌ی بیشتر نابهنجاربودن می‌باشد. همین‌طور امتیازی که از یک روش خاص به ازای یک مجموعه‌داده‌ی مشخص حاصل می‌شود، می‌تواند در یک مجموعه‌ی داده‌ی دیگر معنای متفاوتی داشته باشد. بدین معنی که در یکی، داده‌ی مربوطه پرت به حساب آمده و در دیگری اصلاً داده‌ی غیرمعمولی شناخته نمی‌شود. در برخی موارد نیز به ازای یک مجموعه‌داده‌ی مشخص، اما در نواحی با توزیع متفاوت، یک مقدار امتیاز ثابت می‌تواند مفهوم متفاوتی داشته باشد.

با توجه به این‌که در رویکردهای مبتنی بر چگالی مانند LOF، در صورت انتخاب یک مقدار نامناسب برای  $k$ ، ممکن است که یک داده‌ی پرت امتیاز درستی دریافت ننماید. لذا در این رویکرد احتمالاتی این مسئله رعایت شده است و زیرفضائی که هر داده در آن مورد بررسی قرار می‌گیرد، یک محدوده‌ی قطعی

<sup>111</sup> Hans-Peter Kriegel

<sup>112</sup> Local Outlier Probability (LoOP)

(مانند محدوده‌ی همسایگی  $k$ -فاصله‌ی یک داده) نبوده و بلکه نسخه‌ی «توسعه‌یافته‌ی آماری»<sup>۱۱۳</sup> آن می‌باشد.

بدین منظور در ادامه، مجموعه‌داده‌ی  $n$  عضوی مربوطه را با  $D$  و تابع فاصله‌ای که جهت تشخیص داده‌های پرت از آن استفاده می‌شود را با  $d$  نشان خواهیم داد. حال اگر بخواهیم محدوده‌ی همسایگی آماری یک داده‌ی  $o \in D$  را تعریف نمائیم، معیار «فاصله‌ی احتمالاتی»<sup>۱۱۴</sup> آن داده را نسبت به مجموعه‌ی قرائن و همسایگان آن که با  $S$  نشان می‌دهیم، به صورت  $pdist(o, S)$  نشان می‌دهیم. این معیار دارای خاصیت زیر می‌باشد:

$$\forall s \in S: P[d(o, s) \leq pdist(o, S)] \geq \varphi \quad (5.3)$$

از این عبارت می‌توان این‌گونه برداشت نمود که اگر یک کره را با شعاع  $pdist$  حول نقطه‌ی  $o$  تصور نمائیم، این کره، هر داده‌ای در مجموعه‌ی قرائن  $S$  را با احتمال  $\varphi$  در برخواهد گرفت. به عبارتی این کره، همان نسخه‌ی توسعه‌یافته‌ی آماری مجموعه‌ی قرائن  $S$  می‌باشد. تنها تفاوت حالت توسعه‌یافته‌ی آماری با حالت نرمال در آن است که در حالت آماری، مقداری نرخ خطای نیز در عضویت اعضای مجموعه‌ی قرائن  $S$  پذیرفته می‌شود. خاصیت دیگر این فاصله‌ی احتمالاتی در آن است که می‌توان از آن میزان چگالی داده‌ها در ناحیه‌ی  $S$  را نیز به صورت زیر تخمین زد:

$$pdens(S) = \frac{1}{pdist(o, S)} \quad (6.3)$$

در اینجا اگر بخواهیم به صورت عملیاتی از پارامتر  $\varphi$  استفاده نمائیم، می‌توانیم از «قانون سه-سیگما»<sup>۱۱۵</sup> و «تابع خطای گاووسین»<sup>۱۱۶</sup> استفاده کرده و یک پارامتر دیگر با نام  $\lambda$  را به صورت زیر معرفی نمائیم:

$$\lambda = \sqrt{2} \cdot erf^{-1}(\varphi) \quad (7.3)$$

<sup>113</sup> Statistical extent

<sup>114</sup> Probabilistic distance

<sup>115</sup> Three-Sigma rule

<sup>116</sup> Gaussian error function

مقدار  $\lambda$  می‌تواند هر کدام از مقادیر ۱ الی ۳ در قانون سه-سیگما باشد. داریم:

$$\lambda = 1 \Leftrightarrow \varphi \approx 68\%; \quad \lambda = 2 \Leftrightarrow \varphi \approx 95\%; \quad \lambda = 3 \Leftrightarrow \varphi \approx 99.7\% \quad (8.3)$$

همان‌طور که می‌دانید، حُسن استفاده از قانون سه-سیگما آن است که می‌توان مانند آن‌چه پیش از این در مورد مقادیر کرانی مطرح شد، داده‌هایی را که بیش از  $\lambda$  ضرب در مقدار «انحراف از معیار»<sup>۱۱۷</sup>  $\sigma$ ، از میانگین فاصله دارند را به عنوان داده‌ی پرت در نظر گرفت. حال در اینجا مقدار انحراف از معیار  $\sigma$  را به صورت زیر تعریف می‌نمائیم:

$$\sigma(o, S) = \sqrt{\frac{\sum_{s \in S} d(o, s)^2}{|S|}} \quad (9.3)$$

به عبارت دیگر در این‌جا، نقطه‌ی ۰ را مرکز ثقل محدوده‌ی  $S$  تصور نموده‌ایم که در آن صورت مقدار میانگین برابر با  $\mathbf{0} = \mathbf{d}(o, o)$  خواهد بود. لازم به ذکر است که این مسئله با محاسبه‌ی امید ریاضی  $d(o, S)$  متفاوت می‌باشد. تفاوت موجود در آن است که در این‌جا قادر به بیان این مسئله نمی‌باشیم که مقادیر فواصل یک داده با دادگان همسایه را دارای توزیع نرمال فرض نمائیم، بلکه در عوض می‌توانیم این‌گونه تصور نمائیم که مجموعه‌ی قرائن  $S$  به صورت نرمال در اطراف نقطه‌ی ۰ پراکنده شده‌اند. این موضوع، با توجه به این‌که ما نیز مجموعه‌ی قرائن  $S$  را با استفاده از  $k$ -نزدیک‌ترین همسایگان یک داده به دست می‌آوریم، عموماً منطقی بوده و صحت دارد.

با توجه به آن‌چه در مورد انحراف از معیار فواصل همسایگان یک داده از خود داده و البته قانون سه-سیگما مطرح شد، می‌توانیم مقدار چگالی اطراف داده‌ی ۰ را با توجه به  $S$  به صورت زیر تخمین بزنیم:

$$pdist(\lambda, o, S) = \lambda \cdot \sigma(o, S) \quad (10.3)$$

این مقدار را «فاصله‌ی مجموعه‌ی احتمالاتی»<sup>۱۱۸</sup> داده‌ی ۰ به  $S$  با توجه به «درجه‌ی اهمیت»<sup>۱۱۹</sup>  $\lambda$  می‌نامیم. مقدار پارامتر  $\lambda$ ، در واقع به نوعی تعیین‌کننده‌ی شدت تغییرات در امتیازات نهایی خواهد بود و در ترتیب امتیازات حاصله تأثیری نخواهد داشت.

<sup>117</sup> Standard deviation

<sup>118</sup> Probabilistic set distance

در ادامه، باید مانند روش LOF که پس از تعریف یک مقدار چگالی به ازای هر داده، مقدار امتیاز نهائی را با توجه به مقادیر چگالی همسایگان هر داده تعیین می‌نمود، عمل نمائیم. بدین‌منظور مقدار «ضریب داده‌ی پرت محلی احتمالاتی»<sup>۱۲۰</sup> یک داده‌ی  $o \in D$  را با توجه به درجه‌ی اهمیت  $\lambda$  و مجموعه‌ی قرائن  $S(o) \subseteq D$  به صورت زیر تعریف می‌نمائیم:

$$PLOF_{\lambda,S}(o) = \frac{pdist(\lambda, o, S(o))}{E_{s \in S(o)}[pdist(\lambda, s, S(s))]} - 1 \quad (11.3)$$

همان‌طور که پیداست، مقدار PLOF برای داده‌ی  $o \in D$  در واقع نسبت چگالی آماری اطراف این داده به مقدار میانگین چگالی‌های آماری داده‌های همسایه‌ی آن می‌باشد. این که در این فرمول، مقدار منهای یک مشاهده می‌شود، از آن جهت است که ما مقادیر PLOF اصلی را دارای توزیعی با میانگین برابر یک فرض نموده‌ایم و جهت این که بتوانیم ازتابع خطای گاووسین استفاده نمائیم، باید این مقادیر را به اصطلاح استانداردسازی نمائیم. بدین‌ترتیب که آن‌ها را منهای مقدار میانگین یعنی یک نموده و در نهایت بر مقدار انحراف از معیار این امتیازات تقسیم کنیم. مرحله‌ی اول که کسر میانگین بود به صورت ذاتی اعمال شده است. اما جهت کسب مقدار انحراف از معیار به صورت زیر عمل می‌نمائیم:

$$nPLOF = \lambda \cdot \sqrt{E[(PLOF)^2]} \quad (12.3)$$

در نهایت، جهت تعریف مقدار ضریب داده‌ی پرت محلی احتمالاتی به ازای هر داده به صورت زیر عمل می‌نمائیم:

$$LoOP_S(o) = \max \left\{ 0, \operatorname{erf} \left( \frac{PLOF_{\lambda,S}(o)}{nPLOF \cdot \sqrt{2}} \right) \right\} \quad (13.3)$$

این‌که چرا در این‌جا به دنبال مقدار بیشینه‌ی صفر و خروجی تابع خطای گاووسین می‌باشیم، از آن روست که خروجی این تابع در برخی موارد منفی می‌شود. مقدار منفی یعنی این‌که داده‌ی مربوطه تحت هیچ شرایطی پرت نمی‌باشد و از آن‌جا که مقدار خروجی این روش، یک مقدار احتمال مابین صفر و یک می‌باشد، باید به ازای این داده‌ها مقدار صفر را به عنوان امتیاز پرت‌بودن تعریف نمائیم.

<sup>119</sup> Significance

<sup>120</sup> Probabilistic Local Outlier Factor (PLOF)

بنابراین، هر چه که مقدار PLOF به صفر نزدیک‌تر باشد، احتمال پرت‌بودن آن کمتر می‌باشد و بالعکس. حُسن این مقدار امتیاز در آن است که علی‌رغم سایر روش‌های مبتنی بر چگالی (که امتیاز حاصله از آن‌ها را نمی‌توان به صورت سراسری چه در میان داده‌های یک مجموعه‌داده و چه در میان مجموعه‌داده‌های مختلف مقایسه نمود)، یک امتیاز مقایسه‌پذیر به صورت کلی می‌باشد.

### ۳.۳ کشف داده‌های پرت محلی در جریان داده‌ها با استفاده از یک روش

#### افزایشی

در مورد جریان داده‌ها، پیش از این نیز قید گردید که داده‌ها به مرور زمان به مجموعه‌داده افزوده می‌گردند و در نتیجه‌ی آن، نظم درونی مجموعه‌داده ثابت نخواهد بود. لذا نمی‌توانیم در مورد چنین مجموعه‌داده‌هایی، از روش‌هایی استفاده نمائیم که نیازمند آن هستند تا تمامی مجموعه‌داده را یکجا دریافت نموده و سپس در مورد آن تصمیم‌گیری کنند. «پوکراجاک»<sup>۱۲۱</sup> و همکاران [۱۵]، یک «ضریب داده‌ی پرت محلی افزایشی»<sup>۱۲۲</sup> را جهت کشف ناهنجاری در جریان داده‌ها معرفی نموده‌اند که در واقع توسعه‌یافته‌ی همان LOF می‌باشد که پیش از این نیز معرفی شد و البته در مورد مجموعه‌داده‌های ثابت و بلا تغییر کاربرد داشت. در ادامه، این امتیاز پرت‌بودن را با LOF نشان خواهیم داد. در این روش پیشنهادی، اثبات می‌شود که اگر بعد از اضافه‌شدن هر داده و حصول امتیاز افزایشی LOF، امتیاز ثابت LOF را مجدداً بر روی مجموعه‌داده تغییریافته محاسبه نمائیم، علاوه بر این که کسب LOF از زمان محاسباتی کمتری نسبت به LOF پایه برخوردار می‌باشد، خواهیم دید که مقادیر امتیازات یادشده با یکدیگر قربات بالائی داشته و البته که در نهایت امر، عملکرد هر دو الگوریتم به سبب این مسئله، یکسان خواهد بود.

الگوریتم LOF، به ازای هر داده که تاکنون مشاهده شده است، یک سری اطلاعات را در قالب «نمایه»<sup>۱۲۳</sup> نگهداری می‌نماید. نمایه‌های مربوط به تمامی داده‌ها، به مرور زمان و به صورت پویا به روز

<sup>121</sup> Dragoljub Pokrajac

<sup>122</sup> Incremental LOF (Local Outlier Factor)

<sup>123</sup> Profile

می‌شوند. این یک مسئله‌ی بسیار مهم در مورد نمایه‌ها می‌باشد، چرا که نمایه‌ی مربوط به هر داده، می‌تواند به مرور آمدن داده‌های جدید و تغییر در مجموعه‌ی همسایگی آن داده، دستخوش تغییر شود. تغییرات در نمایه‌ی هر داده، می‌تواند شامل افزوده شدن یک داده‌ی جدید به مجموعه‌ی همسایگی داده‌ی فعلی و یا حتی حذف یک داده از این مجموعه‌ی همسایگی باشد. می‌توان اثبات نمود که این تغییرات در نمایه‌ی هر داده، تنها تعداد محدودی از نزدیک‌ترین همسایگان آن داده را تحت تأثیر قرار داده و در نتیجه، رویه‌ی به روزرسانی مشتمل بر افزودن به و یا حذف یک داده از مجموعه‌ی همسایگی مربوطه، به همه‌ی داده‌های مجموعه‌داده وابسته نخواهد بود.

نتایج آزمایشات مربوط به iLOF<sup>124</sup> بر روی مجموعه‌داده‌های متنوع مصنوعی و واقعی، نشان می‌دهد که این روش، در عین حال که توزیع داده‌ها در جریان داده‌ها به مرور آمدن داده‌ها در حال تغییر است، با موفقیت قادر به شناسائی داده‌های پرت می‌باشد و همین‌طور به لحاظ محاسباتی کارآمد نیز می‌باشد.

## ۴.۳ جنگل جداسازی

یکی از جدیدترین و مطرح‌ترین روش‌ها در زمینه‌ی کشف داده‌های پرت، «جنگل جداسازی»<sup>125</sup> [۱۶] نام دارد که در ادامه‌ی این پایان‌نامه از آن، با نام اختصاری iForest<sup>126</sup> اسماً می‌بریم. «لیو»<sup>127</sup> و همکاران، در این روش، با توجه به این واقعیت که داده‌های پرت، همان داده‌هایی هستند که هم تعداد آن‌ها اندک است و هم نسبت به مابقی داده‌ها رفتار متفاوتی را از خود بروز می‌دهند، نشان می‌دهند که می‌توان داده‌های پرت را با استفاده یک مکانیزم خاص، تحت عنوان مکانیزم جداسازی، شناسائی نمود.

روش Forest<sup>128</sup>، بسیار مؤثر عمل کرده و نسبت به سایر روش‌های موجود در زمینه‌ی کشف داده‌های پرت، تفاوت‌های بنیادی دارد. در این روش، از مکانیزم جداسازی به عنوان یک ابزار مؤثر و کارآمد به جای معیارهای معمول فاصله و چگالی استفاده می‌گردد. علاوه بر این، این روش، یک الگوریتم با پیچیدگی زمانی خطی پایین بوده و حافظه‌ی مصرفی مورد نیاز آن نیز کم می‌باشد. این الگوریتم، به ازای هر

<sup>124</sup> Isolation Forest

<sup>125</sup> Fei Tony Liu

مجموعه‌داده‌ی مورد بررسی، یک مدل کارآمد را با تعداد اندکی درخت که با استفاده از زیرنمونه‌های با اندازه‌ی ثابت و فارغ از اندازه‌ی کل مجموعه‌داده تهیه شده‌اند، می‌سازد.

عملکرد کلی الگوریتم iForest بدین‌گونه است که تلاش می‌کند تا داده‌ها را تا آن‌جا که ممکن است نسبت به سایر داده‌ها جداسازی نماید و با توجه به تعداد شرایطی که برای این جداسازی نیاز خواهد بود، به هر داده امتیازی مبنی بر میزان پرت‌بودن تعلق خواهد گرفت. به این صورت که در ابتدا به صورت تصادفی، یک ویژگی را انتخاب نموده و سپس یک «مقدار شکست»<sup>۱۲۶</sup> را میان «مقدار بیشینه»<sup>۱۲۷</sup> و «مقدار کمینه»<sup>۱۲۸</sup> آن ویژگی بر می‌گزیند. حال اگر بخواهیم داده‌های پرت را جداسازی نمائیم، به تعداد شرایط محدود‌کننده‌ی کمتری نسبت داده‌های نرمال نیاز خواهیم داشت. علی‌رغم داده‌های پرت، جداسازی داده‌های نرمال به تعداد شرایط بیشتری نیاز خواهد داشت. با توجه به آن‌چه مطرح شد، می‌توان به ازای هر داده، با توجه به تعداد شرایط محدود‌کننده‌ای که جهت جداسازی آن نیاز خواهد بود، امتیازی مبنی بر میزان پرت‌بودن نسبت داد.

### ۵.۳ ماشین‌های بردار پشتیبان تک‌کلاسه

«اسکولکوف»<sup>۱۲۹</sup> و همکاران [۱۷]، روشی را جهت کشف داده‌های پرت ارائه داده‌اند که در آن تلاش می‌شود تا با توجه به دادگان ورودی، مدلی احتمالاتی ارائه گردد تا به وسیله‌ی آن بتوان داده‌های نرمال را به بهترین شکل ممکن برآذش نمود. نام این روش، «ماشین‌های بردار پشتیبان تک‌کلاسه»<sup>۱۳۰</sup> می‌باشد که در ادامه‌ی این پایان‌نامه به اختصار از آن با OCSVM یاد خواهیم کرد.

کلیّت این روش بدین صورت است که اگر تصور کنیم که مجموعه‌ی داده‌ی مربوطه، با توجه به توزیع احتمالاتی  $P$  به وجود آمده باشد، قصد ما آن است تا زیرمجموعه‌ی «ساده‌ی»<sup>۱۳۱</sup>  $S$  از فضای ورودی را به

<sup>126</sup> Split value

<sup>127</sup> Maximum value

<sup>128</sup> Minimum value

<sup>129</sup> Bernhard Schölkopf

<sup>130</sup> One Class Support Vector Machines (OCSVM)

<sup>131</sup> Simple

گونه‌ای تعریف کنیم تا با استفاده از آن بتوانیم با یک مقدار احتمالاتی، امکان حضور هر داده‌ای در خارج از  $S$  که با توجه به  $P$  تولید شده است را بیان نمائیم. داده‌هایی که درون  $S$  قرار داشته و به اصطلاح با استفاده از آن برازش شده‌اند، داده‌های نرمال و داده‌هایی که درون این ناحیه قرار نمی‌گیرند، داده‌های پرت خواهند بود.

در این روش، برای حل این موضوع تلاش می‌کنیم تا یک تابع  $f$  را به گونه‌ای تخمین بزنیم که خروجی آن به ازای نقاطی از  $P$  که درون  $S$  واقع شده‌اند، مثبت و نقاطی که خارج از آن واقع شده‌اند، منفی می‌باشد. قالب اساسی تابع  $f$  به این صورت ارائه می‌گردد که با یک «توسعه‌ی هسته»<sup>۱۳۲</sup> مرتبط با یک زیرمجموعه‌ی کوچک از مجموعه‌داده‌ی آموزشی، آن را به دست می‌آوریم. تابع  $f$  را با کنترل طول «بردار وزن»<sup>۱۳۳</sup> مرتبط با فضای تبدیل یافته‌ی مربوط به آن، تنظیم می‌نمائیم. ضرایب این بردار وزن نیز در اصل، با استفاده از حل یک «مسئله‌ی برنامه‌سازی درجه‌ی دوم»<sup>۱۳۴</sup> یافت می‌شوند. ولی ما در این روش، آن‌ها را در یک رویه‌ی بهینه‌سازی ترتیبی به دست می‌آوریم که از داده‌های ورودی به صورت جفت‌جفت به این منظور استفاده می‌نماید. در پایان باید گفت که این الگوریتم کشف داده‌ی پرت، در واقع یک حالت توسعه‌یافته‌ی طبیعی از الگوریتم ماشین‌های بردار پشتیبان یا همان SVM می‌باشد که در مورد مجموعه‌داده‌های به اصطلاح «نامیزان»<sup>۱۳۵</sup> بهبود یافته است. در مجموعه‌داده‌های نامیزان، تعداد داده‌های دو کلاسی که قصد جداسازی آن‌ها را از یکدیگر داریم (در اینجا داده‌های نرمال و پرت) با یکدیگر اختلاف فاحشی دارند.

<sup>132</sup> Kernel expansion

<sup>133</sup> Weight vector

<sup>134</sup> Quadratic programming problem

<sup>135</sup> Unbalanced

## ۶.۳ کشف داده‌های پرت در مجموعه‌داده‌های نامی و با مقیاس بزرگ با

### استفاده از یک رویکرد مبتنی بر تئوری اطلاعات

«وو»<sup>۱۳۶</sup> و «ونگ»<sup>۱۳۷</sup> [۱۲]، یک روش مبتنی بر «تئوری اطلاعات»<sup>۱۳۸</sup> جهت کشف داده‌های پرت در مورد داده‌های با مقادیر ویژگی نامی ارائه داده‌اند که در ادامه تنها با عنوان «داده‌های نامی»<sup>۱۳۹</sup> از آن‌ها اسم می‌بریم. در مورد داده‌های نامی، بزرگترین چالشی که وجود دارد آن است که چه معیار شباهت مناسبی میان داده‌ها تعریف کنیم تا به دنبال آن فواصل میان داده‌ها نیز با تقریب درستی به دست آمده و در نهایت صحت محاسبات ما نیز بالا باشد. در این روش، هدف آن است تا یک تعریف دقیق و رسمی را برای داده‌ی پرت ارائه نموده و همین‌طور یک مدل بهینه‌سازی را جهت کشف آن معرفی نمائیم، که از یک مفهوم جدید تحت عنوان «آنتروپی تام»<sup>۱۴۰</sup> بهره می‌برد. آنتروپی تام از دو مفهوم آنتروپی و «همبستگی تام»<sup>۱۴۱</sup> استفاده می‌نماید و در نهایت طی یک سری محاسبات و اثبات‌های ریاضیاتی، به همان تجمعی آنتروپی روی تک‌تک ویژگی‌های نامی خلاصه می‌شود. سپس بر اساس این مدل بهینه‌سازی، تابعی را جهت تعریف «ضریب داده‌ی پرت معرفی خواهیم نمود که ورودی آن، اطلاعات خود داده به تنهائی می‌باشد و البته که این مسئله یک نوآوری خاص به حساب می‌آید. چرا که در روش‌های معمول و شناخته‌شده‌ی کشف داده‌ی پرت، علاوه بر اطلاعات خود داده، به اطلاعات سایر داده‌های موجود از جمله همسایگان آن داده نیز جهت تعریف ضریب داده‌ی پرت احتیاج می‌باشد. علاوه بر بی‌نیازبودن ضریب داده‌ی پرت مربوطه از اطلاعات سایر داده‌ها، رویه‌ی به‌روزرسانی آن نیز بسیار سریع بوده و نیازی به انجام مجدد یک سری محاسبات سنگین روی کل مجموعه‌داده نمی‌باشد. در نهایت دو الگوریتم کشف داده‌ی پرت را معرفی خواهیم نمود که تنها ورودی آن‌ها، تعداد داده‌های پرت مورد درخواست کاربر می‌باشد و نیازی به این ندارند که کاربر چگونگی تعریف داده‌ی پرت را برای آن‌ها

<sup>136</sup> Shu Wu

<sup>137</sup> Shengrui Wang

<sup>138</sup> Information Theory

<sup>139</sup> Categorical Data

<sup>140</sup> Holoentropy

<sup>141</sup> Total correlation

مشخص نماید. الگوریتم اول که **ITB-SP** نام دارد، در یک رویه‌ی غیر تکراری یا به عبارتی در یک مرحله، داده‌های پرت را کشف نموده و به کاربر ارائه می‌نماید. اما الگوریتم دوم، که **ITB-SS** نام دارد، برخلاف الگوریتم اول در یک رویه‌ی تکراری و تدریجی داده‌های پرت را با دقت و ریزبینی بیشتری کشف نموده و در اختیار کاربر قرار می‌دهد. در ادامه در قسمت شرح روش و پارامترها به بیان جزئیات بیشتر در مورد این الگوریتم‌ها خواهیم پرداخت.

### ۱.۶.۳ شرح روش و پارامترها

در این قسمت در ابتدا به بیان این مسئله خواهیم پرداخت که چگونه آنتروپی و همبستگی تام در تعیین میزان پرت‌بودن هر داده و به عبارتی درستنمائی کاندیداهای داده‌ی پرت، ما را یاری خواهند نمود. سپس مفهوم آنتروپی تام را که از آنتروپی و همبستگی تام بهره می‌برد، به صورت فرمولی بیان خواهیم نمود. در ادامه مطرح خواهیم کرد که سهم هر ویژگی در میزان آنتروپی تام متفاوت بوده و لذا می‌بایست به هر یک از ویژگی‌ها یک مقدار وزن خاص را بنا به سهم آن در میزان بی‌نظمی نسبت دهیم. پس از وزن‌دار کردن ویژگی‌ها، مفهوم آنتروپی تام وزن‌دار را معرفی خواهیم نمود و به دنبال آن مدل بهینه‌سازی‌ای که پیش از این قید شد و البته ضریب داده‌ی پرت مبتنی بر مدل بهینه‌سازی مربوطه را به تفصیل شرح خواهیم داد.

### ۲.۶.۳ آنتروپی و همبستگی تام

مجموعه‌داده‌ی  $\mathbf{X}$  را با  $n$  عضو به صورت  $\{x_1, x_2, \dots, x_n\}$  در نظر می‌گیریم، به گونه‌ای که هر  $x_i$  به ازای  $1 \leq i \leq n$  یک بردار از ویژگی‌های نامی  $[y_1, y_2, \dots, y_m]^T$  می‌باشد، و هر  $y_j$  نیز دامنه‌ی مقادیر مشخصی دارد که به صورت  $[y_{1,j}, y_{2,j}, \dots, y_{n_j,j}]$  نشان داده می‌شود، به طوری که  $1 \leq j \leq m$  بوده و  $n_j$  نیز معرف تعداد مقادیر مشخص و مبین ویژگی  $y_j$  می‌باشد. بردار ویژگی  $[y_1, y_2, \dots, y_m]^T$  را می‌توان با  $\mathbf{Y}$  نشان داده و  $x_i$  نیز به صورت  $[x_{i,1}, x_{i,2}, \dots, x_{i,m}]^T$  نشان داده می‌شود. در اینجا از علائم  $C_X()$ ،  $I_X()$  و  $H_X()$ <sup>۱۴۲</sup> برای نمایش به ترتیب معیارهای آنتروپی، «اطلاعات دوطرفه»<sup>۱۴۳</sup> و همبستگی

<sup>۱۴۲</sup> Mutual information

تام روی مجموعه داده‌ی  $\mathbf{X}$  استفاده خواهیم کرد. از آن جا که مجموعه‌ی داده‌ی مورد بررسی در همه‌جای مسئله یکی است، لذا از قید زیرنویس  $\mathbf{X}$  در هر کدام از این فرمول‌ها خودداری می‌نمائیم.

فرمول آنتروپی روی کل مجموعه داده‌ی  $\mathbf{X}$  با مجموعه‌ی ویژگی‌های  $\mathbf{Y}$  بنا به «قانون زنجیره‌ای»<sup>۱۴۳</sup> به صورت زیر تعریف می‌شود:

$$\begin{aligned} H(\mathbf{Y}) &= H(y_1, y_2, \dots, y_m) = \sum_{i=1}^m H(y_i | y_{i-1}, \dots, y_1) \\ &= H(y_1) + H(y_2 | y_1) + \dots + H(y_m | y_{m-1}, \dots, y_1) \end{aligned} \quad (14.3)$$

به گونه‌ای که:

$$H(y_m | y_{m-1}, \dots, y_1) = - \sum_{y_m, y_{m-1}, \dots, y_1} p(y_m, y_{m-1}, \dots, y_1) \log_2 p(y_m | y_{m-1}, \dots, y_1) \quad (15.3)$$

در تئوری اطلاعات، معیار آنتروپی، معرف میزان عدم قطعیت با توجه به یک متغیر تصادفی خاص می‌باشد؛ به عبارتی اگر مقدار یک ویژگی نامعین باشد، مقدار آنتروپی این ویژگی بیانگر آن است که چه میزان اطلاعات نیاز است تا مقدار صحیح آن را پیش‌بینی نموده و به عبارتی تخمین بزنیم. در اینجا باید خاطرنشان کرد که خود معیار آنتروپی نیز می‌تواند به عنوان یک مقیاس سنجش سراسری جهت کشف داده‌های پرت مورد استفاده واقع شود. به گونه‌ای که اگر در یک مجموعه داده، تعدادی از داده‌های کاندید داده‌ی پرت را حذف نموده و مجددآ آنتروپی را روی کل مجموعه داده حساب نمائیم، این مقدار می‌بایست کاهش چشم‌گیری داشته باشد. هر چه این کاهش بیشتر باشد، احتمال پرت‌بودن آن داده‌های منتخب نیز به مراتب بیشتر خواهد بود. اما آزمایشات انجام شده نشان می‌دهند که معیار آنتروپی، به تنهاشی شاخص خوبی جهت کشف داده‌های پرت نمی‌باشد و معیار آنتروپی تام که در ادامه معرفی خواهد شد، به شکل مناسب‌تری عمل می‌نماید.

حال در اینجا معیار همبستگی تام را معرفی می‌نمائیم که از معیار اطلاعات دوطرفه روی کل مجموعه داده بهره می‌برد و در ادامه نشان می‌دهیم که این معیار نیز می‌تواند مانند آنتروپی جهت کشف داده‌های پرت مورد استفاده واقع شود. همبستگی تام برابر با مجموع اطلاعات دوطرفه‌ی مجموعه‌ی

<sup>143</sup> Chain rule

ویژگی  $Y$  می‌باشد، که در اینجا مجموعه‌ی  $Y$  در قالب یک سری بردارهای تصادفی گسسته نمایش داده می‌شود. داریم:

$$\begin{aligned} C(Y) &= \sum_{i=2}^m \sum_{\{r_1, \dots, r_i\} \subset \{1, \dots, m\}} I(y_{r_1}; \dots; y_{r_i}) \\ &= \sum_{\{r_1, r_2\} \subset \{1, \dots, m\}} I(y_{r_1}; y_{r_2}) + \dots + I(y_{r_1}; \dots; y_{r_m}) \end{aligned} \quad (16.3)$$

معیار همبستگی تام، میزان وابستگی دوطرفه یا همان اطلاعات به اشتراک‌گذاشته شده را روی کل مجموعه‌داده نشان می‌دهد. در اینجا لازم است بیان کنیم که هر چه همبستگی تام بین دو ویژگی (یا همان متغیر تصادفی) بیشتر باشد، نشان از آن دارد که تعداد زوج مرتبهای یکسان به ازای دو ویژگی کمتر بوده و به همان اندازه تعداد زوج مرتبهای متفاوت و به عبارتی یکتا نیز بیشتر می‌باشد. هر چه تعداد زوج مرتبهای یکتا بیشتر باشد، میزان بی‌نظمی (آنتروپی) نیز بیشتر خواهد بود. عکس این مسئله نیز برقرار می‌باشد. در نتیجه مشاهده می‌کیم که معیار همبستگی تام هم می‌تواند مانند معیار آنتروپی جهت کشف داده‌های پرت و به عبارتی تعیین میزان خوب‌بودن یک سری داده‌ی کاندید داده‌ی پرت به کار رود. در ادامه به معرفی معیار جدید آنتروپی تام می‌پردازیم که از هر دوی معیارهای آنتروپی و همبستگی تام استفاده می‌نماید.

### ۳.۶.۳ آنتروپی تام روی بردار تصادفی $Y$

از آن‌جا که هر کدام از معیارهای آنتروپی و همبستگی تام به تنها نمی‌توانند معیار مناسبی جهت کشف داده‌های پرت باشند، لذا ناچاریم تا از معیار مناسب‌تر و دقیق‌تری تحت عنوان آنتروپی تام بهره ببریم. اگر توزیع مقادیر ویژگی‌های یک مجموعه‌داده را داشته باشیم، بنا به «اثبات وatanabe»<sup>۱۴۴</sup> می‌توان رابطه‌ی میان آنتروپی و همبستگی تام را به صورت زیر بیان نمود:

$$C_X(Y) = \sum_{i=1}^m H_X(y_i) - H_X(Y) \quad (17.3)$$

<sup>144</sup> Watanabe's proof

با توجه به این فرمول مفهوم جدید آنتروپی تام را به صورت زیر تعریف می‌نمائیم که برابر با مجموع آنتروپی و همبستگی تام روی بردار تصادفی  $\mathbf{Y}$  بوده و می‌تواند به صورت مجموع آنتروپی‌ها روی تک‌تک ویژگی‌ها تعریف گردد:

$$\mathbf{HL}_X(\mathbf{Y}) = \mathbf{H}_X(\mathbf{Y}) + \mathbf{C}_X(\mathbf{Y}) = \sum_{i=1}^m \mathbf{H}_X(\mathbf{y}_i) \quad (18.3)$$

### ۴.۶.۳ وزن دار کردن ویژگی‌ها

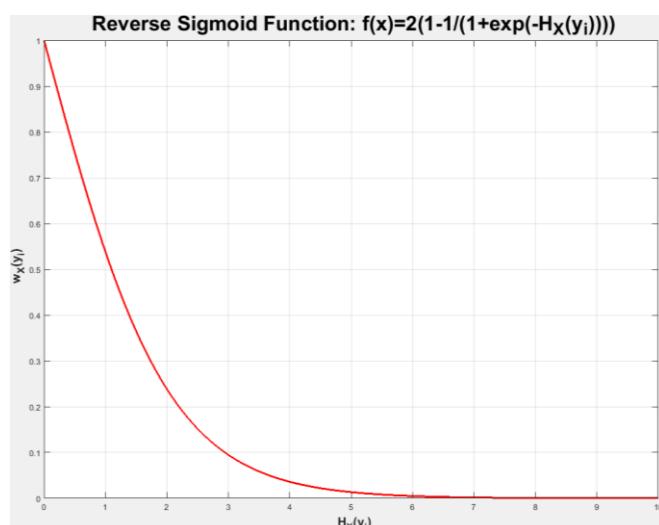
همان‌طور که از فرمول معیار آنتروپی تام قابل برداشت است، این معیار به همه‌ی ویژگی‌ها به یک اندازه اهمیت داده و ارزش همگی آن‌ها را در میزان پراکندگی و بی‌نظمی در کل مجموعه‌داده یکسان فرض می‌نماید. این در حالی است که در کاربردهای واقعی، هر ویژگی به یک اندازه‌ی خاص در شکل‌گیری ساختار کلی مجموعه‌داده نقش داشته و در نتیجه سهم آن در شدت آنتروپی کل متفاوت می‌باشد. حال با توجه به این‌که رویه‌ای که ما قصد پیروی از آن را جهت کشف داده‌های پرت در یک مجموعه‌داده داریم، آن است که آن دسته از داده‌ها که حذف آن‌ها سبب کاهش به مراتب بیشتر آنتروپی گردد را به عنوان کاندیدهای برتر داده‌ی پرت معرفی نمائیم، لذا می‌بایست به آن دسته از ویژگی‌ها که آنتروپی روی آن‌ها به تنها مقدار کمتری دارد وزن بیشتری اختصاص دهیم. علت این مسئله آن است که اگر یک ویژگی دارای مقادیر یکتای بیشتری نسبت به ویژگی دیگری باشد، آن‌گاه آنتروپی آن نیز به مراتب بیشتر خواهد بود. حال اگر یک داده‌ی کاندید پرت‌بودن را از ویژگی اول حذف نمائیم، می‌بینیم که میزان آنتروپی کاهش چشم‌گیری پس از حذف ندارد، زیرا که تعداد مقادیر یکتا در آن ویژگی هنوز زیاد است. اما در مورد ویژگی دوم خواهیم دید که در صورت حذف یکی از مقادیر یکتای موجود در آن ویژگی، میزان آنتروپی به نسبت ویژگی اول به مراتب بیشتر کاهش می‌یابد. چرا که تعداد مقادیر یکتا در آن ویژگی کم می‌باشد و در واقع این همان مقادیر یکتا می‌باشند که در هر مجموعه بیشترین سهم را در آنتروپی روی آن مجموعه دارند. از آن‌چه گفته شد می‌توان فهمید که آن دسته از ویژگی‌ها که آنتروپی کمتری دارند، بیشتر ما را در یافتن داده‌های پرت یاری کرده و به سبب آن می‌بایست به آن‌ها وزن بیشتری اختصاص دهیم. چرا که با این کار، در صورت حذف آن دسته از کاندیدهای داده‌ی پرت که در

آن ویژگی‌ها مقادیر یکتاتر و به اصطلاح برجسته‌تری دارند، میزان آنتروپی کاهش چشمگیرتری داشته و به دنبال آن مقصود ما که پیش از این به آن اشاره شد نیز برآورده می‌گردد.

اما برای وزن دار کردن هر ویژگی، در اینجا ما از یک تابع سیگموئید معکوس استفاده می‌کنیم که با توجه به مقتضیات مسئله به صورت زیر تعریف می‌شود:

$$w_X(y_i) = 2 \left( 1 - \frac{1}{1 + \exp(-H_X(y_i))} \right) \quad (19.3)$$

از آن‌جا که آنتروپی، همیشه مقداری بزرگتر یا مساوی صفر دارد، نمودار این تابع به صورت زیر خواهد بود:



شکل ۲.۳ نمودار تابع سیگموئید معکوس؛ از آن‌جا که آنتروپی همیشه مقادیر بزرگتر یا مساوی صفر دارد، لذا دامنه تابع محدود شده است. همان‌طور که مشاهده می‌شود، این تابع به مقادیر آنتروپی بیشتر وزن کمتری اختصاص داده و مقادیر وزن نیز همواره مابین صفر و یک خواهد بود

از نمودار تابع پیداست که کاملاً مطابق میل ما عمل کرده و هر چه مقدار آنتروپی بیشتر می‌شود، به آن وزن کمتری اختصاص می‌دهد. مقدار وزن نیز یک عدد مابین صفر و یک می‌باشد و هر چه مقادیر آنتروپی رو به بینهایت می‌رود، وزن‌های اختصاص داده شده به آن‌ها نیز بسیار نزدیک به هم خواهند بود. به عبارتی به ازای مقادیر آنتروپی نزدیک به صفر، میزان تفاوت در وزن اختصاص یافته چشمگیرتر خواهد بود تا به ازای مقادیر آنتروپی خیلی دورتر از صفر. در ادامه خواهیم دید که چگونه همین نکته‌ی مهم، ما را در مختصرسازی محاسبات سنگین یاری خواهد نمود.

### ۵.۶.۳ آنتروپی تام وزن دار روی بردar تصادفی Y

با توجه رویه‌ی وزن دار کردن ویژگی‌ها که به آن اشاره گردید، معیار جدید آنتروپی تام وزن دار روی بردar تصادفی  $Y$  را به صورت زیر و برابر مجموع آنتروپی‌های وزن دار روی تک‌تک ویژگی‌ها تعریف می‌نمائیم:

$$W_X(Y) = \sum_{i=1}^m w_X(y_i) H_X(y_i) \quad (20.3)$$

آزمایشات انجام شده نشان می‌دهند که نه تنها در مورد مجموعه‌داده‌های مصنوعی، بلکه در مورد مجموعه‌داده‌های واقعی نیز معیار آنتروپی تام وزن دار نسبت به نسخه‌ی بی‌وزن آن، ما را در کشف داده‌های پرت بهتر یاری نموده و سبب افزایش صحّت و سُقم عملیات می‌شود.

### ۶.۶.۴ یک تعریف رسمی از مسئله‌ی کشف داده‌های پرت

در اینجا قصد داریم تا یک توجیه مبرهن و رسمی را برای علت پرت‌بودن یک زیرمجموعه از داده‌ها با استفاده از آنتروپی تام وزن دار ارائه نمائیم. می‌گوئیم تعداد  $\mathbf{o}$  کاندید داده‌ی پرت، به عنوان بهترین زیرمجموعه معرفی خواهد شد، اگر حذف آن‌ها از مجموعه‌داده نسبت به حذف سایر زیرمجموعه‌های کاندید با همین اندازه، سبب بیشترین کاهش میزان آنتروپی تام وزن دار گردد. با توجه به آن‌چه گفته شد، ما با یک مسئله‌ی بهینه‌سازی روبرو هستیم که در آن می‌بایست به دنبال بهترین زیرمجموعه با اندازه‌ی  $\mathbf{o}$  باشیم که حذف آن سبب بیشترین کاهش در میزان آنتروپی تام وزن دار گردد. این مسئله‌ی بهینه‌سازی را به صورت زیر تعریف می‌نمائیم:

$$J_X(Y, \mathbf{o}) = W_{X \setminus Set(\mathbf{o})}(Y) \quad (21.3)$$

که در آن،تابع  $J$  برابر مقدار آنتروپی تام وزن دار مجموعه‌ی  $X$  پس از حذف  $\mathbf{o}$  تا از کاندیدهای داده‌ی پرت می‌باشد. (Nیز برابر هر زیرمجموعه‌ی ممکن با اندازه‌ی  $\mathbf{o}$  از اعضای مجموعه‌ی  $X$  می‌باشد. به عبارت بهتر می‌توان گفت که خروجی روش پیشنهادی در این روش به سادگی در قالب زیر قابل نمایش است:

$$Out(\mathbf{o}) = \operatorname{argmin} J_X(Y, \mathbf{o}) \quad (22.3)$$

اما از آن جا که هم پیدا کردن تمامی زیرمجموعه‌های ممکن با اندازه‌ی  $\mathbf{0}$  از مجموعه‌داده‌ی  $\mathbf{X}$  شدیداً به لحاظ محاسباتی دشوار می‌باشد و هم تعریف مقدار مناسب برای  $\mathbf{0}$  نیز امر ساده‌ای نخواهد بود (به طوری که حتی می‌تواند به عنوان یک مسیر جدید تحقیقاتی پیگیری شده و از همان خواص متغیر تابع بهینه‌سازی که مطرح شد بهره ببرد)، لذا ناچاریم تا به یک سری از الگوریتم‌های حریصانه جهت حل مسئله متولّش شویم. در ادامه نشان خواهیم داد که زمانی که تنها یکی از داده‌های کاندید پرتبودن از مجموعه‌داده حذف می‌گردد، می‌توان مقدار آنتروپی تام را به طرز بهینه‌ای به روزرسانی نمود و این مسئله در مورد حذف یک زیرمجموعه‌ی کاندید با اندازه‌ی بیشتر از یک به سادگی برقرار نخواهد بود. جالب آن است که در این به روزرسانی تنها به اطلاعات خود داده‌ای که حذف می‌گردد احتیاج بوده و نیازی به تخمین مجدد توزیع احتمالاتی کل مجموعه پس از حذف داده‌ی کاندید نمی‌باشد. علاوه بر این روشی را ارائه خواهیم نمود که با استفاده از آن می‌توان برای تعداد داده‌ی پرتی که کشف خواهند شد، یک حد بالا در نظر گرفته و به موجب آن فضای جستجو را کوچک‌تر خواهیم نمود تا مسئله‌ی بهینه‌سازی با سهولت بیشتری حل گردد. در ادامه نیز دو «الگوریتم حریصانه»<sup>۱۴۵</sup> با نام‌های ITB-SP و ITB-SS را معرفی خواهیم نمود که اولی به صورت یکباره و به عبارتی با یک حرکت، و دومی به صورت تدریجی و البته با دقت و صحت بیشتر، اقدام به کشف داده‌های پرت می‌نمایند.

### ۷.۶.۳ یک مفهوم جدید از ضریب داده‌ی پرت

در این جا برای اینکه بتوانیم برای هر داده یک مقدار امتیاز یا همان ضریب معرف میزان پرتبودن را تعریف نمائیم، می‌بایست ابتدا تابع بهینه‌سازی  $J$  را که پیش‌تر معرفی شد، تحلیل کنیم. از آن جا که بنای تابع بهینه‌سازی گفته شده، میزان تفاوت در آنتروپی تام وزن‌دار قبل و بعد از حذف زیرمجموعه‌ی کاندید می‌باشد، لذا می‌بایست توزیع احتمالاتی مجموعه‌ی  $\mathbf{Y}$  را پس از حذف زیرمجموعه‌ی مربوطه مجدداً محاسبه نمائیم که البته امر بسیار دشوار و طاقت‌فرسائی خصوصاً در مورد مجموعه‌داده‌های با مقیاس بزرگ می‌باشد. اما نکته‌ی جالب توجه آن است که می‌توان میزان تفاوت در آنتروپی تام وزن‌دار، قبل و بعد از حذف را تخمین زد. این مسئله زمانی که تنها یک داده از مجموعه‌داده حذف می‌گردد، بسیار ساده‌تر شده و حتی نیازی به تخمین توزیع‌های احتمالاتی ویژگی‌ها هم نخواهد بود، و درنتیجه این

<sup>145</sup> Greedy algorithms

موضوع می‌تواند یک «راه حل ابتکاری»<sup>۱۴۶</sup> جهت حل مسئله‌ی بهینه‌سازی (۲۱.۳) ارائه نماید. در ادامه به معرفی یک مفهوم جدید تحت عنوان «آنتروپی تام تفاضلی»<sup>۱۴۷</sup> می‌پردازیم که در نهایت راهکاری خواهد بود تا معیار ضریب داده‌ی پرت را به صورت رسمی تعریف نمائیم.

### ۱.۷.۶.۳ آنتروپی تام تفاضلی

اگر داده‌ی  $x_0$  را در نظر بگیریم، تفاوت آنتروپی تام وزن دار میان مجموعه‌داده‌ی  $\mathbf{X}$  و مجموعه‌داده‌ی  $\mathbf{X} \setminus \{x_0\}$  (همان مجموعه‌داده‌ی  $\mathbf{X}$  پس از حذف داده‌ی  $x_0$ ) را تحت عنوان آنتروپی تام تفاضلی معرفی کرده و با  $\mathbf{h}_X(x_0)$  به صورت زیر نشان می‌دهیم:

$$\begin{aligned}\mathbf{h}_X(x_0) &= \mathbf{W}_X(\mathbf{Y}) - \mathbf{W}_{X \setminus \{x_0\}}(\mathbf{Y}) \\ &= \sum_{i=1}^m [\mathbf{w}_X(\mathbf{y}_i) \mathbf{H}_X(\mathbf{y}_i) - \mathbf{w}_{X \setminus \{x_0\}}(\mathbf{y}_i) \mathbf{H}_{X \setminus \{x_0\}}(\mathbf{y}_i)]\end{aligned}\quad (۲۳.۳)$$

با توجه به نکته‌ای که در قسمت وزن دار کردن ویژگی‌ها به آن اشاره گردید، از آنجائی که وزن آنتروپی همیشه مقداری مابین صفر و یک دارد و البته به ازای مقادیر آنتروپی بزرگتر نیز، تفاوت میان وزن‌ها بسیار اندک و قابل چشم‌پوشی است، لذا می‌توان مقدار وزن را به ازای هر دوی  $(\mathbf{H}_X(\mathbf{y}_i)$  و  $\mathbf{H}_{X \setminus \{x_0\}}(\mathbf{y}_i)$ ، یکسان و برابر همان  $(\mathbf{y}_i)$  در نظر گرفت. بنابراین معادله‌ی ساده‌شده‌ی آنتروپی تام تفاضلی که در اینجا آن را آنتروپی تام تفاضلی تخمینی می‌نامیم، به صورت زیر خواهد بود:

$$\hat{\mathbf{h}}_X(x_0) = \sum_{i=1}^m \mathbf{w}_X(\mathbf{y}_i) [\mathbf{H}_X(\mathbf{y}_i) - \mathbf{H}_{X \setminus \{x_0\}}(\mathbf{y}_i)] \quad (۲۴.۳)$$

بنا به آزمایشات انجام شده، مشخص شده است که تفاوت میان آنتروپی تام تفاضلی اصلی و تخمینی بسیار اندک بوده و عملکرد آن‌ها شدیداً به یکدیگر شبیه می‌باشد، و به عبارتی ضریب داده‌ی پرت اصلی و تخمینی نیز که به دنبال آن حاصل می‌گردد، با یکدیگر تفاوت چندانی ندارند. طی یک سری محاسبات ریاضیاتی می‌توان نشان داد که می‌توان آنتروپی تام تفاضلی تخمینی را به طور مستقیم و به صورت زیر محاسبه نمود:

<sup>146</sup> Heuristic approach

<sup>147</sup> Differential Holoentropy

$$\begin{aligned}\hat{h}_X(x_o) &= \sum_{i=1}^m w_X(y_i) \left( \log_2 a - \frac{a}{b} \log_2 b \right) - a W_X(Y) \\ &\quad + a \sum_{i=1}^m \begin{cases} 0, & \text{if } n(x_{o,i}) = 1; \\ w_X(y_i) \cdot \delta[n(x_{o,i})], & \text{else.} \end{cases}\end{aligned}\tag{۲۵.۳}$$

به طوری که  $\delta[x] = (x - 1) \log_2(x - 1) - x \log_2 x$  بوده و  $x_{o,i}$  نیز معرف مقداری است که در ویژگی  $i$ -ام داده  $x_o$  ظاهر می‌گردد.  $n(x_{o,i})$  نیز معرف تعداد دفعاتی است که مقدار  $x_{o,i}$  در ویژگی  $i$ -ام ظاهر می‌گردد. مقادیر  $b$  و  $a$  نیز به ترتیب معکوس تعداد اعضای مجموعه‌های  $X$  و  $\{x_o\}$  می‌باشند، به عبارتی اگر تعداد اعضای مجموعه‌ی اصلی برابر  $n$  باشد، آن‌گاه  $b = 1/n$  و  $a = 1/(n - 1)$  خواهد بود.

فرمول (۱۰) در واقع راهکار ما جهت بهروزرسانی مقادیر آنتروبی و همین‌طور وزن‌های مربوطه در مراحل بعدی خواهد بود. نکته‌ی قابل توجه در مورد فرمول  $\hat{h}_X(x_o)$  یا همان مقدار آنتروبی تام تفاضلی به ازای داده  $x_o$ ، آن است که مقدار آن در دو جمله‌ی اول معادله‌ی (۲۵.۳) تنها به مجموعه‌داده‌ی  $X$  به تنها مشاهده می‌کنیم که جمله‌ی سوم معادله نیز تنها به خود داده  $x_o$  وابسته می‌باشد. با توجه به خاص و مختلف و البته در مراحل بهروزرسانی بعدی دیگر نیازی به محاسبه‌ی مجدد آن‌ها نخواهد بود؛ همین‌طور مشاهده می‌کنیم که جمله‌ی سوم معادله (۲۵.۳) به ازای هر کدام از داده‌های مجموعه، می‌توان آن را به عنوان معیار «ضریب داده‌ی پرت» به کار برد.

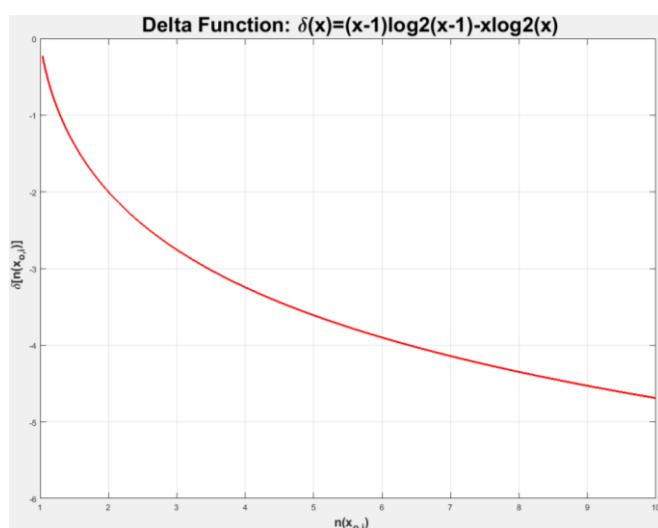
### ۲.۷.۶.۳ ضریب داده‌ی پرت

با توجه به آن‌چه که در قسمت قبل قید شد، معیار ضریب داده‌ی پرت به ازای داده  $x_o$  را به صورت زیر تعریف می‌نمائیم:

$$OF(x_o) = \sum_{i=1}^m OF(x_{o,i}) = \sum_{i=1}^m \begin{cases} 0, & \text{if } n(x_{o,i}) = 1; \\ w_X(y_i) \cdot \delta[n(x_{o,i})], & \text{else.} \end{cases}\tag{۲۶.۳}$$

به طوری که  $\mathbf{OF}(x_{0,i})$  برابر مقدار ضریب داده‌ی پرت برای داده‌ی  $x_0$  و به ازای ویژگی  $n^{\text{اُم}}$  می‌باشد. به عبارتی هر کدام از ویژگی‌ها به یک اندازه‌ی خاص در تعیین ضریب داده‌ی پرت برای یک داده نقش دارد.

ضریب داده‌ی پرت را می‌توان این‌گونه تفسیر نمود که چقدر احتمال دارد که یک داده‌ی خاص مورد بررسی، یک داده‌ی پرت باشد. هر چه این مقدار بیشتر باشد، احتمال پرت‌بودن داده‌ی مورد نظر نیز بیشتر خواهد بود. لازم به ذکر است که مقدار  $\mathbf{OF}(x_0)$  بنا به خاصیتتابع  $(\cdot)$  همواره مقداری کوچکتر یا مساوی صفر دارد. نمودار تابع  $(\cdot)$  به صورت زیر می‌باشد:



شکل ۳.۲ نمودار تابع  $\mathbf{δ}(x) = (x - 1) \log_2(x - 1) - x \log_2(x)$ : از آن‌جا که مقدار این تابع همواره منفی است، لذا مقدار ضریب داده‌ی پرت نیز همواره مقداری کوچکتر یا مساوی صفر دارد

از آن‌جا که هر ویژگی یک سری مقادیر ممکن را به خود می‌گیرد و فراوانی هر کدام از این مقادیر ممکن در کل مجموعه‌داده متفاوت می‌باشد، لذا برای آن دسته از داده‌هایی که مقادیر با فراوانی بالاتری از هر ویژگی را دارند، میزان  $\mathbf{OF}$  به مراتب کمتر می‌باشد و بالعکس. این مسئله را می‌توان با یک سری محاسبات ریاضیاتی روی خواص  $\mathbf{OF}$  نشان داد. به عبارت دیگر، اگر یک داده در ویژگی‌های خود دارای مقادیر یکتاًتر و خاص‌تری به ازای هر ویژگی نسبت به سایر داده‌ها باشد، آن‌گاه احتمال پرت‌بودن آن داده به مراتب بالاتر خواهد بود.

### ۸.۶.۳ به روزرسانی ضریب داده‌ی پرت

در اینجا قصد بررسی حالتی را داریم که پس از کشف یک داده‌ی پرت، می‌بایست آن را از مجموعه داده حذف کرده و سپس به دنبال داده‌ی پرت با اولویت بیشتر باشیم. کاملاً پیداست که پس از حذف یک داده، ساختار کلی مجموعه داده متحول شده و در نتیجه نیاز خواهد بود تا مجدداً توزیع احتمالاتی ویژگی‌ها را به دست آورده و میزان آنتروپی روی هر یک را محاسبه کنیم، و این مسئله شدیداً به لحاظ زمانی طاقت‌فرساست. لذا همان‌طور که در مورد آنتروپی تام تفاضلی توانستیم حجم محاسبات را کاهش دهیم، در اینجا نیز به همان شکل عمل کرده و مقدار آنتروپی تام تفاضلی بی‌وزن –  $HL_X(Y) - HL_{X \setminus \{x_o\}}(Y)$  را به صورت زیر بازنویسی می‌نمائیم:

$$\begin{aligned} & HL_X(Y) - HL_{X \setminus \{x_o\}}(Y) \\ &= m \left[ \left( \frac{a}{b} - a \right) \log_2 a - (b+1) \log_2 b \right] - b HL_X(Y) \\ &+ a \sum_{i=1}^m \begin{cases} 0, & \text{if } n(x_{o,i}) = 1; \\ \delta[n(x_{o,i})], & \text{else.} \end{cases} \end{aligned} \quad (27.3)$$

در نتیجه می‌توان فرمول ساده‌شده‌ی آنتروپی تام به روزشده را به صورت زیر بازنویسی نمائیم:

$$\begin{aligned} & HL_{X \setminus \{x_o\}}(Y) = (1+b) HL_X(Y) \\ & - m \left[ \left( \frac{a}{b} - a \right) \log_2 a - (b+1) \log_2 b \right] \\ & - a \sum_{i=1}^m \begin{cases} 0, & \text{if } n(x_{o,i}) = 1; \\ \delta[n(x_{o,i})], & \text{else.} \end{cases} \end{aligned} \quad (28.3)$$

با استفاده از (۲۸.۳) می‌توانیم مقدار آنتروپی به روزشده را به ازای تک‌تک ویژگی‌ها محاسبه نمائیم. داریم:

$$\begin{aligned} & H_{X \setminus \{x_o\}}(y_i) = (1+b) H_X(y_i) - \left[ \left( \frac{a}{b} - a \right) \log_2 a - (b+1) \log_2 b \right] \\ & - a \begin{cases} 0, & \text{if } n(x_{o,i}) = 1; \\ \delta[n(x_{o,i})], & \text{else.} \end{cases} \end{aligned} \quad (29.3)$$

پس از محاسبه‌ی مجدد آنتروپی به ازای هر کدام از ویژگی‌ها، می‌توانیم وزن مربوط به هر یک را نیز با استفاده از (۱۹.۳) مجدداً محاسبه نموده و در نهایت با استفاده از (۲۶.۳) ضریب داده‌ی پرت را به روزرسانی نمائیم.

### ۹.۶.۳ تعیین یک حد بالا برای تعداد داده‌های پرت

با توجه به این که در روش‌های یادگیری بدون نظارت، اکثریت داده‌ها نرمال فرض می‌شوند، لذا ناچاریم تا برای تعداد داده‌های غیرنرمال یا پرتی که در مجموعه داده حضور دارند، یک حد بالا تعیین نمائیم. در اینجا سه مفهوم جدید را بدین ترتیب معرفی می‌نماییم: حد بالای تعداد داده‌های پرت ( $\text{UO}^{148}$ ), مجموعه‌ی کاندید داده‌های پرت ( $\text{AS}^{149}$ )، و مجموعه داده‌های نرمال ( $\text{NS}^{150}$ ).

سه مفهوم جدید مطرح شده در بالا بنا به این دیدگاه حاصل شده‌اند که حذف داده‌های پرت از مجموعه داده سبب کاهش آنتروپی تام وزن دار ( $W_X(Y)$  و بیشتر خالص شدن کل مجموعه داده می‌شود. خلاف این مسئله در مورد داده‌های نرمال برقرار می‌باشد، بدین معنی که حذف آن‌ها سبب افزایش  $W_X(Y)$  خواهد شد. بنابراین می‌توان با استفاده از علامت آنتروپی تام تفاضلی ( $\hat{h}_X(x_0)$  به ازای هر داده‌ی  $x_0$ ، به ماهیت نرمال یا پرت بودن آن پی برد. در نتیجه داریم:

$$\begin{aligned} \text{NS} &= \{x_i, \hat{h}(x_i) \leq \mathbf{0}\}, \\ \text{AS} &= \{x_i, \hat{h}(x_i) > \mathbf{0}\}, \\ \text{UO} &= N(\text{AS}) = \sum_{i=1}^n (\hat{h}(x_i) > \mathbf{0}) \end{aligned} \quad (30.3)$$

در اینجا باید خاطرنشان کرد که حداکثر تعداد داده‌های پرتی که توسط الگوریتم‌های پیشنهادی در این گزارش قابل کشف شدن می‌باشد، برابر تعداد اعضای مجموعه  $\text{AS}$  و برابر  $\text{UO}$  می‌باشد. حتی در حالی که قصد پیدا کردن داده‌های پرت را به صورت مرحله به مرحله داریم، باز هم فضای جستجو همان مجموعه‌ی  $\text{AS}$  خواهد بود و این مسئله قابل اثبات است که پس از حذف یک داده‌ی پرت از مجموعه داده و به تبع آن در هم ریخته شدن نظم سراسری مجموعه داده، هیچ کدام از داده‌های نرمال مجموعه، از حالت نرمال خارج نشده و اصطلاحاً مشکوک به پرت بودن نخواهند شد.

<sup>148</sup> Upper Bound on Outliers

<sup>149</sup> Anomaly Candidate Set

<sup>150</sup> Normal Object Set

### ۱۰.۶.۳ معرفی الگوریتم‌های ITB-SS و ITB-SP

در اینجا قصد داریم تا با توجه به ماهیت ضریب داده‌ی پرت که پیش از این به آن اشاره گردید، دو الگوریتم حریصانه را جهت کشف داده‌های پرت در مجموعه داده‌های با ویژگی‌های نامی استخراج نمائیم.

اولین الگوریتم **ITB-SP (Information-Theory-Based Single-Pass)** نام دارد که در آن مقدار ضریب داده‌ی پرت به ازای تمامی داده‌ها تنها یک بار محاسبه گشته و سپس تعداد **0** داده‌ی پرت مورد درخواست کاربر با بالاترین میزان **OF** به عنوان خروجی ارائه می‌گردد. دومین الگوریتم نیز **ITB-SS** (Information-Theory-Based Step-by-Step) نام دارد که در یک رویه‌ی گام‌به‌گام اقدام به کشف داده‌های پرت می‌نماید. به این ترتیب که ابتدا با استفاده از مقدار آنتروپی تام تفاضلی  $(x_0 \hat{h}_X)$  به ازای هر داده‌ی  $x_0$ ، مجموعه‌ی کاندید داده‌ی پرت یا همان **AS** را پیدا نموده و سپس داده‌های از این مجموعه که بیشترین مقدار **OF** را دارد، به عنوان اولین داده‌ی پرت معرفی می‌نماییم. سپس داده‌ی مربوطه را از مجموعه‌ی **AS** حذف نموده و مقدار **OF** را به ازای تمامی داده‌های باقیمانده‌ی **AS** به روزرسانی می‌کنیم و همین رویه را آنقدر تکرار خواهیم کرد تا داده‌های پرت به تعداد درخواستی کاربر کشف گردند. لازم به ذکر است که هر دوی این الگوریتم‌ها، داده‌های پرت را درون مجموعه‌ی **AS** جستجو می‌کنند و به عبارتی فضای جستجو همواره محدود به مجموعه‌ی **AS** خواهد بود. این مسئله در مورد **ITB-SP** چندان تفاوتی نمی‌کند، زیرا که این الگوریتم، داده‌های پرت را در همان اولین مرحله و پس مرتب‌سازی ضرایب داده‌ی پرت پیدا می‌کند. اما در مورد **ITB-SS** این مسئله متفاوت‌تر می‌باشد، زیرا پس از هر مرحله کشف، می‌بایست یک سری محاسبات مجدداً انجام شود، اما با این حال اثبات می‌شود که فضای جستجو باز هم محدود به همان **AS** خواهد بود.

فرض ما در این روش آن است که کاربر مربوطه تعداد داده‌های پرت درخواستی خویش را ارائه می‌دهد و این تعداد که با **0** نشان داده می‌شود، همواره از **UO** یا همان حد بالای تعداد داده‌های پرت کمتر خواهد بود. ولی در صورت بیشتر بودن هم تنها با یک تغییر جزئی می‌توان این تعداد درخواستی را به همان اندازه‌ی **UO** محدود نمود. اما نکته‌ی قابل توجه آن است که همواره تعداد معقول و منطقی داده‌های پرت بسیار کمتر از حد **UO** می‌باشد و به عبارتی این حد بالا، حد غایی داده‌های پرت ممکن موجود در مجموعه داده می‌باشد.

در اینجا الگوریتم **ITB-SP** را به صورت زیر ارائه می‌نماییم:

***ITB-SP Single Pass Algorithm***

```

1: Input: dataset X and number of outliers requested o
2: output: outlier set OS
3: Compute  $w_X(y_i)$  for ( $1 \leq i \leq m$ ) by (3-2)
4: Set OS =  $\emptyset$ 
5: for  $i = 1$  to  $n$  do
6:   Compute OF( $x_i$ ) and obtain AS
7: end for
8: if  $o > UO$  then
9:    $o = UO$ 
10: else
11:   Build OS by searching for the  $o$  objects with greatest
      OF( $x_i$ ) in AS using heapsort
12: end if

```

لازم به ذکر است که پیچیدگی زمانی الگوریتم ITB-SP برابر با  $O(nm)$  میباشد که در آن  $n$  برابر تعداد داده‌های مجموعه‌داده و  $m$  نیز برابر تعداد ویژگی‌ها میباشد.

در اینجا هم الگوریتم **ITB-SS** را به قرار زیر ارائه می‌دهیم:

***ITB-SS Step-by-Step Algorithm***

```

1: Input: dataset X and number of outliers requested o
2: output: outlier set OS
3: Set OS =  $\emptyset$ 
4: Compute  $w_X(y_i)$  for ( $1 \leq i \leq m$ ) by (3-2)
5: for  $i = 1$  to  $n$  do
6:   Compute OF( $x_i$ ) and obtain AS
7: end for
8: if  $o > UO$  then
9:    $o = UO$ 
10: else
11:   for  $i = 1$  to  $o$  do
12:     Search for the object with greatest OF( $x_o$ ) from AS
13:     Add  $x_o$  to OS and remove it from AS
14:     Update all the OF(x) of AS
15:   end for
16: end if

```

پیچیدگی زمانی الگوریتم **ITB-SS** نیز برابر با  $O(om^*(UO))$  میباشد، که معمولاً کمی بیشتر از پیچیدگی زمانی الگوریتم اول یعنی **ITB-SP** بوده و علت آن نیز انجام مرحله‌به‌مرحله‌ی کشف داده‌های پرت میباشد. اما آزمایشات انجام شده نشان از آن دارند که این مقدار اختلاف در زمان محاسبات، ارزش دقت بالاتر در کشف داده‌های پرت را دارد.

## ۷.۳ کشف داده‌های پرت محلی در داده‌های با مقیاس بزرگ با استفاده

### از یک روش کاهش بُعد در ضمن حفظ چگالی داده‌ها

«وریس»<sup>۱۵۱</sup> و همکاران [۱۸]، روشی را جهت کشف داده‌های پرت محلی در مجموعه داده‌های با مقادیر ویژگی عددی، با مقیاس و ابعاد وسیع ارائه نموده‌اند. بنای رویکرد مورد استفاده در این روش، همان LOF می‌باشد که پیش از این به تفصیل شرح داده شد. این روش پایه یعنی LOF، در مورد مجموعه داده‌های با مقیاس و ابعاد نرمال از کارائی لازم برخوردار می‌باشد، اما با افزایش مقیاس و ابعاد داده‌ها کارائی خود را از دست داده و یا به لحاظ زمانی و محاسباتی استفاده از آن به صرفه نخواهد بود. روش پیشنهادی در این روش، ابتدا داده‌های مجموعه را با استفاده از یک روش کاهش بُعد معتبر با نام «تصویرسازی تصادفی»<sup>۱۵۲</sup> (که در ادامه از آن به اختصار به صورت RP یاد می‌شود)، به فضای با ابعاد کمتری می‌برد، و سپس در این فضای جدید به دنبال نزدیک‌ترین همسایگان هر داده ولی با یک دامنه‌ی بیشتر می‌گردد. این دامنه‌ی بیشتر تضمین می‌نماید تا تخمین دقیق‌تری از نزدیک‌ترین همسایگان هر داده داشته باشیم، چرا که در ضمن عمل کاهش بُعد، ساختار درونی داده‌ها از جمله میزان مقاربت میان آن‌ها دچار تزلزل شده و در نتیجه عملیات ما دارای دقت کامل نخواهد بود. رویکرد مورد استفاده در این روش با نام «نزدیک‌ترین همسایگان حاصله از شاخص تصویرسازی»<sup>۱۵۳</sup> یا به اختصار PINN شناخته می‌شود. لازم به ذکر است که پس از حصول نزدیک‌ترین همسایگان واقعی هر داده، نوبت به استفاده از روش پایه می‌رسد تا در نهایت به هر داده، یک امتیاز LOF مبنی بر میزان پرت محلی بودن نسبت داده شود. نتایج آزمایشات انجام‌شده حاکی از آن است که PINN قادر است تا توزیع چگالی درونی مجموعه داده را پس از تصویرسازی یا همان کاهش بُعد، حفظ نموده و البته بر روش‌های پایه‌ی کاهش بُعد، مانند RP و البته روش محبوب «تحلیل مؤلفه‌ی اصلی»<sup>۱۵۴</sup> یا همان PCA، به هنگام محاسبه‌ی معیار LOF برتری یابد. در ادامه، در قسمت شرح روش و پارامترها به تفصیل در مورد جزئیات این روش صحبت خواهیم نمود.

<sup>151</sup> Timothy de Vries

<sup>152</sup> Random Projection (RP)

<sup>153</sup> Projection-Indexed Nearest-Neighbors (PINN)

<sup>154</sup> Principle Component Analysis (PCA)

### ۱.۷.۳ شرح روش و پارامترها

در اینجا لازم است تا پیش از پرداختن به اصل رویکرد پیشنهادی در این روش، چند مسئله‌ی پایه را به صورت مختصر توضیح دهیم. این مسائل شامل دو روش کاهش بُعد PCA و RP می‌باشد. رویه‌ی LOF نیز پیش از این توضیح داده شد. بنای PINN بر همین مفاهیم پایه می‌باشد.

### ۲.۷.۳ تحلیل مؤلفه‌ی اصلی (PCA)

از جمله محبوب‌ترین روش‌هایی که جهت آماده‌سازی مجموعه‌داده‌های با ابعاد زیاد برای بازنمایی و پردازش به کار می‌روند، روش‌های کاهش بعد می‌باشند که در آن‌ها، داده‌ها را به فضای با ابعاد کمتر انتقال داده و محاسبات را با یک درصد خطای نسبی انجام می‌دهیم. به عنوان مثال، اگر بخواهیم مجموعه‌داده‌ی  $X$  با  $n$  داده و  $m$  بُعد یا ویژگی را به فضای با  $t$  بُعد ( $t < m$ ) ببریم، می‌توانیم از یک ماتریس تبدیل  $R$  با ابعاد  $m^*t$  استفاده کرده و ماتریس داده‌های خروجی به صورت  $XR = Y$  و با ابعاد  $n^*t$  خواهد بود. به عبارت دیگر، به ازای هر داده‌ی  $x \in X$ ، تصویر آن در فضای جدید به صورت  $x' = xR$  می‌باشد.

آن‌چه گفته شد، خلاصه‌ای از رویه‌ی کلی روش‌های کاهش بُعد یا همان روش‌های «تصویرسازی داده‌ها»<sup>۱۵۵</sup> از یک فضای با ابعاد زیاد به یک زیرفضا با ابعاد کمتر و با استفاده از ماتریس تبدیل  $R$  بود. البته که از محبوب‌ترین و پرکاربردترین این روش‌ها، روش خوش‌نم تحلیل مؤلفه‌ی اصلی یا همان PCA می‌باشد. در این روش جهت انتقال داده‌ها از فضای  $m$ -بُعدی به زیرفضای  $t$ -بُعدی، می‌بایست ابتدا تعداد  $t$  «بردار ویژه‌ی»<sup>۱۵۶</sup> ماتریس کوواریانس  $m^*m$  داده‌ها را مطابق با تعداد  $t$  بزرگترین «مقادیر ویژه‌ی»<sup>۱۵۷</sup> ماتریس مربوطه محاسبه نمائیم تا ماتریس  $R$  را به دست آوریم. اما مسئله آن است که در مورد داده‌های با ابعاد خیلی زیاد، محاسبه‌ی این تعداد بردار ویژه از مرتبه‌ی زمانی بالائی برخوردار بوده و به همین

<sup>155</sup> Data Projection

<sup>156</sup> EigenVector

<sup>157</sup> EigenValue

سبب استفاده از آن در مورد این نوع داده‌ها به صرفه نبوده و می‌بایست از روش‌های کاهش بُعد کاراتر و بهینه‌تری مانند RP که در ادامه معرفی خواهد شد استفاده نمائیم.

### ۳.۷.۳ تصویرسازی تصادفی (RP)

با توجه به هزینه‌ی محاسباتی بالای PCA در مورد مجموعه‌داده‌های با ابعاد زیاد، به ناچار می‌بایست به روش‌های پایه‌ی جایگزینی جهت کاهش بعد و تصویرسازی داده‌ها روی آوریم. از جمله‌ی این روش‌ها، روش تصویرسازی تصادفی یا RP می‌باشد که در آن درایه‌های ماتریس تبدیل را به صورت تصادفی و مستقل از یکدیگر و با توجه به میزان «تنکبودن»<sup>۱۵۸</sup> ماتریس اصلی داده‌ها تولید می‌نمائیم. اثبات می‌شود که تحت چنین تبدیلی، «فاصله‌ی اقلیدسی»<sup>۱۵۹</sup> میان دوبه‌دوی داده‌ها به صورت تخمینی و با دقیق بالاتر حفظ می‌شود. علاوه بر این، تعداد ابعاد در فضای جدید به صورت لگاریتمی به تعداد داده‌های مجموعه‌داده وابسته بوده و مستقل از تعداد ابعاد در فضای اصلی می‌باشد. درایه‌های ماتریس R را به صورت تصادفی و مستقل از یکدیگر به صورت زیر تولید می‌نمائیم:

$$r_{ij} = \sqrt{s} \begin{cases} +1 & \text{with probability } \frac{1}{2s} \\ 0 & \text{with probability } 1 - \frac{1}{s} \\ -1 & \text{with probability } \frac{1}{2s} \end{cases} \quad (31.3)$$

پارامتر  $s$ ، معرف میزان تنکبودن ماتریس داده‌های اصلی می‌باشد، که سبب می‌شود تا RP به صورت تخمینی به اندازه‌ی  $\frac{1}{s}$  از فضای ویژگی اصلی را به ازای هر کدام از ویژگی‌های زیرفضای جدید نمونه‌برداری نماید. به عبارت دیگر، تعداد درایه‌های غیر صفر ماتریس R برابر با  $\frac{1}{s}$  کل درایه‌های آن می‌باشد. لذا هرچه میزان تنکبودن یا همان اندازه‌ی  $s$  بیشتر بیشتر باشد، مقدار کسر  $\frac{1}{s}$  نیز به مراتب کمتر بوده و به تبع آن میزان نمونه‌برداری از فضای ویژگی اصلی نیز کمتر خواهد بود و بالعکس. می‌توان اثبات نمود که به ازای یک مقدار  $0 < \gamma$ ، تعداد ابعاد در فضای جدید که با  $t$  نمایش داده می‌شود، در نامعادله‌ی زیر صدق می‌نماید:

<sup>158</sup> Sparsity

<sup>159</sup> Euclidean distance

$$t \geq \frac{4 + 2\gamma}{\frac{\epsilon^2}{2} - \frac{\epsilon^3}{3}} \ln n \quad (32.3)$$

که به تبع آن می‌توان گفت که با احتمال حداقل  $n^{-\gamma} - 1$ ، ماتریس کاهش بعدیافته‌ی  $Y$ ، فاصله‌ی اقلیدسی میان دوبه‌دوی داده‌ها را به صورت تخمینی و با دقت بالائی حفظ می‌نماید. به عبارت دیگر، می‌توان این‌گونه بیان نمود که به ازای هر دو داده‌ی  $u$  و  $v$  در مجموعه‌داده‌ی اصلی، که تصویر آن‌ها در فضای جدید را با  $u'$  و  $v'$  نمایش می‌دهیم، داریم:

$$(1 - \epsilon).d(u, v) \leq d(u', v') \leq (1 + \epsilon).d(u, v) \quad (33.3)$$

به طوری که  $d(u, v)$  معرف فاصله‌ی میان  $u$  و  $v$  در فضای اصلی، و  $d(u', v')$  معرف فاصله‌ی میان همین داده‌ها در فضای کاهش بعدیافته می‌باشد. می‌توان از نامعادله‌ی (32.3) این‌گونه برداشت نمود که هر چه اندازه‌ی  $\gamma$  بیشتر باشد، صورت کسر سمت راست نیز به مراتب بیشتر شده و به تبع آن اندازه‌ی  $t$  یا همان اندازه‌ی ابعاد در فضای جدید نیز بزرگتر خواهد بود، و به ازای این مقدار  $t$ ، نامعادله‌ی (33.3) برقرار خواهد بود. قطعاً این یک امر طبیعی می‌باشد که هرچه اندازه‌ی ابعاد در فضای جدید بیشتر و به عبارت دیگر میزان کاهش بعد کمتر باشد، حجم اطلاعات کمتری از دست رفته و تخمین ما به مراتب دقیق‌تر خواهد بود.

#### ۴.۷.۳ تصویرسازی تصادفی ضمن حفظ توزیع چگالی درونی داده‌ها

با توجه به آن‌چه پیش از این قید شد، اگرچه که معیار بنیادی LOF، یک معیار مناسب و خوش‌تعریف جهت کشف داده‌های پرت محلی می‌باشد، اما از آن‌جا که بنای آن بر یافتن  $k$ -نزدیک‌ترین همسایه‌ی هر داده استوار می‌باشد، در مورد مجموعه‌داده‌های با مقیاس و ابعاد زیاد، این معیار کارائی خود را از دست می‌دهد. چرا که از یک سو، برای یافتن  $k$ -نزدیک‌ترین همسایگان یک داده می‌بایست فاصله‌ی آن داده را از تمامی مابقی مجموعه‌داده محاسبه نمود که این مسئله در مورد مجموعه‌داده‌های با مقیاس و ابعاد بالا بسیار پرهزینه خواهد بود. از سوی دیگر، در ابعاد بالاتر نیز احتمال این‌که دوبه‌دوی داده‌ها از یکدیگر فاصله‌ی یکسانی داشته باشند زیاد شده و موجب می‌شود تا نتوان نزدیک‌ترین همسایگان واقعی هر داده را پیدا کنیم.

یکی از راه حل های معمول برای این مسئله، کاهش بعد داده ها به گونه ای می باشد که فواصل میان دوبه دوی داده ها پس از تصویرسازی به صورت تقریبی حفظ گردد. چرا که در برخی از روش های کاهش بُعد، این مهم برآورده نشده و به دنبال آن، نتایج نهائی نیز قابل اتكاء نخواهند بود. اما در مورد روش تصویرسازی تصادفی، می توان نشان داد که نه تنها فاصله های میان دوبه دوی داده ها با تقریب بالائی حفظ می گردد، بلکه فاصله های هر داده از  $k$ -نزدیک ترین همسایه های آن نیز با دقت بالائی محفوظ باقی خواهد ماند. حال نکته ای قابل توجه در این مورد خاص یعنی تصویرسازی تصادفی آن است که اگرچه فاصله های یک داده تا  $k$ -نزدیک ترین همسایگان آن با تقریب بالائی حفظ می گردد، اما مجموعه های  $k$ -همسايگي يك داده ممکن است تغیير نماید. علت آن نیز می تواند آن باشد که داده هایی که بر روی شعاع همسایگی  $k$ -همسايگي دچار تغیير خواهد شد. این مسئله زمانی حادثه خواهد شد که بخواهیم در فضای کاهش بعدیافتہ برای داده های مربوطه، امتیاز LOF را محاسبه نمائیم، چرا که در آن صورت ممکن است یک همسایه های نامرتبط و نامناسب در مجموعه های همسایگی یک داده قرار گرفته و به تبع آن مقدار امتیاز LOF برای آن داده شدیداً متحوّل گردد.

با توجه به آن چه گفته شد، می توان این گونه بیان نمود که مجموعه های  $k$ -همسايگي هر داده در فضای اصلی، در واقع زیرمجموعه های از مجموعه های  $h$ -همسايگی همان داده در فضای کاهش بعدیافتہ خواهد بود، اگر مقدار  $h$  را با توجه برخی خواص ذاتی مجموعه داده که به آن «بعدیت درونی»<sup>۱۶۰</sup> گویند، به حد کافی بزرگتر از  $k$  انتخاب کنیم.

### ۱,۴,۷,۳ محفوظ ماندن فاصله های یک داده تا $k$ -نزدیک ترین همسایه های آن تحت تصویرسازی تصادفی

در این قسمت قصد داریم تا بدون توجه به این که احتمال دارد که در اثر تصویرسازی داده ها یا همان رویه های کاهش بُعد، مجموعه های همسایگی هر داده دچار تحول شود، نشان دهیم که فاصله های هر داده تا  $k$ -نزدیک ترین همسایه های آن در اثر این تصویرسازی به صورت تخمینی و با دقت بالائی محفوظ باقی می ماند. حال اگر تصویر نقطه هی  $p$  را در فضای جدید با  $p'$  نشان دهیم و مجموعه های  $k$ -همسايگي هر

<sup>160</sup> Intrinsic Dimensionality

کدام را نیز به ترتیب با  $(p) N_k$  و  $(p') N_k$  مشخص نمائیم، می‌توانیم مدعی شویم که فاصله‌ی هر دو داده‌ی دلخواه در فضای کاهش‌بعدیافت، به صورت زیر محدود به ضرایبی از فاصله‌ی آن دو در فضای اصلی می‌باشد:

$$(1 - \epsilon) \cdot d(x, y) \leq d(x', y') \leq (1 + \epsilon) \cdot d(x, y) \quad (34.3)$$

همین‌طور می‌توانیم بیان نمائیم که فاصله‌ی هر داده تا  $k$ -امین نزدیک‌ترین همسایه‌ی آن در فضای جدید و به ازای هر مقدار دلخواه  $k$  نیز به صورت زیر، محدود به ضرایبی از همین مقدار در فضای اصلی و به صورت زیر می‌باشد:

$$(1 - \epsilon) \cdot d_k(p) \leq d_k(p') \leq (1 + \epsilon) \cdot d_k(p) \quad (35.3)$$

به تبع این دو نامعادله، می‌توانیم مدعی شویم که با احتمال  $n^{-\gamma} - 1$  (به طوری که  $n$  برابر تعداد داده‌های مجموعه و  $\gamma$  نیز یک عدد دلخواه بزرگ‌تر از صفر می‌باشد)، به ازای هر داده‌ی  $p$  که تصویر آن در فضای جدید را با  $p'$  نمایش می‌دهیم، «چگالی نسبی»<sup>۱۶۱</sup> آن در فضای جدید محدود به بازه‌ی زیر می‌باشد:

$$\frac{1}{1 + \epsilon} \cdot rd(p) \leq rd(p') \leq \frac{1}{1 - \epsilon} \cdot rd(p) \quad (36.3)$$

حال با توجه به نامعادله‌های (34.3) تا (36.3)، می‌توانیم نشان دهیم که مجموعه‌ی  $k$ -همسایگی هر داده در فضای اصلی، در واقع زیرمجموعه‌ای از تصویر معکوس مجموعه‌ی  $h$ -همسایگی تصویر آن داده در فضای جدید می‌باشد، به شرط این‌که  $h$  را به حد کافی بزرگ‌تر از  $k$  انتخاب کنیم و این مسئله خود وابسته به خصیصه‌های ذاتی هر مجموعه‌داده می‌باشد.

## ۲.۴.۷.۳ محفوظماندن مجموعه‌های همسایگی هر داده در فضای کاهش‌بعدیافت با توجه به یک سری شرایط خاص

می‌توان نشان داد که با توجه به یک سری از خواص ذاتی هر مجموعه‌داده که از آن‌ها با نام بعديت درونی یاد می‌شود، مجموعه‌ی همسایگی هر داده در فضای اصلی، در واقع زیرمجموعه‌ای از مجموعه‌ی

<sup>161</sup> Reachability density (rd)

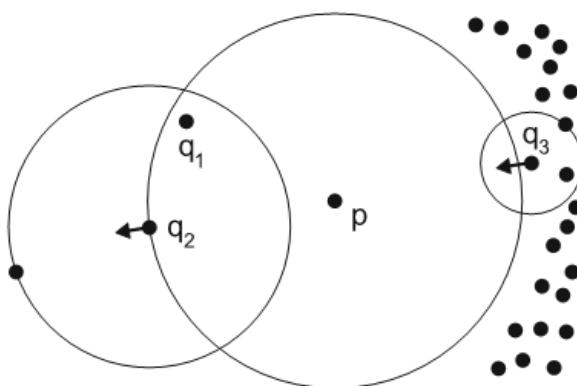
همسایگی بزرگتر آن داده در فضای کاهش بُعدیافته می‌باشد. اگر مجموعه‌ی  $h$ -همسایگی تصویر یک داده را در فضای جدید با  $(p')_{N_h}$  نشان دهیم، اعضای این مجموعه در واقع تصویر یک زیرمجموعه از داده‌ها در فضای اصلی و به صورت زیر می‌باشد:

$$RP^{-1}(N_h(p')) = \{x \in D | RP(x) \in N_h(p')\} \quad (37.3)$$

می‌توان اثبات نمود که مجموعه‌ی  $k$ -همسایگی هر داده‌ی  $p$  که با  $(p)_{N_k}$  نشان داده می‌شود، با احتمال  $n^{-1} - 1$  زیرمجموعه‌ای از  $RP^{-1}(N_h(p'))$  یا همان تصویر معکوس مجموعه‌ی  $h$ -همسایگی  $p'$  می‌باشد. هر چه مقدار  $\gamma$  بیشتر باشد، حد این احتمال نیز بالاتر خواهد بود. لازم به ذکر است که بالاتر بودن مقدار  $\gamma$ ، با توجه به (32.3) منتج به بالاتر بودن مقدار  $t$  یا همان اندازه‌ی ابعاد در فضای جدید خواهد شد، و این مسئله طبیعی است که هر چه کاهش بعد کمتری داشته باشیم، به تبع آن ساختار درونی مجموعه‌داده کمتر دچار تزلزل شده و در نتیجه روابط و فواصل میان داده‌ها از جمله مجموعه‌های همسایگی با احتمال بالاتری محفوظ باقی می‌مانند.

### ۵.۷.۳ نزدیک‌ترین همسایگان حاصله از شاخص تصویرسازی (PINN)

همان‌طور که پیش‌تر نیز قید شد، پس از عمل تصویرسازی تصادفی فاصله‌ی هر داده از  $k$ -نزدیک‌ترین همسایگان آن داده با تقریب بالائی حفظ می‌گردد، اما این مسئله در مورد مجموعه‌ی  $k$ -همسایگی هر داده به صورت مطلق صحت ندارد. به این معنی که ممکن است تصویر معکوس اعضای مجموعه‌ی  $k$ -همسایگی تصویر یک داده در فضای جدید، با مجموعه‌ی  $k$ -همسایگی همان داده در فضای اصلی متمانز باشد. این مسئله هنگامی که قصد داریم تا مقدار امتیاز LOF را به ازای هر داده محاسبه نمائیم، شدیداً تأثیرگذار خواهد بود، چرا که مقدار این امتیاز برای هر داده مبتنی بر چگالی‌های نسبی همسایگان آن داده بوده و همان‌طور که بیان شد این مجموعه‌های همسایگی پس از تصویرسازی تصادفی پایدار نخواهند بود. به همین دلیل قطعاً محاسبه‌ی امتیاز LOF به ازای هر داده در فضای تصویرشده، به دلیل تغییرات تأثیرگذار در مجموعه‌های همسایگی منطقی نخواهد بود. شکل ۴.۳، یک مثال از حالتی است که تغییر در مجموعه‌ی همسایگی یک داده پس از تصویرسازی تصادفی، می‌تواند شدیداً در مقدار امتیاز LOF برای آن داده تأثیرگذار باشد.



شکل ۴.۳ یک نمونه از تغییر در مجموعه‌ی همسایگی پس از تصویرسازی تصادفی به ازای  $k=2$ : به عبارتی ممکن است پس از تصویرسازی تصادفی، نقطه‌ی  $q_2$  از شعاع همسایگی نقطه‌ی  $p$  خارج شده و برعکس نقطه‌ی  $q_3$  در شعاع همسایگی آن قرار گیرد. همان‌طور که پیداست چگالی نسبی  $q_2$  از  $q_3$  به مراتب کمتر می‌باشد و در نتیجه پس از تصویرسازی تصادفی به دلیل تغییر در مجموعه‌ی  $k$ -همسایگی نقطه‌ی  $p$ , چگالی نسبی  $p$  تغییر جدی خواهد داشت که این مسئله در مقدار امتیاز LOF برای این نقطه شدیداً تأثیرگذار خواهد بود. در نتیجه، محاسبه‌ی امتیاز LOF در فضای تصویرشده منطقی نخواهد بود [۱۸].

در اینجا می‌توانیم به جای این که مقدار LOF را به ازای نقطه‌ی  $p$ ، در فضای جدید تخمین زده و متحمل خطاهای ممکن شویم، در ابتدا پس از تصویرسازی، مجموعه‌ی همسایگی بزرگتری را به ازای هر نقطه در نظر گرفته و سپس این مجموعه‌ی بزرگتر را با یک نگاشت معکوس به فضای اصلی منتقل نمائیم. سپس در فضای اصلی به دنبال مجموعه‌ی  $k$ -همسایگی واقعی نقطه‌ی  $p$  باشیم (همان‌طور که پیش از این نیز قید شد، مجموعه‌ی  $k$ -همسایگی  $p$  در فضای اصلی با احتمال بسیار بالائی در واقع زیرمجموعه‌ای از مجموعه‌ی  $h$ -همسایگی  $p'$  در فضای تصویرشده است، به این شرط که  $h$  را به حد کافی بزرگتر از  $k$  انتخاب نمائیم). پس از حصول  $k$ -نزدیک‌ترین همسایه‌ی نقطه‌ی  $p$  از میان مجموعه‌ی  $(RP^{-1}(N_h(p'))$  در فضای اصلی که آن را با  $\bar{N}_k(p)$  نمایش می‌دهیم، می‌توانیم مقدار  $LOF(p)$  را به صورت زیر و با دقت بالائی تخمین بزنیم:

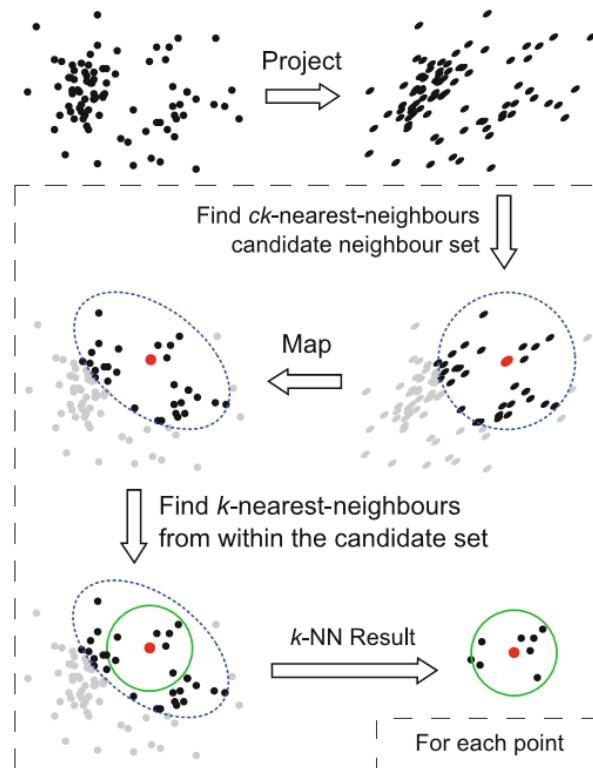
$$\overline{LOF}(p) = \frac{\frac{1}{k} \sum_{q \in \bar{N}_k(p)} rd(q')}{rd(p')} \quad (38.3)$$

با توجه به (۳۸.۳)، مشاهده می‌کنیم که جهت محاسبه‌ی LOF برای نقطه‌ی  $p$ ، نه از مقادیر چگالی نسبی همسایگان آن که در فضای اصلی محاسبه شده‌اند، بلکه از مقادیر چگالی نسبی همسایگان  $p$  که در مجموعه‌ی  $(RP^{-1}(N_h(p'))$  قرار دارند و در فضای تصویرشده محاسبه شده‌اند، استفاده می‌کنیم. حسن این کار در آن است که محاسبه‌ی مقادیر چگالی نسبی در فضای تصویرشده، به لحاظ محاسباتی به مراتب سبک‌تر از همین محاسبات در فضای اصلی می‌باشد، ولی همین مسئله سبب می‌شود تا مقادیر

LOF نهائی از دقت تام برخوردار نباشند. اما با این وجود می‌توان نشان داد که این مقادیر تقریبی محدود به بازه‌ی زیر می‌باشند:

$$\frac{1 - \epsilon}{1 + \epsilon} \cdot LOF(p) \leq \overline{LOF}(p) \leq \frac{1 + \epsilon}{1 - \epsilon} \cdot LOF(p) \quad (39.3)$$

لازم به ذکر است که مقدار  $h$  را معمولاً ضریب کوچکی از  $k$  و برابر  $2k$  یا  $3k$  در نظر می‌گیرند، و البته که با افزایش مقدار  $h$  صحت نتایج افزایش پیدا می‌کند، اما طبعاً حجم محاسبات نیز با توجه به یافتن  $-h$  نزدیک‌ترین همسایه در فضای کاهش‌بعدیافته به صورت خطی زیاد می‌شود.



شکل ۵.۳ مراحل الگوریتم PINN.[۱۸]

تا این جای کار توانستیم نزدیک‌ترین همسایگان حاصله از شاخص تصویرسازی یا همان PINN را به دست آوریم و دیدیم که در ابتدای کار می‌بایست داده‌ها را به یک فضای با ابعاد کوچکتر تصویر کنیم. روشی که ما در اینجا جهت تصویرسازی یا همان رویه‌ی کاهش بُعد استفاده کردیم، روش تصویرسازی تصادفی یا همان RP بود، اما نکته این‌جاست که PINN می‌تواند از هر رویه‌ی کاهش بعدی که فاصله‌های اقلیدسی میان داده‌ها در آن پس از تصویرسازی، با تخمین بالائی حفظ گردد، به عنوان

ورودی استفاده کند. از جمله‌ی این روش‌های کاهش بعد، روش محبوب PCA می‌باشد که پیش‌تر به آن اشاره شد و مشکل اصلی آن نیز بار محاسباتی و زمانی بالا بود.

در اینجا الگوریتم پیشنهادی این روش را جهت محاسبه‌ی مقدار LOF به ازای هر داده و با استفاده از روش‌های RP و PINN ارائه می‌نماییم. شکل ۵.۳ نیز رویه‌ی کلی الگوریتم PINN را به صورت مرحله‌به‌مرحله نشان می‌دهد.

### **RP + PINN + LOF Algorithm**

*Input:* The  $n$  by  $m$  matrix  $X$  of data in the original space.

*Output:* The Local Outlier Factor (LOF) score and ranking for each point in  $X$ .

*RP:*

Project  $X$  to a  $n$  by  $t$  matrix  $Y$ ,  $t < m$ , using the random projection scheme described in (16).

*PINN:*

Define  $h$  as the parameter for defining the size of the set of candidate nearest-neighbors used, where  $h \geq k$ .

For each point  $p \in X$ :

Find  $h$ -nearest-neighbors of  $p'$  in the projected space  $Y$ , forming the candidate nearest-neighbor set  $N_h(p')$ .

Map the points in the candidate set  $N_h(p')$  back to the original space ( $X$ ), forming the set  $RP^{-1}(N_h(p'))$ .

Find the  $k$  items of  $RP^{-1}(N_h(p'))$  closest to  $p$ . Call this set  $\bar{N}_k(p)$ .

*LOF:*

For each point  $p \in X$ , estimate  $LOF(p)$  by computing

$$\overline{LOF}(p) = \frac{\frac{1}{k} \sum_{q \in \bar{N}_k(p)} rd(q')}{rd(p')}$$

## ۸.۳ کشف داده‌های پرت مکانی با استفاده از مدل یادگیری

### خودسازمان‌دهنده‌ی تکراری و تخمین فاصله‌ی مستحکم

«کای»<sup>۱۶۲</sup> و همکاران [۱۹]، روشی را جهت کشف داده‌های پرت در مجموعه‌داده‌های به اصطلاح «مکانی»<sup>۱۶۳</sup> ارائه نموده‌اند. داده‌های مکانی، به آن دسته از داده‌ها اطلاق می‌گردد که دارای یک سری

<sup>162</sup> Qiao Cai

ویژگی‌های مکانی و موقعیتی مانند طول جغرافیایی، عرض جغرافیایی و ارتفاع از سطح دریا، در کنار ویژگی‌های غیر مکانی مانند تراکم جمعیت و توزیع سن افراد، به ازای هر نمونه‌داده می‌باشند. روابط مکانی میان داده‌ها عموماً با یک «ماتریس مجاورت»<sup>۱۶۴</sup> مشخص می‌گردد که درایه‌های آن معرف وجود مجاورت یا فاصله‌ی میان داده‌ها می‌باشد. وجود ویژگی‌های مکانی و موقعیتی در این نوع مجموعه‌داده‌ها سبب می‌شود تا به هنگام خوشبندی، آن دسته از داده‌ها که به لحاظ جغرافیائی در مجاورت یکدیگر قرار دارند، تشکیل یک خوش‌دهند. این جاست که رویه‌ی کشف داده‌های پرت مکانی، به دنبال آن دسته از داده‌های ویژگی‌های غیر مکانی آن‌ها نسبت به سایر همسایگان مکانی آن‌ها که در یک خوش‌دهند، قرار گرفته‌اند، به طرز قابل توجهی متفاوت است. به عبارت دیگر، داده‌ی پرت مکانی را می‌توان در قالب یک ناسازگاری محلی در نظر گرفت که در آن یک داده‌ی خاص، نسبت به همسایگان مکانی خود دارای ویژگی‌های غیر مکانی بالذات مرتبط می‌باشد، ولی به صورت آشکارا رفتار متفاوتی از خود بروز می‌دهد. داده‌های پرتی که در مجموعه‌داده‌های مکانی وجود دارد در دو دسته‌ی زیر قرار می‌گیرند: داده‌های پرت مبتنی بر فضای چندبعدی، و داده‌های پرت مبتنی بر گراف. داده‌های پرت مبتنی بر فضای چندبعدی، مبتنی بر فواصل اقلیدسی هستند، در حالی که داده‌های پرت مبتنی بر گراف از قوانین نمایش در فضای گرافی یا همان اتصالات گرافی پیروی می‌نمایند. دلایل بسیاری وجود دارد که با توجه به آن‌ها کشف داده‌های پرت مکانی هم‌چنان یک مسئله‌ی جدی باقی مانده است. مهم‌ترین آن‌ها این است که تعریف یک همسایگی دقیق و مشخص به ازای هر داده، خود یک مسئله‌ی بسیار مهم جهت شناسائی داده‌های پرت مکانی می‌باشد. مشکل دیگر در مورد رویکردهای آماری می‌باشد که نیازمند آن هستند تا توزیع آماری مقادیر هر یک از ویژگی‌ها را در مکان‌های مختلف، در مقایسه با توزیع توأم مقادیر تمامی ویژگی‌ها و نسبت به داده‌های همسایه قیاس نمایند.

از جمله روش‌هایی که جهت کشف داده‌های پرت مکانی به کار رفته‌اند، روش‌های آماری هستند که داده‌های پرت را در مجموعه‌داده‌های مکانی دارای بیش از یک ویژگی غیرمکانی کشف می‌نمایند. بدین ترتیب که از ویژگی‌های مکانی، جهت تعریف همسایگی هر داده استفاده کرده و از ویژگی‌های غیر مکانی، جهت یافتن کاندیداهای داده‌ی پرت بر اساس معیار فاصله‌ی مahaalanobis با حدآستانه‌ی برش

<sup>163</sup> Spatial

<sup>164</sup> Contiguity matrix

مبتنی بر توزیع  $\chi^2$  استفاده می‌نمایند. مشکل این روش آن است که موقعیتی را که در آن ویژگی‌های مکانی به حدی پیچیده می‌شوند که دیگر نمی‌توان به راحتی برای هر داده یک همسایگی مشخص تعریف نمود، نادیده می‌گیرد و به همین سبب، در بسیاری موارد جهت کشف داده‌های پرت سراسری کاربرد داشته و از کشف داده‌های محلی عاجز می‌مانند. یک روش دیگر که جهت کشف داده‌های پرت مکانی به کار می‌رود آن است که داده‌های مکانی را با توزیع نرمال برآش نموده و به دنبال آن دسته از داده‌هایی باشیم که ویژگی‌های غیر مکانی آن‌ها از حد نرمال مثلاً برابر  $3\sigma + \mu$  تجاوز می‌نماید، که البته این روش نیز ایراداتی دارد. از جمله آن که این حدآستانه‌ی برش می‌تواند سبب گردد تا برخی داده‌های نرمال به عنوان داده‌ی پرت شناسائی گردند. به طور کلی، اکثر روش‌های آماری که تاکنون به کار رفته‌اند، به سختی می‌توانند یک حدآستانه‌ی برش صدکی (بر حسب درصد) مُتقن و مستحکم جهت یافتن داده‌های پرت مکانی تعریف نمایند.

در این روش، یک رویکرد مبتنی بر «نقشه‌ی خودسازمان‌دهنده‌ی تکراری با تخمین فاصله‌ی مستحکم»<sup>۱۶۵</sup>، که به اختصار ISOMRD نامیده می‌شود، جهت کشف داده‌های پرت مکانی ارائه گشته است. در این رویکرد، از «شبکه‌ی عصبی نقشه‌ی خودسازمان‌دهنده»<sup>۱۶۶</sup> یا به اختصار SOM، جهت کسب ساختار مکانی و ارتباطات ذاتی میان داده‌ها استفاده می‌شود تا به دنبال آن خوشه‌های به اصطلاح مکانی کل مجموعه‌داده حاصل گشته و بدین‌وسیله رویه‌ی کشف داده‌های پرت آغاز گردد. رویه‌ی تکرارشونده‌ی شبکه‌ی SOM به همراه تخمین فاصله‌ی مستحکم پیشنهادشده به طور مختص در این روش، سبب می‌گردد تا بتوان با استفاده از آن، داده‌های پرت را در مجموعه‌داده‌های با ابعاد مکانی بسیار زیاد که ویژگی‌های غیر مکانی آن‌ها متمایزتر است کشف نمود. در ادامه در قسمت شرح روش و پارامترها به بیان جزئیات بیشتری از روش پیشنهادی می‌پردازیم.

<sup>۱۶۵</sup> Iterative Self-Organizing Map (SOM) with Robust Distance Estimation (ISOMRD)

<sup>۱۶۶</sup> Self-Organizing Map (SOM) Neural Network

### ۱.۸.۳ شرح روش و پارامترها

در این قسمت باید خاطرنشان نمود که استفاده از شبکه‌ی عصبی SOM جهت کشف داده‌های پرت مکانی پیش از این نیز مرسوم بوده است، اما در این روش، رویکرد پیشنهادشده مبتنی بر تخمین فاصله‌ی مستحکم جهت شناسائی دقیق‌تر داده‌های پرت مکانی است.

### ۲.۸.۳ تجمیع شبکه‌ی عصبی SOM با تخمین فاصله‌ی مستحکم جهت کشف داده‌های پرت مکانی

شبکه‌ی عصبی SOM از جمله شبکه‌های عصبی است که بنای آن بر یادگیری رقابتی می‌باشد و هدف اصلی آن تصویرسازی بردارهای ورودی با ابعاد بسیار زیاد به یک نقشه‌ی گسسته‌ی تکبعدی و یا دوبعدی است که در نهایت خوشبندی نهائی با استفاده از همین نقشه‌ی با ابعاد کمتر صورت می‌گیرد. در اینجا اگر بخواهیم به اختصار رویه‌ی شبکه‌ی SOM را توضیح دهیم، باید گفت که روند یادگیری در این شبکه‌ی رقابتی شامل سه مرحله‌ی زیر می‌شود: «فاز رقابتی»<sup>۱۶۷</sup>، «فاز همکاری»<sup>۱۶۸</sup> و «فاز سازگاری»<sup>۱۶۹</sup>. اگر تصور کنیم که  $x_i$  نمایان‌گر  $i$ -امین بردار ورودی و  $w_j$  نشان‌گر  $j$ -امین بردار وزن شبکه‌ی SOM و با همان ابعاد بردارهای ورودی باشد، آن‌گاه به هنگام شروع یادگیری، «بهترین واحد همتا»<sup>۱۷۰</sup> نسبت به  $x_i$  که با  $b_i$  نمایش داده می‌شود، آن بردار وزنی خواهد بود که کمترین فاصله را نسبت به بردار ورودی و در میان سایر بردارهای وزن داشته باشد. داریم:

$$b_i = \arg \min_j \|x_i - w_j\| \quad (40.3)$$

پس از برنده شدن یکی از نورون‌ها، همسایگان آن نورون نیز به نسبت فاصله‌ای که از آن دارند، شانس برنده شدن در مراحل بعدی را خواهند داشت و در نتیجه بدین ترتیب، «نورون‌های سیناپسی»<sup>۱۷۱</sup>، یک

<sup>167</sup> Competitive phase

<sup>168</sup> Cooperative phase

<sup>169</sup> Adaptive phase

<sup>170</sup> Best Matching Unit (BMU)

<sup>171</sup> Synaptic Neurons

زیرمجموعه‌ی رقابتی را تشکیل داده و در نهایت منجر به تشکیل یک خوشه می‌گردد. در اینجا، می‌بایست برای تعیین شدت برنده‌شدن نورون‌های همسایه که به معنی میزان جابجایی آن‌ها پس از مشاهده‌ی بردار ورودی  $x_i$  می‌باشد، از یک تابع به اصطلاح کرنل یا هسته استفاده نمائیم، که انتخاب ما در اینجا تابع کرنل گاوسین خواهد بود. داریم:

$$\Phi_{j,i}(n) = \exp\left(-\frac{\|r_i - r_j\|^2}{2\sigma_0^2 \exp\left(-\frac{2n}{\tau_1}\right)}\right) \quad (41.3)$$

به طوری که  $n$  برابر شماره‌ی «ایپاک»<sup>۱۷۲</sup> فعلى،  $r_j$  معرف بردار وزن در فضای دو بعدی،  $\sigma_0$  برابر شعاع اولیه‌ی همسایگی در فضای دو بعدی و  $\tau_1$  نیز برابر ثابت زمانی در فاز یادگیری رقابتی می‌باشد. بدین ترتیب به ازای هر کدام از داده‌های ورودی، شبکه‌ی دو بعدی SOM تحریک شده و تعدادی از نورون‌های آن به سمت بردار ورودی متمایل می‌گردد. فرمول به روزرسانی هر کدام از بردارهای وزن در قالب زیر می‌باشد:

$$\begin{aligned} w_j(n+1) &= w_j(n) + \eta(n)\Phi_{j,i}(n)\left(x_i(n) - w_j(n)\right) \\ &= \left(1 - \eta(n)\Phi_{j,i}(n)\right)w_j(n) + \eta(n)\Phi_{j,i}(n)x_i(n) \end{aligned} \quad (42.3)$$

به طوری که  $\eta(n)$  معرف تابع نرخ یادگیری می‌باشد.

همان‌طور که پیش از این قید شد، رویه‌ی تکراری شبکه‌ی SOM قادر به آن است تا در مجموعه داده‌های مکانی، همسایگی‌های میان داده‌ها را با توجه به تعداد زیاد ویژگی‌های مکانی و غیر مکانی پیدا نموده و بدین ترتیب خوشه‌های مکانی و موقعیتی را شکل می‌دهد. در این‌جا لازم است تا برای یافتن داده‌های پرت مکانی، یک حدآستانه‌ی برش را جهت مرز میان داده‌های نرمال و داده‌های غیرنرمال یا پرت تعریف نمائیم. این حدآستانه‌ی برش برای یک فاصله‌ی خاص، تحت عنوان «فاصله‌ی مستحکم»<sup>۱۷۳</sup> تعریف شده و با نام «کمینه‌ی دترمینان ماتریس کوواریانس»<sup>۱۷۴</sup> شناخته می‌شود.

<sup>172</sup> Epoch

<sup>173</sup> Robust Distance

<sup>174</sup> Minimum Covariance Determinant (MCD)

در اینجا، باید خاطرنشان کرد که در برخی موارد میان ویژگی‌های مکانی و غیر مکانی، وابستگی وجود دارد. به عنوان مثال، اگر شیء مورد نظر یک ساختمان باشد، برخی ویژگی‌های مکانی آن شامل مکان جغرافیائی آن، شکل آن (سقف شیروانی یا مسطح)، ارتفاع ساختمان و شمالی یا جنوبی بودن آن می‌شود؛ و برخی ویژگی‌های غیر مکانی آن نیز شامل مصالح ساختمانی استفاده شده در آن، رنگ، تاریخ ساخت و یا سبک ساخت آن می‌شود. حال اگر در اینجا قیمت ساختمان را نیز به عنوان یکی از ویژگی‌های غیرمکانی آن در نظر بگیریم، آن‌گاه مکان و موقعیت جغرافیائی‌ای که ساختمان در آن‌جا بنا شده است، به نوعی با قیمت آن مرتبط خواهد بود. چرا که قیمت یک ساختمان در مثلاً پائین شهر با قیمت همین ساختمان در بالای شهر شدیداً اختلاف داشته و در نتیجه نمی‌توان برخی ارتباطات محتمل میان ویژگی‌های مکانی و غیر مکانی را نادیده گرفت.

حال در اینجا پیش از آن که به معرفی الگوریتم پیشنهادی در این روش با نام ISOMRD بپردازیم، می‌بایست دو تعریف زیر را بیان نمائیم:

- **تعریف اول:** اگر مجموعه‌داده‌ی ورودی را به صورت  $\{o_1, \dots, o_n\}$  نشان دهیم، می‌توانیم تابع ویژگی مکانی را به صورت  $S(o_i) \leftarrow s_i$  تعریف کنیم که از خروجی آن جهت کسب همسایگان مکانی  $i$  استفاده می‌شود. حال تابع همسایگی که ورودی آن  $s_i$  می‌باشد را به صورت  $G(s_i) = (1/|N_{s_i}|) \sum_{k=1}^{|N_{s_i}|} n_k$  تعریف می‌کنیم که در آن  $N_{s_i} = \{n_1, \dots, n_K\}$  معرف همسایگی و  $K$  نیز تعداد اعضای این مجموعه می‌باشد. پس از تعریف توابع مربوط به ویژگی‌های مکانی، تابع ویژگی غیرمکانی را به صورت  $A(o_i) \leftarrow a_i$  معرفی می‌کنیم. می‌توان جهت تسهیل در امر کشف داده‌های پرت، مقادیر ویژگی‌های غیر مکانی را به صورت  $a_i = A(o_i) - \mu_A/\sigma_A$  به اصطلاح نرمال‌سازی نمود تا بدین‌وسیله مقادیر ویژگی‌های غیر مکانی با توزیع گاویسین برازش شده و بدین‌ترتیب بتوان از مقادیر حدآستانه‌ی مربوطه جهت کشف داده‌های پرت استفاده نمود. در این‌جا یک تابع دیگر با نام تابع قیاس را نیز تعریف می‌نمائیم که مقادیر نرمال‌شده ویژگی‌های غیر مکانی یک داده را با مقادیر مربوط به همسایگان مکانی آن مقایسه می‌نماید. این تابع به صورت  $H(o_i) = H(a_i) = G(s_i)$  تعریف می‌شود.

- **تعریف دوم:** جهت تعریف حدآستانه‌ی برش برای کشف داده‌های پرت، می‌بایست به توزیع  $F$  متولّ شویم. اگر توزیع  $F$  با پارامتر حد اطمینان خاص  $\beta$  را به صورت  $F_{q,m-q+1}(\beta)$  نشان دهیم، آن‌گاه می‌توانیم حدآستانه‌ی لازم جهت کشف داده‌های پرت مکانی را به صورت  $Th = \lambda F_{q,m-q+1}(\beta)$  تعریف نمائیم، که در آن می‌توان پارامتر  $\lambda$  را با استفاده از کمینه‌ی دترمینان ماتریس کوورایانس یا همان  $MCD$  که پیش‌تر معرفی شد، تخمین زد. با توجه به همین تخمین‌گر  $MCD$  می‌توان معیار فاصله‌ی مستحکم به ازای هر داده‌ی مکانی را به صورت  $rd_i \leftarrow MCD(o_i)$  :  $rd_i \leftarrow MCD(o_i)$  به دست آورد.

حال می‌توانیم با توجه به تعاریف و توضیحاتی که تا این جای کار معرفی گشتند، الگوریتم پیشنهادی را به صورت زیر تعریف نمائیم:

### ISOMRD Algorithm

*Input:*

(1) Spatial dataset:  $O : \{o_1, \dots, o_n\}$ , where  $n$  is the number of input data

(2) Total neuron number:  $N_m$

(3) Maximum iteration:  $\text{max\_iter}$

$$a_i \leftarrow \frac{A(o_i) - \mu_A}{\sigma_A}$$

repeat

    for  $i=1$  to  $\text{max\_iter}$  do

        Search BMU  $b(a_i)$  via Eq. (27)

        for  $j=1$  to  $N_m$  do

            Update  $w_j(i+1)$  via Eq. (29)

        end for

    end for

Calculate the neighborhood function  $G(s_i) = \frac{1}{|N_{s_i}|} \sum_{k=1}^{|N_{s_i}|} n_k$

Calculate the comparison function  $H(o_i) = a_i - G(s_i)$

Calculate the robust distance RD through MCD estimator in Eq. (33)

Select the input data with largest RD and remove it from input dataset

until The  $m_{th}$  outlier candidate is obtained

for  $k=1$  to  $m$  do

    if  $rd_k^2 > Th$  then

$o_k$  can be identified as a spatial outlier

    end if

end for

## ۱,۲,۸,۳ فاصله‌ی ماهالانوبیس

حال اگر بخواهیم معیار فاصله‌ی مستحکم به ازای هر داده مکانی را دقیق‌تر تعریف کنیم، ابتدا می‌بایست اشاره‌ی کوتاهی به فاصله‌ی ماهالانوبیس بنماییم. فاصله‌ی ماهالانوبیس در امورات یادگیری ماشین و داده‌کاوی از جمله ابزارهای آماری کاربردی و قدرتمند جهت یافتن الگوهای موجود در داده‌ها بوده و خصوصاً در امر خوشبندی داده‌ها شدیداً کارا می‌باشد. به طوری که به عنوان مثال، فاصله‌ی اقلیدسی تنها می‌تواند خوشبتهای با شکل «ابرگره»<sup>175</sup> را پیدا نماید، در حالی که فاصله‌ی ماهالانوبیس قادر به

<sup>175</sup> Hyper-sphere

یافتن خوشهای به شکل «ابربیضی»<sup>۱۷۶</sup> نیز می‌باشد. علت این مسئله نیز آن می‌باشد که فاصله‌ی ماهالانوبیس، «همبستگی»<sup>۱۷۷</sup> و «وابستگی»<sup>۱۷۸</sup> میان ویژگی‌های یک مجموعه‌داده را در نظر می‌گیرد و به همین سبب می‌توان با استفاده از آن، الگوهای ناموزون را در مجموعه‌داده کشف نمود. فاصله‌ی ماهالانوبیس را می‌توان به عنوان یک شاخص تمایز میان دو متغیر تصادفی با ماتریس کوواریانس یکسان و به عبارتی با «توزیع آماری کاملا همسان»<sup>۱۷۹</sup> نیز در نظر گرفت که به صورت زیر نمایش داده می‌شود:

$$MD_i = \sqrt{(x_i - \bar{x})^T S^{-1} (x_i - \bar{x})} \quad (43.3)$$

به طوری که  $\bar{x}$  معرف میانگین نمونه‌ها در تمامی ابعاد بوده و  $S$  نیز معرف ماتریس کوواریانس داده‌ها می‌باشد که در واقع نرمال‌شده‌ی همان «ماتریس پراکندگی»<sup>۱۸۰</sup> داده‌ها در فضای  $m$ -بعدی می‌باشد.

### ۲.۲.۸.۳ معیار فاصله‌ی مستحکم مبتنی بر کمینه‌ی دترمینان ماتریس کوواریانس

با توجه به این‌که فاصله‌ی ماهالانوبیس اطلاعاتی در مورد شکل ساختاری مجموعه‌داده در فضای چندبعدی در اختیار ما قرار می‌دهد، می‌تواند خصیصه‌های مجانبی مربوط به توزیع  $\chi^2$  را نیز با استفاده از ماتریس کوواریانس مربوطه پوشش دهد. کمینه‌ی دترمینان ماتریس کواریانس یا MCD می‌تواند به عنوان یک تخمین‌گر مستحکم و با « نقطه‌ی شسکت بالا»<sup>۱۸۱</sup> عمل کند. لازم به ذکر است که هرچه نقطه‌ی شسکت یا همان به اصطلاح breakdown یک تخمین‌گر بالا باشد، به همان نسبت نرخ مشاهدات نادرستی که آن تخمین‌گر پیش از دچار شدن به خطای مهلک می‌تواند تحمل نماید نیز بیشتر خواهد بود. به عبارتی هرچه breakdown تخمین‌گر بالاتر باشد، به همان نسبت صحت نتایج نیز بیشتر خواهد بود. با این وجود در مورد مجموعه‌داده‌های با مقیاس بزرگ، عملکرد تخمین توزیع  $\chi^2$  با استفاده از MCD چندان رضایت‌بخش نمی‌باشد، چرا که به سختی می‌توان مقادیر حدآستانه‌ی برش را برای چنین

<sup>176</sup> Hyper-ellipse

<sup>177</sup> Correlation

<sup>178</sup> Dependency

<sup>179</sup> Identical Statistical Distribution

<sup>180</sup> Scatter Matrix

<sup>181</sup> High Breakdown

مجموعه‌داده‌های بزرگی حتی با وجود چنین تخمین‌گر مناسب و با نقطه‌ی شکست بالائی به دست آورد. در این جاست که می‌توان از توزیع  $F$  که در واقع، بنای آن بر توزیع  $\chi^2$  استوار است، جهت تعیین معیار فاصله‌ی مستحکم استفاده نموده و به ازای مجموعه‌داده‌های با اندازه‌های متنوع، کاندیداهای مناسب داده‌ی پرت را یافت نمود. البته نتایج حاصله از توزیع‌های  $F$  و  $\chi^2$  نشان‌دهنده مقادیر متفاوت برای پارامترهای مورد نیاز جهت کشف داده‌های پرت می‌باشد، اما باز هم توزیع  $F$  با توجه به آزمایشات انجام‌شده نتایج بهتری را خصوصاً درباره‌ی مجموعه‌داده‌های بزرگ به دست می‌دهد.

حال در اینجا می‌خواهیم فاصله‌ی مستحکم هر داده از مرکز خوش‌های که به آن تعلق دارد را تعریف نمائیم. به همین دلیل بهتر آن است تا به ازای  $d$  داده‌ای که از تمامی  $n$  عضو مجموعه‌داده به یک خوش تعلق دارند، دترمینان ماتریس کوواریانس را کمینه‌سازی نمائیم و برای همین از بردار میانگین و ماتریس کوواریانس داده‌های خوش‌های مربوطه استفاده می‌نمائیم. داریم:

$$\begin{aligned} \bar{\mu}_{MCD}^* &= \frac{1}{d} \sum_{j \in G} x_i \\ \Sigma_{MCD}^* &= \frac{1}{d} \sum_{j \in G} (x_i - \bar{\mu}_{MCD}^*)(x_i - \bar{\mu}_{MCD}^*)^T \\ RD_i &= \sqrt{(x_i - \bar{\mu}_{MCD}^*)^T \Sigma_{MCD}^* (x_i - \bar{\mu}_{MCD}^*)} \end{aligned} \quad (44.3)$$

لازم به ذکر است که نقطه‌ی شکست یا همان *breakdown* مقدار  $d$ ، به تعداد داده‌های مجموعه‌داده و البته تعداد ابعاد آن وابسته می‌باشد.

۴

## فصل چهارم

### روش پیشنهادی

در این فصل، قصد داریم تا یک روش پیشنهادی را جهت کشف داده‌های پرت محلی در کلان‌داده‌ها ارائه نمائیم. بنای این روش بر خوشبندی مقیاس‌پذیر می‌باشد، به طوری که در هر مرحله، از یک روش خوشبندی مبتنی بر چگالی جهت شناسائی نواحی چگال استفاده می‌شود. بدین معنی که از آنجایی که یک مجموعه‌داده‌ی کلان، قابلیت جاگرفتن در حافظه‌ی اصلی یا RAM را به یکباره دارا نیست، لذا ناچاریم تا آن را به صورت «قطعه‌قطعه»<sup>۱</sup> پردازش نمائیم، به گونه‌ای که هر قطعه از کلان‌داده در آن واحد، هم قابلیت جاگرفتن در RAM را داشته باشد و هم این‌که فضای کافی جهت پردازش آن را نیز در اختیار داشته باشیم. سپس در هر مرحله، با توجه به داده‌هایی که در حال حاضر در حافظه قرار دارند و البته داده‌هایی که پیش از این در RAM بلا تکلیف مانده‌اند، اطلاعات مربوط به مدل خوشبندی را به‌روز می‌نمائیم. این رویه‌ی به‌روزرسانی می‌تواند شامل تغییراتی در اطلاعات خوشبندی فعلی و یا هم اضافه‌شدن اطلاعات مربوط به خوشبندی جدید باشد. در تمام طول رویه‌ی خوشبندی، تلاش ما آن است تا داده‌های پرت در تشکیل و به‌روزرسانی خوشبندی نخواهند بودند. در پایان امر خوشبندی مقیاس‌پذیر، پس از ساخته‌شدن مدل خوشبندی موقّت، با اعمال پارامترهای مناسب و با استفاده از یک الگوریتم خوشبندی مناسب، این خوشبندی موقّت را با یکدیگر ترکیب نموده و ساختار خوشبندی نهائی را به دست می‌آوریم. در ادامه، برای این‌که به هر داده، امتیازی مبنی بر میزان پرت‌بودن انتساب دهیم، کافی است تا از معیار فاصله‌ی مahaalanobis جهت پیداکردن نزدیک‌ترین خوشبندی به هر داده استفاده کنیم و این کمترین مقدار فاصله را به عنوان ضریب داده‌ی پرت محلی برای آن داده در نظر بگیریم. این مقدار فاصله‌ی مahaalanobis را «فاصله‌ی مahaalanobis محلی»<sup>۲</sup> نیز گویند.

روش پیشنهادی ما، در واقع حالت توسعه‌یافته‌ای از الگوریتم BFR [۸] می‌باشد. در این الگوریتم، «فرض قوی»<sup>۳</sup> اولیه بر آن است که مجموعه‌داده‌ی ورودی، متشكل از خوشبندی دارای توزیع نرمال با «مقادیر ویژگی ناهمبسته»<sup>۴</sup> می‌باشد و هم این‌که روش BFR، مجموعه‌داده‌ی ورودی را عاری از هر گونه داده‌ی پرت فرض می‌نماید. اما در روش پیشنهادی، مجموعه‌داده‌ی ورودی می‌تواند دارای خوشبندی نرمال اما

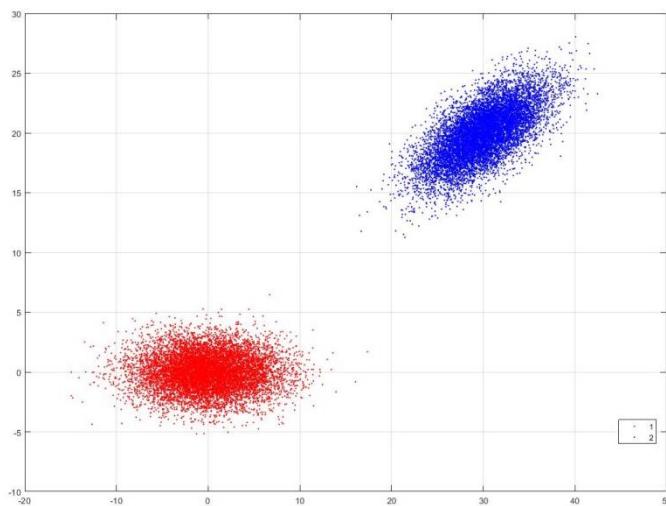
<sup>1</sup> Chunk by chunk

<sup>2</sup> Local Mahalanobis distance

<sup>3</sup> Strong assumption

<sup>4</sup> Uncorrelated features

با «مقادیر ویژگی همبسته»<sup>۱</sup> باشد و البته روش ما، وجود داده‌های نویزی و پرت را در دادگان ورودی پذیرفته و قادر به شناسائی آن‌ها با دقت بالائی در صورت برقراربودن فرضیات قوی اولیه می‌باشد. شکل ۱.۴ نمونه‌ای از خوشه‌های دارای توزیع نرمال و با مقادیر ویژگی ناهمبسته و همبسته را نشان می‌دهد.



شکل ۱.۴ نمونه‌ای از خوشه‌های دارای توزیع نرمال در فضای دو بعدی؛ در خوشه‌ی قرمزنگ، مقادیر ویژگی‌ها ناهمبسته و در مورد خوشه‌ی آبی‌رنگ، همبسته می‌باشند

در مورد الگوریتم BFR باید گفت که به طور پیش‌فرض از الگوریتم Kmeans [۲۰] جهت خوشه‌بندی داده‌های کنونی موجود در RAM و به روزرسانی مدل خوشه‌بندی استفاده می‌نماید و این در حالی است که در صورت حضور داده‌های پرت در مجموعه‌داده، آن‌ها نیز بالاخره به یک خوشه تعلق گرفته و در شکل گیری ساختار آن خوشه نقش مؤثر ایفا خواهند نمود. چرا که الگوریتم Kmeans قادر به تشخیص داده‌های نویزی و پرت نبوده و تمرکز آن بر روی بهبود موقعیت مکانی میانگین خوشه‌ها می‌باشد. لذا در نهایت، هر داده، چه نرمال و چه غیر نرمال باید به یک خوشه تعلق گرفته باشد. با توجه به فرض قوی اولیه‌ی الگوریتم BFR، زمانی که دادگان ورودی عاری از هر گونه نویز باشد، استفاده از الگوریتم مشکلی را ایجاد نخواهد نمود. اما در مورد دادگان دارای نویز، باید از یک رویکرد دیگر جهت خوشه‌بندی استفاده نمائیم. رویکردی که مبتنی بر بهبود موقعیت مکانی مرکز خوشه‌ها نبوده و بلکه قادر به شناسائی نواحی چگال یک مجموعه‌داده از نواحی غیرچگال آن باشد. بدین‌وسیله خوشه‌ها که متشکل از داده‌های متراکم می‌باشند، از این‌که داده‌های پرت در شکل‌گیری ساختار آن‌ها نقش داشته

<sup>۱</sup> Correlated features

باشند، مصون خواهند ماند. یکی از الگوریتم‌های محبوب خوشبندی که مبتنی بر چگالی است، الگوریتم DBSCAN [۲۱] می‌باشد. در روش پیشنهادی ما نیز به جهت این فرض اولیه که دادگان ورودی، دارای نویز و داده‌ی پرت می‌باشد، ارجحیت خوشبندی در هر به اصطلاح «بارگیری حافظه از داده‌ها»<sup>۱</sup> با استفاده از الگوریتم DBSCAN می‌باشد. البته باید گفت که در رویه‌ی خوشبندی مقیاس‌پذیر، با توجه به این فرض اولیه که دادگان ورودی دارای داده‌های پرت می‌باشد و هم این که جهت انتساب داده‌های موجود در RAM به هر خوش، از معیار فاصله‌ی مahaلانوبیس استفاده می‌کنیم (که بنای آن بر «ماتریس کوواریانس»<sup>۲</sup> می‌باشد)، الگوریتم DBSCAN نیز از یک نقطه ضعف جدی رنج می‌برد که در ادامه به آن اشاره خواهیم کرد. لذا برای رفع این نقطه ضعف جدی، ناچار خواهیم بود تا پس از یک مرحله خوشبندی با استفاده از الگوریتم DBSCAN، روی خوش‌های حاصله از این مرحله، مجددًا خوشبندی انجام دهیم که در این روش پیشنهادی، ما از همان الگوریتم Kmeans برای خوشبندی مرحله‌ی دوم استفاده خواهیم نمود. در مورد استفاده از الگوریتم Kmeans در مرحله‌ی دوم خوشبندی، مقدار بهینه‌ای که برای K معرفی می‌گردد، کافی است تا حائز دو شرط اصلی باشد که در ادامه معرفی خواهد شد. اما نیازی نمی‌باشد که کاربر این مقدار را معرفی نماید. به عبارتی تنها کافی است تا مقدار K را از کمترین میزان ممکن، تا آن‌جایی افزایش دهیم که دو شرط لازم برآورده گردند و آن مقدار، بهترین مقدار برای K خواهد بود. اما در مورد الگوریتم BFR که تنها از الگوریتم Kmeans بهره می‌برد، می‌بایست مقدار K را از پیش بدانیم که این خود، مستلزم داشتن آگاهی اولیه در مورد مجموعه‌داده‌ی مربوطه می‌باشد، و این مهم همیشه برآورده شدنی نخواهد بود.

نکته‌ی بعدی در مورد روش پیشنهادی، درباره‌ی تفاوت میان روش ما با روش BFR، در فرض اولیه در مورد ساختار خوش‌های نرمال می‌باشد. روش BFR، با خوش‌های نرمال با مقادیر ویژگی همبسته سازگار نمی‌باشد، ولی در عوض حجم اطلاعات ساختاری که در مورد خوش‌ها نگهداری می‌نماید بسیار سبک‌تر از روش پیشنهادی است. علت این مسئله نیز آن است که در روش پیشنهادی، در حالتی که همبستگی میان ویژگی‌ها صفر نیست، ماتریس کوواریانس یک خوش، دیگر قطری نبوده و درایه‌های غیر صفر بسیاری خواهد داشت. از آن‌جا که در هر دوی روش‌های BFR و پیشنهادی از معیار فاصله‌ی

<sup>1</sup> Memory-Load of points

<sup>2</sup> Covariance Matrix

ماهالانوبیس جهت انتساب یک داده به یک خوش استفاده می‌شود و البته استفاده از این معیار، مستلزم در اختیار داشتن ماتریس کوواریانس یک خوش می‌باشد، لذا ماتریس کوواریانس هر خوش، قلب تپنده‌ی آن خوش بوده و با اضافه‌شدن داده‌های جدید و به عبارتی رشد آن، اطلاعات حیاتی هر خوش به‌روز خواهد شد. اما از آن‌جا که برای استفاده از معیار فاصله‌ی ماهالانوبیس، نیاز به معکوس‌گرفتن از ماتریس کوواریانس داریم، این عملیات معکوس‌گیری خصوصاً در ابعاد بالا هزینه‌ی بسیاری دارد (با وجود این‌که این عملیات می‌تواند پس از پردازش هر مرحله بارگیری حافظه از داده‌ها انجام شود). همین‌طور زمانی که می‌خواهیم فاصله‌ی ماهالانوبیس یک داده را از میانگین یک خوش با درنظرگرفتن ماتریس کوواریانس آن محاسبه کنیم، عملیات ماتریسی لازم مشتمل بر ضرب و جمع‌های مربوطه، به لحاظ محاسباتی، بسیار سنگین خواهد بود و در ابعاد بالا این سنگینی، بیشتر نیز حس خواهد شد. توجه به این نکته نیز ضروری می‌نماید که در طول اجرای الگوریتم، باید به ازای تک‌تک داده‌ها بررسی نمائیم که آیا با توجه به پارامترهای مربوطه می‌توانند به خوش‌های تعلق بگیرند یا خیر؛ و در نتیجه می‌بایست همگی محاسبات گفته‌شده از جمله عملیات سنگین ماتریسی را به ازای هر داده و نسبت به همه‌ی خوش‌ها متحمل شویم.

با توجه به آن‌چه که در مورد ماتریس کوواریانس و محاسبات سنگین ناشی از آن مطرح گردید، منطقی خواهد بود تا به دنبال راه سبک‌تری جهت کسب فاصله‌ی ماهالانوبیس یک داده تا یک خوش باشیم. راهی که شاید از دقت تمام برخوردار نباشد، اما به لحاظ تخمینی از صحّت بالائی برخوردار باشد. یکی از این راه‌ها آن است تا ماتریس کوواریانس را به صورت قطری درآورده و سپس مانند آن‌چه در مورد الگوریتم BFR جهت محاسبه‌ی فاصله‌ی ماهالانوبیس انجام می‌شود عمل نمائیم. بدین معنی که دیگر نیازی به معکوس‌گیری از ماتریس کوواریانس و انجام محاسبات سنگین مشتمل بر ضرب و جمع نمی‌باشد. هم‌چنین می‌توان زمانی که ابعاد دادگان مربوطه بالاست، فقط آن دسته از ابعاد را در نظر بگیریم که سهم بالائی از اطلاعات و به اصطلاح «عدم قطعیت»<sup>1</sup> کل خوش را دارا هستند. همه‌ی آن‌چه که گفته شد، مقدمه‌ای بود تا ضرورت استفاده از روش‌های کاهش بُعد را مطرح نمائیم. روش کاهش PCA بُعدی که ما در این‌جا از آن بهره خواهیم برداشت، روش محبوب تحلیل مؤلفه‌ی اصلی یا به اختصار می‌باشد. در ادامه، در مورد این روش به اختصار سخن خواهیم گفت.

<sup>1</sup> Uncertainty

پیش از این مطرح گردید که با توجه به ضعف الگوریتم Kmeans در مورد مجموعه‌داده‌های به اصطلاح نویزی، به ناچار باید از یک روش خوشبندی مبتنی بر چگالی استفاده نمائیم که راهکار ما در روش پیشنهادی، استفاده از الگوریتم DBSCAN بود. اما الگوریتم DBSCAN با توجه به این که به اصطلاح «وابسته با پارامتر»<sup>۱</sup> می‌باشد، در صورت دسترسی نداشتن به مقادیر بهینه برای پارامترها به مشکل جدی برخورده و کارائی لازم و کافی را نخواهد داشت. لذا ناچاریم تا به الگوریتم‌های دیگری که قادر به یافتن پارامترهای بهینه می‌باشند متولّش شویم. یکی از این الگوریتم‌ها، الگوریتم تکاملی «بهینه‌سازی انبوه ذرات»<sup>۲</sup> یا به اختصار PSO می‌باشد که در ادامه به اختصار به جزئیات آن و نکاتی که در آن در مورد روش پیشنهادی لحاظ گشته است، اشاره خواهیم نمود.

## ۱.۴ مقدمات و پیش‌زمینه‌های لازم

در این قسمت با توجه به لوازمی که در ابتدای این فصل در مورد آن‌ها بحث گردید، لازم است تا توضیحات مختصری را در مورد هر یک بیان نمائیم.

### ۱.۱.۴ الگوریتم خوشبندی Kmeans

الگوریتم موسوم به Kmeans [۲۲]، یکی از ساده‌ترین نوع «الگوریتم‌های یادگیری بدون نظارت»<sup>۳</sup> است که قادر به حل مسائل خوش‌نام خوشبندی می‌باشد. رویه‌ی این الگوریتم بدین صورت می‌باشد که یک مجموعه‌داده‌ی ارائه‌شده را با توجه به یک فرض اولیه‌ی قوی در مورد تعداد خوش‌های نهائی یا همان  $K$ ، خوشبندی می‌نماید. ایده‌ی اصلی در مورد تعریف موقعیت مراکز خوش‌ها می‌باشد. موقعیت این مراکز باید به نحو زیرکانه‌ای تعریف گردد، چرا که تعیین نامناسب آن‌ها می‌تواند موجب نتایج دور از انتظاری شود. بنابراین، انتخاب بهتر برای این مراکز می‌تواند به این صورت باشد که آن‌ها را تا آن‌جا که ممکن است از یکدیگر دور انتخاب نمائیم. مرحله‌ی بعدی موجود از دادگان فعلی را به

<sup>1</sup> Parametric

<sup>2</sup> Particle Swarm Optimization (PSO)

<sup>3</sup> Unsupervised learning algorithms

نزدیک‌ترین مرکز که نماینده‌ی خوش می‌باشد، انتساب دهیم. پس از بررسی تمامی داده‌ها، مرحله‌ی اول به پایان رسیده و اولین گروه‌بندی انجام شده است. در این مرحله، می‌بایست موقعیت جدید K مرکز را در قالب میانگین داده‌های تعلق‌گرفته به هر خوش در مرحله‌ی قبل محاسبه نمائیم. بعد از تعریف مراکز جدید، وقت آن است تا دوباره فاصله‌ی تمامی داده‌ها را با این مراکز جدید محاسبه نموده و مجدداً هر داده را به نزدیک‌ترین خوش انتساب دهیم. تا این جای کار یک حلقه ایجاد شده است و در انتهای هر بار اجرای حلقه، موقعیت مراکز در حال تغییر خواهد بود. زمانی که تغییر موقعیت این مراکز متوقف گردد، اجرای حلقه نیز به پایان رسیده است.

در نهایت، مقصود این الگوریتم، آن است که «تابع هدف»<sup>۱</sup> زیر را کمینه سازد که در واقع همان «تابع مربع خطأ»<sup>۲</sup> می‌باشد:

$$J = \sum_{j=1}^k \sum_{i=1}^n \|x_i^{(j)} - c_j\|^2 \quad (1.4)$$

به طوری که  $\|x_i^{(j)} - c_j\|^2$  یک معیار فاصله‌ی انتخاب‌شده میان داده‌ی  $x_i^{(j)}$  و میانگین خوش‌هی  $c_j$  بوده و شاخص فاصله‌ی n داده از مراکزی که به آن‌ها تعلق گرفته‌اند می‌باشد.

الگوریتم Kmeans شامل مراحل زیر می‌باشد:

### Kmeans Algorithm

1. Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.
2. Assign each object to the group that has the closest centroid.
3. When all objects have been assigned, recalculate the positions of the K centroids.
4. Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.

اگرچه که می‌توان اثبات نمود که این رویه در نهایت همیشه پایان‌پذیر خواهد بود، اما مسئله آن است که الگوریتم Kmeans به طور قطعی و ضروری، همیشه بهینه‌ترین موقعیت مکانی مراکز را با در نظر گرفتن

<sup>1</sup> Objective function

<sup>2</sup> Squared error function

کمینه‌ی سراسری تابع هدف پیدا نمی‌کند. همین‌طور باید ذکر کرد که این الگوریتم، شدیداً نسبت به موقعیت اولیه‌ی مراکز خوشه‌ها به صورت تصادفی انتخاب می‌شوند، حساس می‌باشد.

#### ۲.۱.۴ معیار فاصله‌ی ماهالانوبیس

فاصله‌ی ماهالانوبیس [۲۳]، یک نوع معیار فاصله مابین نقطه‌ی  $P$  و توزیع آماری  $D$  می‌باشد. فاصله‌ی ماهالانوبیس، یک حالت تعمیم‌یافته‌ی چندبعدی از حالت تک‌بعدی می‌باشد که در آن بررسی می‌کنیم که داده‌ی  $P$ ، به تعداد چند انحراف از معیار از میانگین  $D$  فاصله دارد. مقدار این فاصله در صورتی که داده‌ی  $P$  بر روی میانگین توزیع قرار داشته باشد، برابر صفر می‌باشد و مادامی که این داده از میانگین دور می‌شود، مقدار این فاصله نیز افزایش می‌باید. به عبارتی اگر دادگان توزیع را فاقد همبستگی در راستای ابعاد مختلف در نظر بگیریم، معیار فاصله‌ی ماهالانوبیس در جهت هر کدام از مؤلفه‌های اصلی، تعداد انحراف از معیارها را از  $P$  تا میانگین  $D$  محاسبه می‌نماید. حال اگر هر کدام از این مؤلفه‌های اصلی را به گونه‌ای تغییر مقیاس دهیم تا مقدار واریانس برابر یک واحد گردد، در آن صورت فاصله‌ی ماهالانوبیس در این فضای تبدیل‌یافته معادل با همان فاصله‌ی اقلیدسی استاندارد خواهد بود. بدین ترتیب معیار فاصله‌ی ماهالانوبیس یک معیار به اصطلاح «بدون واحد»<sup>۱</sup> و «بالاتغییر با توجه به مقیاس»<sup>۲</sup> بوده و البته که همبستگی‌های میان ویژگی‌های مجموعه‌داده را نیز در نظر می‌گیرد.

فاصله‌ی ماهالانوبیس یک بردار داده‌ی  $(x_1, \dots, x_p)^T$  =  $\vec{x}$  از نمونه‌داده‌های مربوط به یک توزیع آماری خاص با میانگین  $(\mu_1, \dots, \mu_p)^T$  =  $\vec{\mu}$  و ماتریس کوواریانس  $\Sigma$  به صورت زیر تعریف می‌شود:

$$D_M(\vec{x}) = \sqrt{(\vec{x} - \vec{\mu})^T \Sigma^{-1} (\vec{x} - \vec{\mu})} \quad (2.4)$$

معیار فاصله‌ی ماهالانوبیس هم‌چنین می‌تواند به عنوان یک معیار عدم شباهت میان دو بردار داده‌ی  $\vec{x}$  و  $\vec{y}$  متعلق به یک توزیع یکسان با ماتریس کوواریانس  $\Sigma$  و به صورت زیر نیز تعریف گردد:

$$d(\vec{x}, \vec{y}) = \sqrt{(\vec{x} - \vec{y})^T \Sigma^{-1} (\vec{x} - \vec{y})} \quad (3.4)$$

<sup>1</sup> Unitless

<sup>2</sup> Scale-invariant

حال اگر ماتریس کوواریانس  $\Sigma$ ، یک ماتریس یکه یا همانی  $I$  باشد، در آن صورت فاصله‌ی ماهالانوبیس دقیقاً معادل با همان فاصله‌ی اقلیدسی استاندارد می‌گردد. اگر هم ماتریس کوواریانس توزیع مربوطه به صورت قطری باشد، آن‌گاه معیار فاصله‌ی حاصل را «فاصله‌ی اقلیدسی نرمال‌سازی شده»<sup>۱</sup> گویند که به صورت زیر محاسبه می‌شود:

$$D_M(\vec{x}) = \sqrt{\sum_{i=1}^p \left( \frac{x_i - \mu_i}{\sigma_i} \right)^2} \quad (4.4)$$

به طوری که  $\sigma_i$ ، مقدار انحراف از معیار در بعد  $i$ -ام توزیع  $D$  می‌باشد. اگر هم به عنوان مثال، به ازای بردار داده‌ی  $\vec{x}$  با  $p$  بعد، این گونه تصور کنیم که در راستای هر بعد  $i$ -ام،  $x_i - \mu_i = k\sigma_i$  باشد، در آن صورت، با توجه به (۴.۴)، مقدار فاصله‌ی ماهالانوبیس برای این داده برابر با  $k\sqrt{p}$  خواهد بود. پس با این حساب، مجموعه‌ی نقاطی که فاصله‌ی ماهالانوبیس آن‌ها از میانگین خوش و با توجه به ماتریس کوواریانس آن، برابر با  $k\sqrt{p}$  می‌باشد، به اصطلاح، شاعر همسایگی ماهالانوبیس  $k\sqrt{p}$  آن خوش را شکل می‌دهند. بدین ترتیب مرزهای یک خوش با توزیع نرمال نیز در واقع یک شاعر همسایگی ماهالانوبیس خاص از آن خوش خواهد بود.

لذا با توجه به آن‌چه در مورد مرزهای یک خوش گفته شد، برای کشف داده‌های پرت، خصوصاً از نوع محلی، می‌توان از معیار فاصله‌ی ماهالانوبیس، با توجه «محل خوش»<sup>۲</sup> (میانگین) و «شاکله‌ی»<sup>۳</sup> آن (ماتریس کوواریانس) برداشت.

### ۳.۱.۴ الگوریتم خوشبندی مقیاس‌پذیر BFR

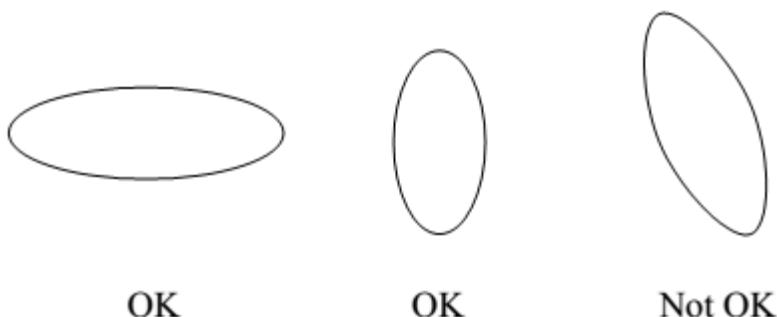
الگوریتم خوشبندی BFR [۸]، که به نام پدیدآورندگان آن نام‌گذاری شده است، یک حالت توسعه‌یافته از الگوریتم Kmeans می‌باشد که برای خوشبندی مجموعه‌داده‌های کلان در فضای اقلیدسی بهینه گشته است. فرض اولیه‌ی قوی این الگوریتم، در مورد شاکله‌ی خوش‌هاست. اول این‌که داده‌های هر

<sup>1</sup> Normalized Euclidean Distance

<sup>2</sup> Location

<sup>3</sup> Shape

خوشه باید به صورت نرمال در اطراف میانگین توزیع شده باشند؛ و دوم این که شاید که مقادیر میانگین و انحراف از معیار به ازای ابعاد مختلف متفاوت باشد، اما خود ابعاد نسبت به یکدیگر مستقل بوده و همبستگی میان دوبه‌دوی آن‌ها صفر می‌باشد. به عنوان مثال، در فضای دو بعدی، یک خوشه، شبیه به یک سیگار خواهد بود، اما این سیگار نباید نسبت به ابعاد اصلی دچار چرخش شده باشد. شکل ۲.۴ وضعیت‌های مناسب و نامناسب یک خوشه‌ی نرمال را نشان می‌دهد.



شکل ۲.۴ فرض اولیه‌ی قوی الگوریتم **BFR** در مورد خوشه‌ها که باید هم توزیع نرمال داشته باشند و هم ابعاد آن‌ها نسبت به یکدیگر همبستگی نداشته باشند. دو خوشه‌ی سمت چپ با این فرض اولیه همخوانی دارند، اما ابعاد خوشه‌ی سمت راست در عین نرمال‌بودن توزیع خوشه، با یکدیگر همبستگی داشته و لذا خوشه‌ی مزبور با فرض اولیه‌ی قوی **BFR** سازگار نمی‌باشد [۲۴].

الگوریتم **BFR**، در ابتدای امر، می‌بایست موقعیت اولیه‌ی مراکز خوشه‌ها را به طرز بهینه‌ای کسب نماید. برای این کار، راه‌های گوناگونی وجود دارد، اما یک راه آن است که از دادگان موجود، نمونه‌برداری انجام داده و سپس الگوریتم Kmeans معمولی را بر روی این نمونه‌داده‌ها اعمال نمائیم. قطعاً در مورد این الگوریتم، با توجه به این که حالت توسعه‌یافته‌ای از الگوریتم Kmeans می‌باشد، باید که تعداد خوشه‌های موجود یا همان مقدار  $K$  را از قبل بدانیم. سپس باید داده‌های موجود را به صورت قطعه‌قطعه از حافظه‌ی جانبی در حافظه‌ی اصلی بارگذاری نمائیم. اندازه‌ی هر قطعه باید به گونه‌ای انتخاب شود که علاوه بر این که قابلیت جاده‌ی در حافظه‌ی اصلی را دارا باشد، فضای کافی برای پردازش آن نیز فراهم باشد. علاوه بر قطعه‌هایی که در حافظه بارگذاری می‌شوند، اطلاعات دیگری از جمله خلاصه‌ی اطلاعات  $K$  تا خوشه و نیز برخی داده‌هایی که بلاتکلیف می‌باشند نیز در حافظه ذخیره می‌گردند. لذا نمی‌توان از کل فضای حافظه جهت ذخیره‌ی یک قطعه استفاده کرد. سایر اطلاعاتی که غیر از هر قطعه در حافظه موجود می‌باشند به شرح ذیل می‌باشد:

۱. **مجموعه‌ی نادیده‌گرفته شده:** «مجموعه‌ی نادیده‌گرفته شده»<sup>۱</sup>، خلاصه‌های ساده‌ای از  $K$  تا خوشه‌ی اصلی می‌باشد. این اطلاعات خلاصه که در واقع ضامن شاکله‌ی خوشه‌ها و بسیار هم ضروری می‌باشند، دور ریخته نخواهند شد. بلکه داده‌هایی که این اطلاعات، نماینده‌ی آن‌ها می‌باشند، نادیده گرفته شده و پس از پردازش، از حافظه‌ی اصلی پاکسازی خواهند شد.

۲. **مجموعه‌ی فشرده شده:** «مجموعه‌ی فشرده شده»<sup>۲</sup>، مانند مجموعه‌ی نادیده‌گرفته شده، خلاصه‌هایی از اطلاعات داده‌هایی می‌باشد که به اندازه‌ی کافی به  $K$  تا خوشه‌ی اصلی نزدیک نبوده‌اند، ولی به اندازه‌ای به یکدیگر نزدیک بوده‌اند که توانسته‌اند تشکیل یک خوشه‌ی موقت دهند. این داده‌ها نیز مانند داده‌های نمایش‌داده شده توسط مجموعه‌ی نادیده‌گرفته شده، از حافظه پاک خواهند شد. چنین خوشه‌هایی که مجموعه‌ی فشرده شده نماینده‌ی آن‌ها می‌باشد را «ریزخوشه»<sup>۳</sup> نامند.

۳. **مجموعه‌ی نگهداری شده:** «مجموعه‌ی نگهداری شده»<sup>۴</sup>، مجموعه‌داده‌هایی می‌باشد که نه می‌توانند به یکی از  $K$  تا خوشه‌ی اصلی تعلق بگیرند و نه هم به تنهایی حائز شرایط تشکیل یک ریزخوشه می‌باشند. این داده‌ها به همان شکل اصلی که در فایل ورودی می‌باشند، در حافظه‌ی اصلی به صورت معلق باقی خواهند ماند.

شکل ۳.۴ نمایی از سه مجموعه‌ی نامبرده شده که پیوسته و تا پایان اجرای الگوریتم، در حافظه حضور دارند را نمایش می‌دهد.

دو مجموعه‌ی نادیده‌گرفته شده و فشرده شده هر کدام با تعداد  $1 + 2d$  مقدار نمایش داده می‌شوند، اگر دادگان ما  $l$ -بعدی باشد. این مقادیر به قرار زیر می‌باشند:

۱. تعداد نقاطی که به صورت خلاصه نمایش داده شده‌اند که با  $N$  نشان داده می‌شود.
۲. مجموع مؤلفه‌های تمام نقاط در هر بعد که در نهایت یک بردار به طول  $d$  می‌باشد که آن را با  $SUM$  نمایش می‌دهیم. مؤلفه‌ی  $i$ -ام این بردار را با  $SUM_i$  نشان می‌دهیم.
۳. مجموع مربعات مؤلفه‌های تمام نقاط در هر بعد که آن هم یک بردار به طول  $d$  بوده و آن را با  $SUMSQ$  نمایش می‌دهیم. مؤلفه‌ی  $i$ -ام این بردار را نیز با  $SUMSQ_i$  نشان می‌دهیم.

هدف واقعی در این الگوریتم، آن است که یک مجموعه از نقاط را به وسیله‌ی تعداد آن‌ها، میانگین و انحراف از معیار آن‌ها در هر بعد بازنمایی کنیم. تعداد  $1 + 2d$  مقدار نیز همگی این اطلاعات مورد نیاز را

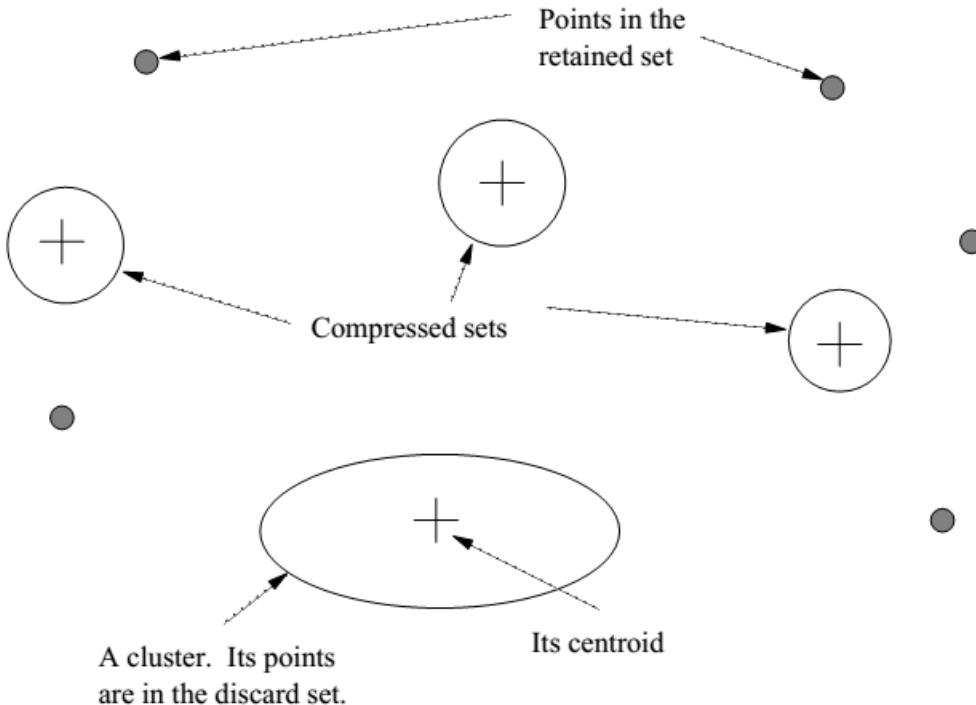
<sup>1</sup> The Discard Set

<sup>2</sup> The Compressed Set

<sup>3</sup> Miniclusters

<sup>4</sup> The Retained Set

در اختیار ما قرار می‌دهند. مقدار میانگین در بُعد  $i$ -ام را با  $SUM_i/N$  و همین‌طور مقدار واریانس در همان بُعد را نیز با  $\frac{SUMSQ_i}{N} - \left(\frac{SUM_i}{N}\right)^2$  نشان می‌دهیم. مقدار انحراف از معیار در بُعد  $i$ -ام نیز برابر جذر واریانس در این بُعد می‌باشد.



شکل ۳.۴ نمایشی از نقاطی که در سه مجموعه‌ی نادیده‌گرفته شده، فشرده شده و نگهداری شده به صورت خلاصه شده یا قطعی قرار دارند [۲۴].

مراحل پردازش یک قطعه در الگوریتم BFR، به صورت زیر می‌باشد:

- در ابتداء، تمامی نقاطی از قطعه که «به اندازه‌ی کافی»<sup>۱</sup> به مرکز یکی از  $K$  تا خوبی اصلی نزدیک هستند، به آن خوبه اضافه می‌شوند. رویه‌ی این اضافه شدن به ازای هر داده به این صورت می‌باشد که تعداد داده‌های تعلق گرفته به خوبه، به میزان یک واحد افزایش یافته و خود بردار داده و مربع آن نیز به ترتیب به بردارهای  $SUMSQ$  و  $SUM$  افزوده می‌گردد. داده‌ی مربوطه پس از بهروزکردن خلاصه‌ی اطلاعات خوبه، نادیده گرفته شده و از حافظه‌ی اصلی پاک می‌شود. این سؤال را که داده‌ی مورد بررسی، به چه مقدار کافی که به خوبه نزدیک باشد، در آن صورت به آن خوبه تعلق خواهد گرفت را در ادامه به اختصار شرح خواهیم داد.
- در مورد تمامی نقاطی از یک قطعه که به اندازه‌ی کافی به هیچ کدام از مراکز اصلی نزدیک نمی‌باشند، آن‌ها را به همراه تمامی نقاطی که از قطعه‌های پیشین در حافظه بلا تکلیف مانده‌اند، خوشه‌بندی می‌کنیم. در اینجا باید

<sup>۱</sup> Sufficiently close

در ابتدا نواحی چگال را که در مرحله‌ی اول به یکی از  $K$  تا خوش‌های اصلی تعلق نگرفته‌اند، به نوعی شناسائی نموده و سپس با توجه به یک شرط محدود‌کننده‌ی «فسردگی»<sup>۱</sup> تعیین نمائیم که آیا حائز شرایط تشکیل یک خوش‌های فشرده‌شده می‌باشد یا خیر. ریزخوش‌های کاندید در این قسمت، با استفاده از الگوریتم Vanilla Kmeans و با در نظر گرفتن تعداد خوش‌های  $K > k_2$  تولید می‌شوند. دلیل بیشتر بودن مقدار  $k_2$  از تعداد خوش‌های اصلی یا  $K$  آن است که این عملیات خوش‌بندی ثانویه بر روی داده‌های متعلق به تمامی خوش‌های اصلی انجام می‌شود. لذا باید به تعداد قطعات ریزتری آن‌ها را شکست تا مطمئن شویم که میانگین حاصل از آن‌ها در محدوده‌ی استحفاظی  $K$  تا خوش‌های اصلی قرار نخواهد گرفت. حال وقت آن است تا بر روی تعداد  $k_2$  ریزخوش‌های حاصل از الگوریتم Vanilla Kmeans، شرط محدود‌کننده‌ی فشردگی را اعمال نمائیم. این شرط، آن است که مقدار بیشینه‌ی واریانس در تمامی ابعاد به ازای یک ریزخوش، نباید بیشتر از مقداری مثلاً برابر با  $\beta$  باشد.

۳. اگر فرض کنیم که تعداد  $k_2 \leq k_3$  ریزخوش‌های حاصل از الگوریتم Vanilla Kmeans، بتوانند از فیلتر  $\beta$  عبور کنند، باید این  $k_3$  ریزخوش را با ریزخوش‌هایی که در پایان قطعه‌های پیشین ایجاد شده‌اند، ترکیب کنیم. این عملیات ترکیب می‌تواند با استفاده از «الگوریتم خوش‌بندی متراکم سلسه‌مراتبی»<sup>۲</sup> صورت پذیرد. بدین صورت که دو نزدیک‌ترین ریزخوش‌هایی که خوش‌های ترکیبی حاصل از آن‌ها، شرط فشردگی  $\beta$  را نقض نکند، می‌توانند با یکدیگر ترکیب شوند. آماره‌ی یک خوش‌های ترکیبی نیز به سادگی از جمع آماره‌های مربوط به ریزخوش‌های پدیدآورنده‌ی آن به دست می‌آید. پس از ترکیب ریزخوش‌ها، خوش‌های حاصل، نگهداری شده و خود ریزخوش‌ها دور ریخته می‌شوند. لازم به ذکر است که تحت هیچ شرایطی، ریزخوش‌ها با هیچ‌کدام از  $K$  تا خوش‌های اصلی ترکیب نخواهند شد.

در اینجا بررسی می‌کنیم که میزان کافی فاصله‌ی ماهالانوبیس جهت انتساب به یک خوش‌ه چگونه تعیین می‌شود. فاصله‌ی ماهالانوبیس، در واقع فاصله‌ی یک داده از میانگین یک خوش‌ه می‌باشد که با استفاده از انحراف از معیار داده‌های خوش در هر بُعد نرمال‌سازی شده است. از آن‌جا که الگوریتم BFR، محورهای اصلی خوش را در راستای محورهای اصلی فضای داده‌ها تصور می‌نماید، لذا در چنین حالتی، محاسبه‌ی فاصله‌ی ماهالانوبیس بسیار ساده خواهد بود. فرض کنیم  $[p_1, \dots, p_d] = p$  یک نقطه‌ی دلخواه در فضا و  $[c_1, \dots, c_d] = c$  نیز مرکز یک خوش‌ه باشد.  $\sigma_i$  را نیز مقدار انحراف از معیار نقاط خوش در بُعد  $i$  در نظر می‌گیریم. فاصله‌ی ماهالانوبیس میان  $p$  و  $c$  به صورت زیر محاسبه می‌شود:

<sup>1</sup> Tightness

<sup>2</sup> Hierarchical agglomerative clustering algorithm

$$\sqrt{\sum_{i=1}^d \left( \frac{p_i - c_i}{\sigma_i} \right)^2} \quad (5.4)$$

همان‌طور که پیداست، ما تفاوت میان  $p$  و  $c$  را در هر بُعد با تقسیم آن بر مقدار انحراف از معیار در آن بُعد، نرمال‌سازی می‌نماییم. مابقی فرمول، تمامی فواصل نرمال‌سازی شده در هر بُعد را با یکدیگر ترکیب می‌نماید تا فاصله‌ی اقلیدسی نرمال‌سازی شده را به دست آورد.

حال برای این‌که یک داده را به یک خوش‌نسبت دهیم، در ابتدا باید فاصله‌ی ماهalanوبیس آن داده را تا تمامی خوش‌ها محاسبه کنیم تا نزدیک‌ترین خوش را بیابیم. سپس برای انتساب این داده به خوش‌ی مربوطه، باید مقدار فاصله‌ی ماهalanوبیس آن از میانگین خوش، از یک مقدار حد آستانه‌ی مشخص، کمتر باشد. به عنوان مثال، فرض کنیم که مقدار حد آستانه برابر با چهار باشد. به این معنی که فاصله‌ی داده از میانگین در هر بُعد برابر با چهار برابر انحراف از معیار در همان بُعد باشد.

#### ۴.۱.۴ الگوریتم خوش‌بندی DBSCAN

«خوش‌بندی فضائی مبتنی بر چگالی برای کاربردهای دارای نویز»<sup>۱</sup> یا به اختصار DBSCAN [۲۱]، یک الگوریتم خوش‌بندی مبتنی بر چگالی می‌باشد. این الگوریتم، نقاطی را که بسیار نزدیک به یکدیگر قرار دارند (نقاطی که همسایگان نزدیک بسیاری دارند) را در قالب یک خوش در نظر گرفته و نقاطی را که به تنها در یک محدوده‌ی تُنک قرار گرفته‌اند (نقاطی که نزدیک‌ترین همسایگان آن‌ها بسیار دور می‌باشند) به عنوان نویز یا داده‌ی پرت معرفی می‌نماید. این الگوریتم دارای دو پارامتر اصلی می‌باشد که عبارتند از  $\epsilon$  یا «بیشینه‌ی شعاع همسایگی یک نقطه» و  $\text{Min}$  یا «حداقل تعداد نقاط لازم جهت تشکیل یک ناحیه‌ی چگال».

<sup>۱</sup> Density-based spatial clustering of applications with noise (DBSCAN)

الگوریتم DBSCAN، دادگان موجود را به سه دسته‌ی اصلی تقسیم می‌نماید. این سه دسته عبارتند از «نقاط هسته‌ای»<sup>۱</sup>، «نقاط (از طریق چگالی-)دسترس پذیر»<sup>۲</sup> و «داده‌های پرت»<sup>۳</sup>. تفسیر این نقاط به قرار زیر می‌باشد:

- یک نقطه‌ی  $p$ ، یک نقطه‌ی هسته‌ای به شمار می‌رود اگر تعداد  $\mu$  نقطه با احتساب خود نقطه، در شعاع همسایگی  $\epsilon$  آن نقطه قرار گرفته باشند. نقاطی که در شعاع همسایگی این نقطه‌ی هسته‌ای قرار می‌گیرند، به اصطلاح «مستقیماً قابل دسترسی از  $p$ »<sup>۴</sup> خواهند بود.
- یک نقطه‌ی  $q$ ، مستقیماً قابل دسترسی از  $p$  خواهد بود اگر نقطه‌ی  $q$  در شعاع همسایگی  $\epsilon$  نقطه‌ی  $p$  قرار داشته و  $p$  نیز یک نقطه‌ی هسته‌ای باشد.
- یک نقطه‌ی  $q$ ، نسبت به  $p$  (از طریق چگالی-)دسترس پذیر خواهد بود اگر یک مسیر از نقاط به صورت  $p_1, p_2, \dots, p_n = q$  باشد و نیز هر  $p_{i+1}$  مستقیماً قابل دسترسی از  $p_i$  باشد. تمامی نقاطی که در این مسیر قرار دارند، باید همگی نقاط هسته‌ای باشند، اما نقطه‌ی  $q$  می‌تواند از این قضیه مستثناء باشد.
- تمامی نقاطی که از هیچ نقطه‌ی دیگری (از طریق چگالی-)دسترس پذیر نمی‌باشند، نقاط نویزی و یا پرت می‌باشند.

حال اگر  $p$  یک نقطه‌ی هسته‌ای باشد، در آن صورت، به همراه تمامی نقاطی (اعم از هسته‌ای یا غیرهسته‌ای) که از آن دسترس پذیر می‌باشند، یک خوش را شکل می‌دهند. هر خوش، قطعاً دارای حداقل یک نقطه‌ی هسته‌ای خواهد بود. نقاط غیر هسته‌ای نیز می‌توانند جزئی از یک خوش باشند، اما «حاشیه‌ی»<sup>۵</sup> خوش را شکل می‌دهند، چرا که نمی‌توان از آن‌ها برای رسیدن به سایر نقاط بهره برد. شکل ۴.۴ یک نمونه از خوش‌بندی حاصل از الگوریتم DBSCAN را با تعداد اندکی نقطه نشان می‌دهد.

<sup>1</sup> Core points

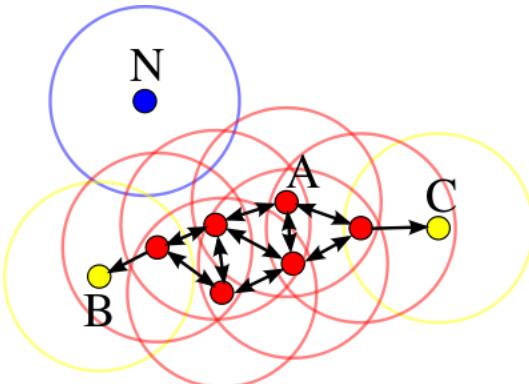
<sup>2</sup> (Density-)reachable points

<sup>3</sup> Outliers

<sup>4</sup> Directly reachable from  $p$

<sup>5</sup> Edge

شکل ۴.۴ در این شکل،  $\mu$  می‌باشد. نقطه‌ی A و سایر نقاط قرمزرنگ، همگی نقاط هسته‌ای می‌باشند، چرا که محدوده‌ی پوشش‌دهنده‌ی این نقاط، با یک شعاع همسایگی ε و با احتساب خود نقاط، حداقل دارای  $\mu$  نقطه می‌باشد. با توجه به این‌که این نقاط همگی از یکدیگر دسترس‌پذیر می‌باشند، تشکیل یک خوشه‌ی منفرد را می‌دهند. نقاط B و C، نقاط هسته‌ای نمی‌باشند، از طریق نقاط هسته‌ای دیگر از A دسترس‌پذیر می‌باشند و در نتیجه به خوشه‌ی تعلق خواهند داشت. این نقاط، نقاط حاشیه‌ای خوشه به حساب می‌آیند. نقطه‌ی N یک نقطه‌ی نویزی و یا پرت به حساب می‌آید، چرا که نه یک نقطه‌ی هسته‌ای به شمار آمده و نه از هیچ نقطه‌ی دیگری مستقیماً دسترس‌پذیر می‌باشد [۲۵].



حال اگر دادگان D را در قالب یک گراف همسایگی G و با ماتریس مجاورت A در نظر بگیریم، به گونه‌ای که درایه‌ی  $a_{ij}$  این ماتریس مجاورت به ازای داده‌های i و j برابر با یک باشد، به شرطی که هر دو در شعاع همسایگی ε یکدیگر قرار گیرند، آن‌گاه می‌توان الگوریتم DBSCAN را به صورت زیر ارائه نمائیم:

### DBSCAN Algorithm

1. Find the ε (eps) neighbors of every point, and identify the core points with more than or equal to  $\mu$  neighbors.
2. Find the connected components of core points on the neighborhood graph, ignoring all non-core points.
3. Assign each non-core point to a nearby cluster if the cluster is an ε (eps) neighbor, otherwise assign it to noise.

یک پیاده‌سازی ساده‌لوحانه از این الگوریتم ارائه شده که مختصر نیز می‌باشد، نیازمند آن خواهد بود تا ماتریس مجاورت گراف همسایگی را در مرحله‌ی اول، به صورت کامل در اختیار داشته باشیم که این مسئله، نیازمند مقدار قابل توجهی حافظه خواهد بود. الگوریتم اصلی DBSCAN، نیازمند این مقدار حافظه در ابتدای امر نمی‌باشد، چرا که مرحله‌ی اول را به ازای هر نقطه به صورت جداگانه انجام می‌دهد.

## ۵.۱.۴ روش کاهش بعد تحلیل مؤلفه‌ی اصلی (PCA)

«تحلیل مؤلفه‌ی اصلی»<sup>۱</sup> یا به اختصار PCA [۲۶]، یک رویه‌ی آماری است که از یک «تبديل متعامد»<sup>۲</sup> برای انتقال داده‌هایی که مقادیر ویژگی‌های آن‌ها احتمالاً با یکدیگر همبستگی دارند، به فضائی استفاده می‌نماید که در آن مقادیر ویژگی‌ها «به صورت خطی ناهمبسته»<sup>۳</sup> می‌باشند. این ویژگی‌های ناهمبسته را «مؤلفه‌های اصلی»<sup>۴</sup> نامند. تعداد مؤلفه‌های اصلی متمایز از یکدیگر، برابر با مقدار کوچک‌تر تعداد ابعاد اصلی و یا تعداد داده‌های موجود منهای یک می‌باشد. خروجی این تبدیل، به این گونه می‌باشد که اولین مؤلفه‌ی اصلی، دارای بیشترین مقدار واریانس می‌باشد و به عبارتی بیشترین مقدار «تغییرپذیری»<sup>۵</sup> در دادگان را به خود اختصاص داده است. مؤلفه‌های اصلی دیگر که به دنبال مؤلفه‌ی اصلی اول می‌آیند، به ترتیب هر کدام بیشترین مقدار واریانس ممکن را به خود اختصاص داده‌اند و قطعاً هر مؤلفه‌ی اصلی، بر مؤلفه‌های اصلی که به ترتیب قبل از آن می‌آیند، متعامد می‌باشد. بردارهای متعامدی که به این ترتیب حاصل می‌شوند، در صورتی که به ترتیب نزولی واریانس در هر مؤلفه‌ی اصلی در کنار یکدیگر قرار گیرند، یک «مجموعه‌ی پایه‌ی متعامد»<sup>۶</sup> غیر همبسته را شکل خواهند داد.

<sup>1</sup> Principle Component Analysis (PCA)

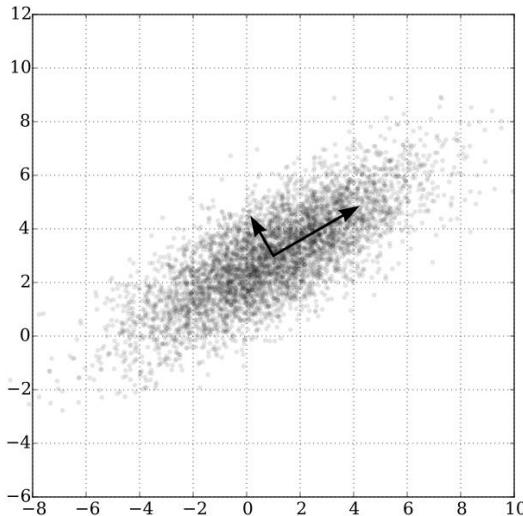
<sup>2</sup> Orthogonal transformation

<sup>3</sup> Linearly uncorrelated

<sup>4</sup> Principle components

<sup>5</sup> Variability

<sup>6</sup> Orthogonal basis set



شکل ۵.۴ ۵.۴ اعمال PCA بر روی یک «توزیع گاوسین چندمتغیره»<sup>۱</sup> که مرکز آن بر روی نقطه‌ی (۱,۳) قرار دارد. مقدار انحراف از معیار در راستای مؤلفه‌ی اول که در جهت تقریبی (۰.۸۶۶, ۰.۵) قرار دارد، برابر با ۳ و در راستای مؤلفه‌ی اصلی دوم که عمود بر مؤلفه‌ی اصلی اول می‌باشد نیز برابر با یک است. بردارهای نشان‌داده شده، همان بردارهای ویژه ماتریس کوواریانس توزیع می‌باشند که با استفاده از ریشه‌ی دوم (جذر) مقادیر ویژه مربوطه، تقسیم مقیاس داده شده‌اند. دنباله‌ی بردارها نیز تغییر مکان داده شده‌اند، به گونه‌ای که بر روی میانگین قرار گیرند [۲۷].

از PCA می‌توان در قالب یک رویه‌ی کاهش بُعد نیز بهره برد. بدین‌ترتیب که اگر تعدادی از مؤلفه‌های اصلی برتر را به گونه‌ای انتخاب کنیم که مجموع مقادیر واریانس به ازای این مؤلفه‌های برتر، نرخ بالائی (مثلاً ۸۰ یا حتی ۹۰ درصد) از واریانس کل (انرژی کل) را به خود اختصاص داده باشند، آن‌گاه از کنار هم قرار گرفتن بردارهای مربوطه، یک ماتریس تبدیل به دست خواهد آمد. از این ماتریس تبدیل، می‌توان برای بردن داده‌ها به یک فضای با ابعاد کمتر و دارای بیشترین تغییرپذیری ممکن استفاده نمود.

الگوریتم زیر، رویه‌ی کاهش بُعد با استفاده از PCA را به اختصار و به صورت شبه کد نشان می‌دهد:

#### *Recipe for Dimensionality Reduction with PCA*

**Input:** Dataset  $X_{n*p} = \{\vec{x}_1, \dots, \vec{x}_n\}$ , as  $n$  is the cardinality of  $X$  and  $p$  is its dimensionality, and each  $\vec{x}_i$  is a row vector of length  $p$ . We wish to use PCA to reduce dimensionality to  $t$ .

**Output:** The desired matrix  $Y_{n*t}$ , which is the closest approximation to  $X$ .

- 1: Find the sample mean  $\hat{\mu} = \frac{1}{n} \sum_{i=1}^n \vec{x}_i$ .
- 2: Subtract the sample mean from the data as  $z_i = x_i - \hat{\mu}$ .
- 3: Compute the scatter matrix  $S = \sum_{i=1}^n z_i^t z_i$ .
- 4: Compute the Eigenvectors  $e_1, \dots, e_t$ , corresponding to the  $t$  largest Eigenvalues of  $S$  in a descending order.
- 5: Let  $e_1, \dots, e_t$  be the columns of transformation matrix  $A$ .
- 6: The desired matrix  $Y_{n*t}$  which is the closest approximation to  $X$ , is calculated through  $Y = XA$ .

<sup>1</sup> Multivariate Gaussian distribution

## ۶.۱.۴ الگوریتم بهینه‌سازی انبوه ذرات (PSO)

الگوریتم «بهینه‌سازی انبوه ذرات»<sup>۱</sup> یا به اختصار PSO [۲۸]، یک الگوریتم تکاملی است که برای بهینه‌سازی یک مسئله با استفاده از تلاش مکرر برای بهبود یک «راحل کاندید»<sup>۲</sup> و با توجه به یک معیار ارزیابی ارائه شده عمل می‌نماید. این الگوریتم، یک مسئله را به این ترتیب حل می‌کند که در ابتدا یک جمعیت اولیه از راه‌حل‌های کاندید را در نظر می‌گیرد که در اینجا تحت عنوان «ذرات»<sup>۳</sup> از آن‌ها یاد می‌کنیم، و سپس این ذرات را در محدوده‌ی «فضای جستجو»<sup>۴</sup> و با استفاده از یک فرمولاسیون ساده‌ی ریاضیاتی که با توجه به «موقعیت»<sup>۵</sup> و «سرعت»<sup>۶</sup> ذرات تعریف می‌شود، حرکت می‌دهد. حرکت هر کدام از ذرات، تحت تأثیر دو چیز خواهد بود، اول، بهترین موقعیت محلی شناخته‌شده به ازای آن ذره و دوم، بهترین موقعیت سراسری شناخته‌شده با توجه به ذرات و در فضای جستجو. هر دوی این بهترین موقعیت‌های محلی و سراسری، در طول تکرارهای متوالی، با توجه به موقعیت‌های بهتری که توسط ذرات به دست می‌آیند، بهروز خواهند شد. با این رویه، انتظار آن است که جمعیت راه‌حل‌ها را به سمت بهترین راه‌حل ممکن سوق دهیم، اما نمی‌توان با قاطعیت تضمین کرد که این راه‌حل بهینه یافت خواهد شد.

اگر تابع هزینه‌ای که باید کمینه شود را به صورت  $f: R^n \rightarrow R$  در نظر بگیریم، این تابع، یک راه‌حل کاندید را به عنوان آرگومان ورودی و در قالب یک بردار از اعداد حقیقی دریافت نموده و یک عدد حقیقی را به عنوان خروجی بر می‌گرداند که همان مقدار تابع هدف به ازای راه‌حل کاندید می‌باشد. در این‌جا، به تابع مشتق تابع هدف، نیازی نمی‌باشد. بلکه هدف، آن است که یک راه‌حل بهینه‌ی سراسری  $a$  یافت شود، به طوری که به ازای هر راه‌حل دلخواه  $b$  در سراسر فضای جستجو،  $f(b) \leq f(a)$  باشد. همین‌طور می‌توان در مورد تابع هدفی که باید مقدار آن بیشینه گردد، از تابع  $-f = h$  استفاده کرده و مطابق رویه‌ای که قید شد، عمل نمائیم.

<sup>۱</sup> Particle Swarm Optimization (PSO)

<sup>۲</sup> Candidate solution

<sup>۳</sup> Particles

<sup>۴</sup> Search-space

<sup>۵</sup> Position

<sup>۶</sup> Velocity

فرض کنیم  $S$  برابر تعداد ذرات جمعیت موجود باشد که موقعیت آن در فضای جستجو و سرعت آن را به ترتیب با  $v_i \in R^n$  و  $x_i \in R^n$  نشان می‌دهیم. اگر  $p_i$  نشان گر بهترین موقعیت محلی به ازای ذره‌ی  $i$ <sup>۱</sup> و  $g$  نیز نمایان گر بهترین موقعیت سراسری در میان تمامی ذرات باشد، آن‌گاه می‌توانیم الگوریتم PSO پایه را به صورت زیر ارائه نمائیم:

### Basic PSO Algorithm

```

for each particle  $i = 1, \dots, S$  do
    Initialize the particle's position with a uniformly distributed random vector:  $x_i \sim U(b_{lo}, b_{up})$ 
    Initialize the particle's best known position to its initial position:  $p_i \leftarrow x_i$ 
    if  $f(p_i) < f(g)$  then
        update the swarm's best known position:  $g \leftarrow p_i$ 
    end if
    Initialize the particle's velocity:  $v_i \sim U(-|b_{up}-b_{lo}|, |b_{up}-b_{lo}|)$ 
end for
while a termination criterion is not met do:
    for each particle  $i = 1, \dots, S$  do
        for each dimension  $d = 1, \dots, n$  do
            Pick random numbers:  $r_p, r_g \sim U(0,1)$ 
            Update the particle's velocity:  $v_{i,d} \leftarrow \omega v_{i,d} + \varphi_p r_p (p_{i,d} - x_{i,d}) + \varphi_g r_g (g_d - x_{i,d})$ 
        end for
        Update the particle's position:  $x_i \leftarrow x_i + v_i$ 
        if  $f(x_i) < f(p_i)$  then
            Update the particle's best known position:  $p_i \leftarrow x_i$ 
        if  $f(p_i) < f(g)$  then
            Update the swarm's best known position:  $g \leftarrow p_i$ 
        end if
    end if
    end for
end while

```

مقادیر  $b_{lo}$  و  $b_{up}$ ، به ترتیب برابر کران‌های پایین و بالای فضای جستجو می‌باشند. «شرط پایان»<sup>۱</sup>، می‌تواند یا برابر تعداد تکرارهای مجاز الگوریتم جهت یافتن راه حل بهینه باشد، و یا هم زمانی که مقدار خروجی تابع هدف، برابر با یک مقدار مشخص و کافی گردد، تکرار الگوریتم، متوقف شود. مقادیر پارامترهای  $\omega$ ،  $\varphi_p$  و  $\varphi_g$  توسط متخصص مربوطه تعیین شده و سبب کنترل رفتار و اثرگذاری روش PSO می‌شوند.

<sup>۱</sup> Termination criterion

## ۲.۴ روش پیشنهادی

در این قسمت، قصد داریم تا روش پیشنهادی را به تفصیل و با ذکر جزئیات معرفی نمائیم. همان‌طور که پیش از این نیز مطرح گردید، روش پیشنهادی ما در واقع حالت توسعه یافته‌تری از الگوریتم BFR می‌باشد. در روش BFR، جهت مکان‌یابی موقعیت اولیه‌ی خوشه‌ها (که البته تعداد آن‌ها را باید از قبل بدانیم) می‌توانیم از یک نمونه‌برداری اولیه استفاده کرده و آن را به صورت بهینه خوشه‌بندی نمائیم. بدین ترتیب، موقعیت اولیه‌ی مراکز خوشه‌ها با یک صحت نسبی حاصل خواهد شد. البته حجم نمونه‌برداری نیز در میزان صحّت تعیین این مراکز اولیه نقش جدی خواهد داشت، که میزان این حجم هم به نوعی به اندازه‌ی حافظه‌ی اصلی یا همان RAM مرتبط خواهد بود. در روش پیشنهادی ما نیز، در ابتدای امر باید برای این‌که از موقعیت اولیه‌ی خوشه‌ها اطلاع حاصل کنیم، نیاز داریم تا یک نمونه‌برداری اولیه از دادگان ورودی انجام داده و عملیات خوشه‌بندی را با الگوریتم DBSCAN آغاز نمائیم. همان‌طور که پیداست در روش پیشنهادی ما، در شروع کار و به هنگام خوشه‌بندی داده‌های نمونه‌برداری شده، نیازی به دانستن تعداد خوشه‌ها از پیش نمی‌باشد. چرا که الگوریتم DBSCAN مبتنی بر چگالی بوده و تمرکز آن بر روی بهبود موقعیت مکانی مراکز خوشه‌هایی که دانستن تعداد آن‌ها از قبل ضروری است، نمی‌باشد. پس از عملیات خوشه‌بندی روی داده‌های نمونه‌برداری شده، کافی است تا یک سری اطلاعات لازم مربوط به هر یک از خوشه‌ها را جمع‌آوری کنیم و سپس RAM را از هر گونه داده‌ی نمونه‌برداری شده پاکسازی نمائیم. در مرحله‌ی بعدی باید مجموعه‌داده‌ی ورودی را قطعه‌قطعه نماییم، به گونه‌ای که در هر بار پردازش یک قطعه، حافظه‌ی اصلی، هم گنجایش جاده‌ی آن قطعه را دارا باشد و هم از فضای کافی برای محاسبات لازم برخوردار باشد. در هر بار پردازش یک قطعه، اطلاعات مدل خوشه‌بندی موقت، به روز می‌شود. بدین معنی که یا اطلاعات خوشه‌های سابق دستخوش تغییر شده و یا هم اطلاعات خوشه‌های جدید به مدل فعلی افزوده می‌شود. پس از پردازش آخرین قطعه نیز هر آن‌چه جز اطلاعات مربوط به مدل خوشه‌بندی موقت در حافظه قرار دارد، از RAM پاک خواهد شد. حال زمان آن است تا از مدل خوشه‌بندی موقت استفاده کرده و با استفاده از پارامترهای مناسب، مدل خوشه‌بندی نهائی را بسازیم. در آخرین مرحله، پس از حصول مدل خوشه‌بندی نهائی که مشتمل بر مراکز خوشه‌ها و ماتریس‌های کوواریانس آن‌هاست، کافی است تا فاصله‌ی ماهalanobis هر داده را تا همگی خوشه‌ها

حساب کنیم تا نزدیکترین خوش به آن داده را بیابیم. داده‌ی مربوطه، به نزدیکترین خوش تعلق گرفته و فاصله‌ی ماهalanobis آن تا خوشی مربوطه را به عنوان امتیاز پرتبودن آن داده در نظر می‌گیریم.

#### **Algorithm 1. Sampling + Scalable Clustering + Scoring**

**Input:** The  $n$  by  $p$  matrix  $X$ , PC's total variance ratio  $\lambda$ ,  $k$ -Sigma membership  $\alpha$ ,  $k$ -Sigma pruning  $\beta$ , Sampling rate  $\eta$ , Sampling Epsilon coefficient  $C_\varepsilon$ , Sampling MinPts coefficient  $C_\mu$ .

**Output:** The outlierness score and ranking for each point in  $X$ .

#### **Sampling:**

Step 1. Take a sample of  $X$  according to the sampling rate  $\eta$ .

Step 2. Run an arbitrary algorithm to find the best parameters for the DBSCAN algorithm, for clustering the sampled data. We call these two parameters  $\varepsilon_{\text{samp}}$  and  $\mu_{\text{samp}}$ . Here we use the PSO algorithm.

Step 3. Cluster the sampled data using the obtained optimal parameters and ignore those clusters that their covariance matrix is not positive semi-definite.

Step 4. Make the very first clusters' information array (clustering model) from step 3 according to Algorithm 2.

Step 5. Assign the least covariance determinant value of the initial clusters to  $\delta$ , as the maximum Permitted Covariance Determinant (PCD) condition.

Step 6. Clear the main memory (RAM) of any sampled data and keep the initial information gained from earlier steps.

#### **Scalable Clustering:**

Prepare data to be processed chunk by chunk, as each chunk could be fit and processed in RAM buffer at the same time.

Step 1. Drag next available chunk from data into the RAM.

Step 2. Update the current model of clustering over the contents of buffer according to Algorithm 3.

Step 3. If there is any available unprocessed chunk, go to step 1.

Step 4. Build the final clusters' structure according to Algorithm 6, using the temporary model of clustering obtained from earlier steps.

#### **Scoring:**

According to final clustering model, for each data point  $q \in X$ , use the Mahalanobis distance criterion to find the closest cluster, and finally assign  $q$  to that cluster and use the criterion's value as the point's outlierness score.

الگوریتم ۱، رویه‌ی کلی روش پیشنهادی را که مشتمل بر سه مرحله‌ی اصلی و متوالی نمونه‌برداری، خوشبندی مقیاس‌پذیر؛ و امتیازدهی می‌باشد، نشان می‌دهد. در ادامه، جزئیات هر یک از مراحل اصلی را به تفصیل شرح خواهیم داد.

## ۱.۲.۴ نمونهبرداری

در این مرحله، می‌بایست به صورت کاملاً «تصادفی»<sup>۱</sup> از تمامی دادگان موجود، اقدام به نمونهبرداری کنیم. این مسئله در ابتدا ساده به نظر می‌رسد، اما در واقعیت می‌تواند یک مسئله‌ی بفرنج باشد. چرا که در برخی مجموعه‌داده‌ها امکان این موضوع وجود دارد که داده‌های موجود با توجه به یک ویژگی یا خاصیت مشخصی مرتب شده باشند. این مسئله سبب می‌شود تا در زمان نمونهبرداری، نتوانیم از داده‌هایی که در نواحی چگال خوشها (عمدتاً همان داده‌های حوالی مرکز ثقل خوشها) قرار دارند، به اندازه‌ی کافی نمونه اخذ کنیم. این مسئله سبب می‌شود تا در زمان اعمال خوشبندی بر روی داده‌های نمونهبرداری شده، مراکز حاصله از خوشها با مراکز خوشها اصلی مجموعه‌داده، قرابت کافی نداشته باشند و در نتیجه این امکان وجود خواهد داشت که داده‌های پرت در طول رویه‌ی خوشبندی، به این خوشها اولیه و ناقص جذب شده و در نهایت منجر به بالارفتن خطای محاسباتی گردد. در چنین شرایطی که دادگان ما از یک ترتیب خاص برخوردار بوده و کار نمونهبرداری را دچار مشکل می‌کنند، تهیه‌ی یک نمونه‌ی کاملاً تصادفی از مجموعه‌داده می‌تواند به قیمت بررسی کل داده‌های موجود تمام گردد. جهت رفع این مشکل، راه حل‌هایی وجود دارد که یکی از آن‌ها می‌تواند اخذ نمونه از اطلاعات مربوط به اولین بارگیری‌های حافظه از داده‌ها، با استفاده از روشی خاص باشد. در اینجا، فرض ما بر آن است که مجموعه‌داده‌ی کلان ما از یک ترتیب خاص برخوردار نبوده و با اولین بارگیری حافظه، نمونه‌های لازم از مجموعه‌داده اخذ شده‌اند. لازم به ذکر است که پس از اخذ نمونه از کلان داده، باید نرخ این نمونهبرداری را که به عنوان مثال می‌تواند برابر  $0.5$  درصد باشد، برای استفاده در مراحل بعدی ذخیره نمائیم. ما در اینجا این نرخ را با نماد یونانی  $\pi$  نشان خواهیم داد. در ضمن، در این روش پیشنهادی مانند الگوریتم BFR، تصور ما بر آن است که خوشها ای که در زمان نمونهبرداری حاصل شده‌اند، تعداد آن‌ها به شرط حصول پارامترهای بهینه، برابر با تعداد خوشها واقعی یا  $K$  در دادگان اصلی بوده و البته موقعیت میانگین‌ها و البته ماتریس‌های کوواریانس آن‌ها نیز با خوشها ای اصلی قرابت بالائی دارد. در الگوریتم BFR نیز فرض بر آن است که در عین دانستن مقدار  $K$ ، میانگین‌هایی که از مرحله‌ی نمونهبرداری حاصل شده‌اند، با میانگین‌های اصلی همخوانی نسبتاً زیادی دارند.

<sup>1</sup> Random

همان طور که پیش از این قید شد، الگوریتم DBSCAN که در مرحله‌ی نمونه‌برداری مورد استفاده قرار می‌گیرد، نیاز به دارابودن مقادیر بهینه به ازای دو پارامتر  $\epsilon$  و  $M$  می‌باشد. الگوریتم‌های بسیاری جهت یافتن مقادیر بهینه برای الگوریتم‌هایی مانند DBSCAN وجود دارند که یکی از آن‌ها الگوریتم تکاملی PSO می‌باشد. ما نیز در روش پیشنهادی از PSO جهت یافتن مقادیر بهینه به ازای  $\epsilon$  و  $M$  استفاده می‌کنیم. اما مسئله آن است که در زمان نمونه‌برداری، داده‌ها نسبت به حالت اصلی مجموعه‌داده، به اصطلاح «پخش شده‌تر»<sup>۱</sup> می‌باشند. لذا در این حالت، مقادیر بهینه‌ای که به ازای دو پارامتر مذبور به دست آمده‌اند، به طور مستقیم قابل استفاده در مورد دادگان اصلی نمی‌باشند و باید از ضریبی از آن‌ها در زمان اعمال الگوریتم DBSCAN در مراحل بعد از نمونه‌برداری استفاده کنیم. ذکر این نکته ضروری است که مقدار پارامتر  $\epsilon$  نسبت به مقدار پارامتر  $M$  شدیداً حساس‌تر می‌باشد. به طوری که با تغییر اندکی در مقدار پارامتر  $\epsilon$ ، شاهد تغییرات جدی در ساختار خوشه‌بندی خواهیم بود، اما در مورد پارامتر  $M$  این‌گونه نمی‌باشد. از آن‌جا که چگالی مجموعه‌داده در زمان نمونه‌برداری بسیار کمتر از میزان چگالی در مورد دادگان اصلی می‌باشد، لذا ما نیز از ضرایبی مابین صفر و یک به ازای هر یک از دو پارامتر حاصله در زمان نمونه‌برداری، برای استفاده در مراحل بعدی بهره خواهیم برد. این ضرایب را به ازای  $\epsilon$  و  $M$ ، به ترتیب با  $C_\epsilon$  و  $C_M$  نشان می‌دهیم. در این‌جا، پارامترهای الگوریتم DBSCAN، حاصله از نمونه‌برداری را به ترتیب با  $\epsilon_{sample}$  و  $M_{sample}$  نشان می‌دهیم. مقادیر ضرایب مابین صفر و یک که پیش از این مطرح شد را نیز می‌توانیم از کاربر دریافت کنیم، ولی با توجه به آزمایشات انجام‌شده، مقادیر بهینه به ازای این ضرایب به صورت  $0.9 = C_M$  و  $0.5 = C_\epsilon$  به دست آمده‌اند.

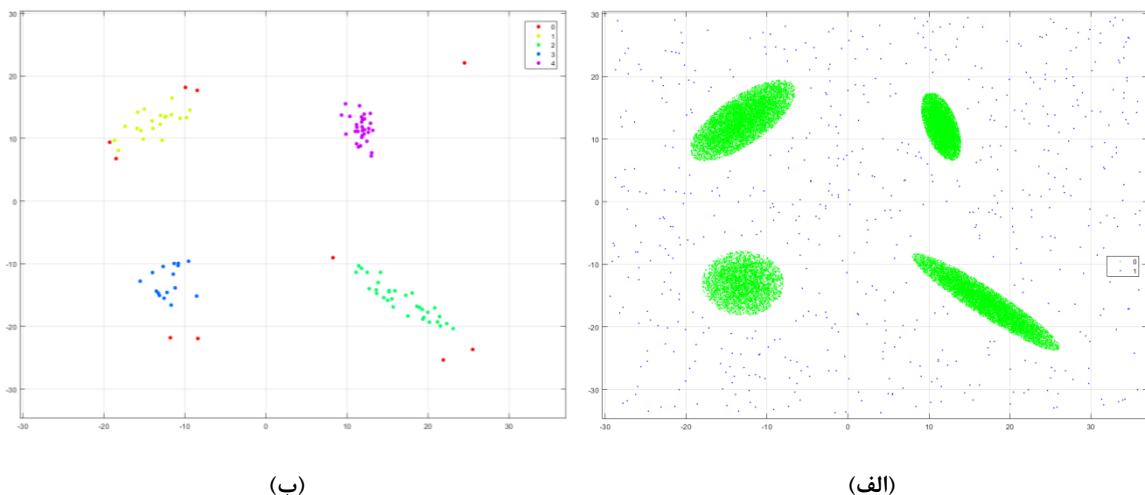
در این‌جا باید در مورد تمامی خوشه‌هایی که چه در مرحله‌ی نمونه‌برداری و چه در مراحل بعد از آن ایجاد می‌شوند، این نکته را متذکر شویم که چنان‌چه ماتریس کوواریانس این خوشه‌ها به اصطلاح «مبیت و نیمه‌قطعی»<sup>۲</sup> نباشد، فاصله‌ی مahaانویسیس هر داده‌ی دلخواه از چنین خوشه‌هایی معتبر نبوده و با احتمال بالایی یک «عدد مختلط»<sup>۳</sup> خواهد بود. لذا ما نیز در صورت روبه‌روشدن با چنین خوشه‌هایی، آن‌ها را حائز شرایط لازم ندانسته و نادیده می‌گیریم. بدین‌ترتیب با نادیده‌گرفتن این خوشه‌ها، از

<sup>1</sup> More scattered

<sup>2</sup> Positive semi-definite

<sup>3</sup> Complex number

پیش‌آمدن خطاهای مُهلك احتمالی، جلوگیری به عمل خواهیم آورد. شکل ۶.۴ الف، یک مجموعه‌داده‌ی دو بعدی با چهار عدد خوش را نشان می‌دهد که از تعدادی داده‌ی پرت نیز تشکیل شده است. شکل ۶.۴ ب نیز داده‌های نمونه‌برداری شده با نرخ نمونه‌برداری برابر با نیم درصد را نشان می‌دهد. داده‌هایی که در این شکل، با رنگ قرمز مشخص شده‌اند، داده‌هایی می‌باشند که توسط الگوریتم DBSCAN، به عنوان نویز شناسائی شده‌اند. این گونه داده‌ها می‌توانند واقعاً همان داده‌های نویزی و یا پرت موجود در مجموعه‌داده‌ی اصلی باشند و یا هم می‌توانند داده‌هایی باشند که در طول رویه‌ی خوش‌بندی، حائز شرایط لازم برای تشکیل یک خوش نبوده و در نتیجه مردود اعلام شده‌اند.



شکل ۶.۴ الف) یک مجموعه‌داده‌ی دو بعدی با تعداد چهار خوش که حاوی تعدادی داده‌ی پرت و نویزی نیز می‌باشد. ب) نمونه‌برداری حاصل از این مجموعه‌داده با نرخ نمونه‌برداری برابر با نیم درصد؛ داده‌های قرمزرنگ، داده‌هایی هستند که در مرحله‌ی خوش‌بندی به عنوان نویز شناسائی شده‌اند و یا هم رویه‌ی حائز شرایط لازم جهت تشکیل خوش نبوده‌اند.

حال زمان آن است تا از خوش‌های اولیه‌ی حاصله از مرحله‌ی نمونه‌برداری، اقدام به ساخت مدل آغازین خوش‌بندی نمائیم. به همین منظور، باید از هر کدام از خوش‌های کشف شده با استفاده از الگوریتم DBSCAN، یک سری اطلاعات را اخذ نموده و به ازای آن خوش در یک آرایه‌ی مخصوص ذخیره نمائیم. از آن جا که در مورد داده‌های با ابعاد بالا، حجم محاسبات در مورد فاصله‌ی ماهالانوبیس زیاد خواهد بود، می‌توانیم مطابق آن‌چه که پیش‌تر مطرح شد، از یک روش کاهش بُعد به ازای هر خوش استفاده نماییم. در این جا ما از روش کاهش بُعد محبوب PCA استفاده می‌کنیم. اما در مورد این روش، برای این که بتوانیم تعداد مؤلفه‌های اصلی‌ای که سهم بالاتری در بازنمایی خوش دارند و به بیان دیگر، میزان اطلاعات و عدم قطعیت بیشتری از خوش را دارا می‌باشند یافت نمائیم، می‌توانیم در ابتدا

مؤلفه‌های اصلی را با توجه به مقادیر واریانس در هر مؤلفه‌ی و به صورت یک لیست نزولی مرتب نمائیم. سپس کافی است تا آن تعداد مؤلفه‌های اصلی را از ابتدای این لیست انتخاب کنیم که نسبت درصدی جمع واریانس این مؤلفه‌های به واریانس کل (که برابر جمع واریانس‌ها به ازای تمامی مؤلفه‌های اصلی می‌باشد) بزرگتر از مثلاً  $80\%$  و یا حتی  $90\%$  درصد باشد. در اینجا ما این مؤلفه‌های اصلی را مؤلفه‌های برتر نامیده و تعداد آن‌ها را با  $p'$  نشان می‌دهیم. اگر  $[x_i]_{p \times 1}$ , یک داده متعلق به خوش‌هی موقت  $[C_t]_{n \times p}$  باشد، آن‌گاه اطلاعات مربوط به این خوش‌هی که در آرایه‌ی مخصوص  $\{C_{inf}\}_{8 \times K}$  مربوط به خوش‌های موقت ذخیره می‌گردد، به شرح ذیل می‌باشد:

- بردار میانگین خوش‌هی در فضای اصلی:  $\mu_{C_t} = \frac{1}{n} \sum_{i=1}^n x_i$
- «ماتریس پراکندگی»<sup>1</sup> خوش‌هی در فضای اصلی  $S_{C_t} = \sum_{i=1}^n (x_i - \mu_{C_t})(x_i - \mu_{C_t})^t$
- تعداد  $p'$  بردار ویژه‌ی برتر  $[e_1, \dots, e_{p'}]$ , حاصل از ماتریس کواریانس خوش‌هی  $(\Sigma_{C_t}) = \frac{1}{n-1} S_{C_t}$  که ستون‌های ماتریس تبدیل  $A_{C_t}$  شکل می‌دهند؛
- میانگین در فضای تبدیل یافته:  $\mu'_{C_t} = A_{C_t} \mu_{C_t}$
- «ریشه‌ی دوم»<sup>2</sup> تعداد  $p'$  مقدار ویژه‌ی برتر یا همان واریانس در مؤلفه‌های برتر:  $[\sqrt{\lambda_1}, \dots, \sqrt{\lambda_{p'}}]$
- تعداد داده‌هایی که تاکنون به خوش‌هی تعلق گرفته‌اند:  $n$
- مقدار عددی  $p'$ :
- مقدار عددی دترمینان ماتریس کواریانس خوش‌هی:  $\det_{\Sigma_{C_t}}$

الگوریتم ۲، روند کسب و اضافه‌نمودن این اطلاعات را به ازای هر خوش‌هی، به آرایه‌ی اطلاعات خوش‌های موقت نشان می‌دهد. از این الگوریتم، در مراحل بعدی و در زمان ساخته‌شدن خوش‌های جدید نیز استفاده خواهد شد.

#### **Algorithm 2. Updating the clusters' information array**

**Input:** The current  $8$  by  $K$  array of clusters' information  $C_{inf}$ ,

Clusters array  $\{X_1, \dots, X_K\}$ , PC's total variance ratio  $\lambda$ .

**Output:** The updated clustering model.

**for each** cluster  $X_i$ , add following data to the end of  $C_{inf}$

Apply PCA on  $X_i$  and obtain its principle components' coefficients and variances.  
Then choose  $p'$  as the number of those top  $p'$  principle components' variances,

<sup>1</sup> Scatter matrix

<sup>2</sup> Square root

which their share of total variance is at least  $\lambda$ .

$C_{inf}\{1,end+1\}$  = mean vector of  $X_i$

$C_{inf}\{2,end+1\}$  = Scatter matrix of  $X_i$

$C_{inf}\{3,end+1\}$  = Top  $p'$  pc coefficients corresponding top  $p'$  pc variances

$C_{inf}\{4,end+1\}$  = The transformed mean vector or  $C_{inf}\{1,end+1\} \times C_{inf}\{3,end+1\}$

$C_{inf}\{5,end+1\}$  = Square root of top  $p'$  pc variances

$C_{inf}\{6,end+1\}$  = Number of data points in  $X_i$

$C_{inf}\{7,end+1\}$  = Value of  $p'$

$C_{inf}\{8,end+1\}$  = Covariance determinant of  $X_i$

**end for**

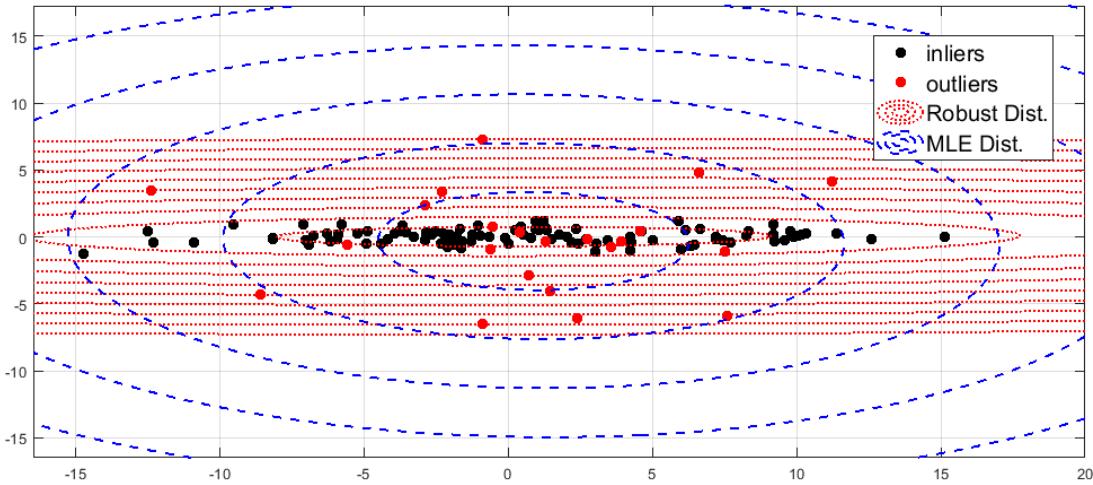
با توجه به [۲۹، ۳۰]، می‌توان اثبات نمود که در صورتی که یک توزیع گاووسین چندمتغیره، توسط تعدادی داده‌ی پرت، آلوده گردد، آن‌گاه دترمینان ماتریس کوواریانس چنین توزیع آلوده‌شده‌ای، نسبت به دترمینان ماتریس کوواریانس اصلی به مقدار قابل توجهی بیشتر بوده و به اصطلاح، دیگر «مستحکم»<sup>۱</sup> نخواهد بود. به دنبال این آلودگی، «منحنی‌های تراز»<sup>۲</sup> فاصله‌ی ماهalanobis نیز نسبت به حالت قبل از آلودگی، عریض‌تر می‌شوند تا بتوانند تمامی داده‌ها اعم از نرمال و غیرنرمال را پوشش دهند. پس می‌توان مطرح نمود که میان دترمینان ماتریس کوواریانس یک خوشه با میزان عریض‌بودن منحنی‌های تراز ماهalanobis آن، یک ارتباط مستقیم وجود دارد. لذا با توجه به این‌که در روش پیشنهادی، داده‌ها به مرور زمان می‌آیند، ممکن است که یک خوشه، به هر دلیلی تعدادی داده‌ی پرت را به عضویت پذیرفته و به دنبال آن و با بهروزشدن اطلاعات حیاتی خوشه، ماتریس کوواریانس خوشه از حالت مستحکم خارج گردد. ادامه‌ی این رویه می‌تواند سبب شود تا داده‌های پرت بیشتری به مرور زمان به چنین خوشه‌ای جذب شده و به همین ترتیب، منحنی‌های تراز ماهalanobis خوشه نیز عریض و عریض‌تر شوند. لذا باید با استفاده از یک شرط محدود‌کننده که هدف اصلی آن، کنترل مرزهای مستحکم خوشه است، از جذب داده‌های پرت به خوشه‌های نرمال جلوگیری نمائیم. این مسئله که داده‌های پرت، به مرور زمان، جذب خوشه‌ها شده و دیگر با استفاده از معیار سنتی فاصله‌ی ماهalanobis، قادر به شناسائی نمی‌باشد را اصطلاحاً «اثر پوشش»<sup>۳</sup> نامند. شکل ۷.۴، نمونه‌ای از یک توزیع گاووسین دومتغیره را نشان می‌دهد که با تعدادی داده‌ی پرت آلوده شده است. نقاط مشکی، داده‌های نرمال خوشه و نقاط قرمزرنگ نیز داده‌های

<sup>1</sup> Robust

<sup>2</sup> Mahalanobis contour lines

<sup>3</sup> Masking effect

پرت را نشان می‌دهند. منحنی‌های تراز مربوط به هر دو حالت اصلی و آلوده شده نیز به ترتیب با نقطه‌چین و خط‌چین نمایش داده شده‌اند. پر واضح است که مرزهای مستحکم مربوط به خوش‌های اصلی، کم‌عرض‌تر از مرزهای مربوط به خوش‌های آلوده شده می‌باشند. منحنی‌های تراز ماهالانوبیس خوش‌های آلوده شده، تلاش می‌کنند تا تمامی داده‌های موجود، اعم از نرمال و غیرنرمال را پوشش دهند.



شکل ۷.۴ نمونه‌ای از یک توزیع گاووسین دومتغیره (نقاط مشکی) که با تعدادی داده‌ی پرت (نقاط قرمزرنگ) آلوده شده است. منحنی‌های تراز ماهالانوبیس که با نقطه‌چین نشان داده شده‌اند، نمایان‌گر مرزهای مستحکم خوش‌های اصلی می‌باشند و منحنی‌های تراز که با خط‌چین نشان داده شده‌اند نیز نمایان‌گر مرزهای خوش‌های آلوده شده می‌باشند.

در این روش پیشنهادی، ما مقدار شرط مزبور جهت کنترل مرزهای خوش را برابر کمینه‌ی دترمینان‌های ماتریس‌های کواریانس مربوط به خوش‌هایی قرار می‌دهیم که در مرحله‌ی نمونه‌برداری یافت شده‌اند. خوش‌هایی که در گذر زمان تولید خواهند شد، نباید دترمینان ماتریس کواریانس آن‌ها از مقدار این شرط اولیه بیشتر باشد. ما تا پایان امر خوش‌بندی مقیاس‌پذیر، کماکان از همین شرط برای کنترل اندازه‌ی خوش‌های جدید بهره خواهیم برداشت.

در این قسمت، عملیات مربوط به نمونه‌برداری اول پایان می‌یابد و می‌بایست به جز اطلاعات اولیه‌ای که در مورد خوش‌ها به دست آورده‌ایم، هر آن‌چه از داده‌های نمونه‌برداری شده در RAM موجود است را پاکسازی نمائیم.

## ۲.۲.۴ خوشبندی مقیاس‌پذیر

در این مرحله، باید مجموعه‌داده‌ی ورودی را به صورت قطعه‌قطعه بررسی نمائیم. به گونه‌ای که حافظه‌ی اصلی به ازای هر قطعه، در آن واحد هم از فضای کافی جهت قرارگیری آن و هم از فضای آزاد جهت پردازش آن بهره‌مند باشد. قطعه‌ها باید یکی پس از دیگری در حافظه بارگذاری شده و پردازش شوند.

### ۱.۲.۲.۴ به روزرسانی مدل خوشبندی موقت با توجه به یک قطعه‌ی داده

الگوریتم ۳، مراحل به روزرسانی مدل خوشبندی موقت را با توجه به یک قطعه‌ی داده نشان می‌دهد.

#### **Algorithm 3. Updating the clustering model using a chunk of main data**

**Input:** A chunk of data  $Y$ , Clusters' information array  $C_{inf}$ ,  $k$ -Sigma membership  $\alpha$ .

**Output:** The updated clustering model.

**for each** data point  $q \in Y$

Find the closest cluster to  $q$  that can accept it as a member and update that cluster's information, according to Algorithm 4, and finally remove  $q$  from buffer.

**else if** there is no true cluster

Retain  $q$  in RAM buffer to be decided on later.

**end for**

After finishing process of the whole chunk, normalize the scatter matrix of each cluster  $C_i$ , according to size of the cluster as  $\left(\frac{1}{C_{inf}\{6, C_i\}-1}\right) \times C_{inf}\{2, C_i\}$ , to obtain the covariance matrix  $\Sigma_i$ . Now, apply PCA on  $\Sigma_i$  to acquire its eigenvalues and eigenvectors. Then, update the information of each cluster as follows:

- $C_{inf}\{7, C_i\}$  = The value of  $p'$  as the number of those top pc's variances, which their share of total variance is at least  $\lambda$ ;
- $C_{inf}\{3, C_i\}$  = Top  $p'$  eigenvectors corresponding to the top  $p'$  eigenvalues;
- $C_{inf}\{4, C_i\}$  = The transformed mean vector or  $C_{inf}\{1, C_i\} \times C_{inf}\{3, C_i\}$ ;
- $C_{inf}\{5, C_i\}$  = Square root of top  $p'$  pc variances.

**for each** data point  $o$  in RAM

Find the closest cluster to  $o$  that can accept it as a member and update that cluster's information, according to Algorithm 4, and finally remove  $o$  from buffer.

**else if** there is no true cluster

Retain  $o$  in RAM buffer to be decided on later.

**end for**

Cluster the data retained in RAM according to Algorithm 5.

**if** this was the last chunk that has been processed, do:

**for each** data point  $o$  in RAM

```

Find the closest cluster to o that can accept it as a member and update
that cluster's information, according to Algorithm 4, and finally remove
o from buffer.

else if there is no true cluster
    Mark o as a temporary outlier and remove it from buffer.
end for
end if

```

## ۱.۱.۲.۲.۴ بررسی امکان تعلق داده‌های یک قطعه به خوش‌های موقت

پس از بارگذاری هر قطعه از داده‌ها در حافظه، باید به ازای هر یک از داده‌های قطعه بررسی نمائیم که آیا در شعاع همسایگی قابل قبول یک خوش‌ه قرار می‌گیرند یا خیر. برای این کار باید فاصله‌ی ماهالانوبیس داده‌ی مربوطه را تا همگی خوش‌های محاسبه نمائیم و سپس از این میان، آن خوش‌های را برگزینیم که داده‌ی ما به آن نزدیک‌تر است. حال برای این خوش‌ه نیز باید بررسی نمائیم که آیا فاصله‌ی داده‌ی مربوطه از این خوش‌ه، کوچک‌تر یا مساوی مقدار حد آستانه‌ی تعیین‌شده از قبل می‌باشد یا خیر. در این صورت داده‌ی فعلی به این خوش‌ه تعلق گرفته و اطلاعات خوش‌ه را با توجه به آن به‌روز می‌نمائیم. این به‌روزرسانی شامل اضافه‌کردن ضرب خارجی بردار داده‌ی مربوطه در خودش به ماتریس پراکندگی خوش‌ه و نیز افزایش تعداد داده‌های تعلق‌گرفته به خوش‌ه تاکنون به اندازه‌ی یک واحد می‌باشد. پس از آن، باید داده‌ی بررسی‌شده را از حافظه‌ی اصلی پاک نمائیم تا فضای بی‌موردی را به خود اختصاص ندهد. اگر خوش‌هی مناسبی برای این داده یافت نشد، می‌بایست در حافظه معلق بماند تا بعدا در مورد آن و سایر داده‌های معلق تصمیم‌گیری شود. الگوریتم ۴، مراحل یافتن نزدیک‌ترین خوش‌ه در صورت وجود و نیز به‌روزرسانی اطلاعات آن خوش‌ه را نشان می‌دهد.

### **Algorithm 4. Finding the true cluster and updating its information**

**Input:** Data point  $q$ , Clusters' information array  $C_{inf}$ ,  $k$ -Sigma membership  $\alpha$ .

**Output:** The true cluster if exists and updating its information according to  $q$ .

*Find the closest cluster to  $q$  according to Mahalanobis distance criterion. Call this cluster  $C_{best}$ .*

*if the distance of  $q$  to  $C_{best}$  is less than or equal to  $\alpha \times \sqrt{C_{inf}[7, C_{best}]}$ , do:*

- Add the outer product of column vector  $q$  to  $C_{inf}[2, C_{best}]$ ;
- Increase the value of  $C_{inf}[6, C_{best}]$  by 1.

**else**

*Return NULL.*

**end if**

## ۲،۱،۲،۲،۴ به روزرسانی اطلاعات حیاتی خوشه‌های موقت

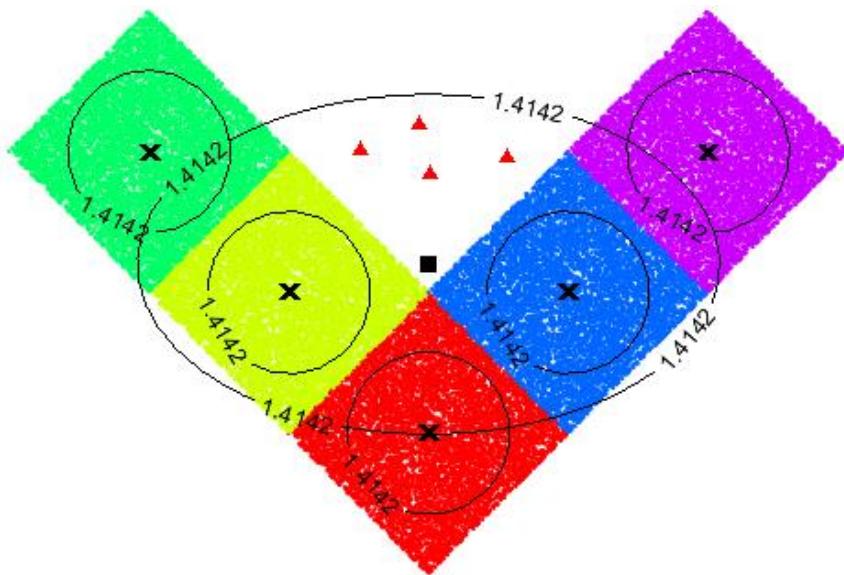
در اینجا پس از آن که تمامی داده‌های یک قطعه به لحاظ تعلق‌گرفتن و یا نگرفتن به خوشه‌ها مورد بررسی قرار گرفتند، با توجه به الگوریتم ۳، وقت آن است تا اطلاعات حیاتی هر کدام از خوشه‌های موقت موجود را به روزرسانی نمائیم. منظور از این اطلاعات حیاتی، همان خصیصه‌هایی از هر خوشه می‌باشد که با استفاده از آن‌ها می‌توانیم فاصله‌ی ماهالاتوبیس هر داده‌ی دلخواه از این خوشه را در فضای تبدیل‌یافته‌ی آن خوشه محاسبه نمائیم. بدین‌وسیله می‌بایست در ابتدا با استفاده از تعداد داده‌هایی که تاکنون به خوشه تعلق گرفته‌اند، ماتریس پراکندگی خوشه را به اصطلاح نرمال‌سازی نمائیم تا ماتریس کوواریانس به روزشده‌ی خوشه حاصل گردد. در اینجا با اعمال PCA بر روی ماتریس کوواریانس به روزشده، مقادیر ویژه و بردارهای ویژه‌ی جدید را به دست می‌آوریم و همان‌گونه که در زمان ایجاد خوشه عمل نمودیم، آن تعداد از مؤلفه‌های اصلی برتر را بر می‌گزینیم که سهم بیشتری را در تعریف واریانس کل به خود اختصاص داده‌اند. بدین‌ترتیب ماتریس تبدیل خوشه و نیز بردار میانگین در فضای تبدیل‌یافته و همین‌طور مقادیر واریانس به ازای هر مؤلفه‌ی اصلی را به روز می‌نماییم.

## ۳،۱،۲،۲،۴ بررسی امکان تعلق داده‌های معلق موجود در RAM به خوشه‌های به روزشده

پس از آن که اطلاعات هر خوشه را در پایان پردازش قطعه به روز کردیم، باید دید که آیا داده‌هایی که تاکنون در حافظه، چه از این قطعه و چه از قطعه‌های پیشین معلق مانده‌اند، می‌توانند به خوشه‌هایی که اطلاعات آن‌ها جدیداً به روز شده‌اند تعلق بگیرند یا خیر. به همین منظور، مانند آن‌چه که در ابتدای پردازش هر قطعه عمل نمودیم، با توجه به الگوریتم ۴، بررسی می‌نماییم که آیا هر کدام از این داده‌های معلق می‌توانند به خوشه‌ای تعلق بگیرند یا خیر. در صورت تعلق‌گرفتن به یک خوشه، اطلاعات آن خوشه دستخوش تغییر شده و داده‌ی مربوطه نیز از حافظه پاک خواهد شد؛ و در صورت عدم تعلق‌گرفتن به یک خوشه، باید باز هم همچنان در حافظه معلق بماند تا در مراحل بعدی در مورد آن و سایر داده‌های بلا تکلیف مشابه آن تصمیم‌گیری شود.

## ۴,۱,۲,۲,۴ خوشبندی داده‌های معلق موجود در RAM

حال در ادامه‌ی الگوریتم ۳، زمان آن رسیده است تا در مورد داده‌هایی که در حافظه بلاکلیف مانده‌اند، بررسی کنیم که آیا می‌توانند روی هم رفته حائز شرایط لازم برای تشکیل خوشبندی باشند یا خیر. بدین‌ترتیب همان‌طور که پیش از این نیز قید شد، باید از یک الگوریتم خوشبندی مبتنی بر چگالی برای خوشبندی این داده‌ها استفاده نمائیم تا از انتساب داده‌های پرت موجود احتمالی به خوشبندی جلوگیری به عمل آورد. چرا که انتظار ما آن است که خوشبندی‌ها در نواحی چگال تشکیل خواهند شد و داده‌های پرت طبعاً از چنین نواحی‌ای فاصله قابل توجهی دارند. مزیت دیگر استفاده از الگوریتم‌های خوشبندی مبتنی بر چگالی در چنین مواقعي در آن است که دیگر نيازی نمی‌باشد تا نسبت به تعداد خوشبندی‌ها آگاهی داشته باشیم. در این روش پیشنهادی، ما از همان الگوریتم DBSCAN که در مرحله‌ی نمونه‌برداری نیز ذکر گردید، به دلیل تفاوت موجود میان چگالی داده‌ها در زمان نمونه‌برداری و چگالی داده‌ها در دادگان اصلی، می‌بایست از ضرایبی مابین صفر و یک برای پارامترهای الگوریتم DBSCAN استفاده کنیم. پارامترهای DBSCAN در زمان نمونه‌برداری را با  $\epsilon_{samp}$  و  $\mu_{samp}$  نشان داده و ضرایبی را که ذکر شد نیز به ترتیب با  $C_\epsilon$  و  $C_\mu$  نشان می‌دهیم. پس از اعمال الگوریتم DBSCAN بر روی داده‌های بلاکلیف موجود در حافظه، در ابتدا باید در مورد هر کدام از خوشبندی‌های جدید یافت شده بررسی نمود که آیا ماتریس کوواریانس داده‌های خوشبندی مثبت و نیمه‌قطعی می‌باشد یا خیر. در صورتی که این شرط لازم برای داده‌های یک خوشبندی برقرار نباشد، خوشبندی مذبور را با توجه به دلیلی که پیش از این در مورد مختلطشدن فاصله‌ی ماهالانوبیس قید شد، نادیده می‌گیریم تا داده‌های آن مجدداً در حافظه معلق بمانند و بعداً در کنار داده‌هایی که در ادامه‌ی الگوریتم و بررسی قطعه‌های دیگر به این مجموعه‌ی معلق اضافه می‌گردد، در مورد آن‌ها تصمیم‌گیری شود.



شکل ۸.۴ نمونه‌ای از یک خوشی غیرمحب که از زیرخوشی‌های محدب و کوچک‌تر تشکیل شده است. مرکز ثقل خوشی غیرمحب با علامت مربع مشکی و مرکز ثقل خوشی‌های محدب کوچک‌تر نیز با علامت ضربدر مشکی نشان داده شده‌اند. منحنی تراز ماهالانوبیس بزرگ‌تر، مربوط به خوشی غیرمحب و بزرگ بوده و سایر منحنی‌های تراز نیز مربوط به خوشی‌های محدب کوچک‌تر می‌باشند. نقاط مثلثی‌شکل و قرمزرنگ نیز نقاط کاندید پرت می‌باشند.

اما همان‌طور که در قسمت نمونه‌برداری نیز در مورد ضرورت استفاده از یک شرط محدود‌کننده جهت جلوگیری از بزرگ‌شدن بی‌رویه‌ی اندازه‌های خوشی‌هایی که به مرور زمان تشکیل می‌شوند، صحبت به میان آمد، در این‌جا نیز با یک مثال شهودی دیگر نشان می‌دهیم که در صورت تشکیل یک خوشی ناموزون و یا به بیان بهتر، غیر محدب، چه اتفاقات دور از انتظاری ممکن است رخ دهد. شکل ۸.۴ یک خوشی ناموزون متشکل از ۵ خوشی کوچک‌تر را نشان می‌دهد که هر یک از ۵ خوشی با یک رنگ مجزا نشان داده شده است. همگی این خوشی‌های کوچک‌تر به لحاظ تعداد و چینش داده‌ها با یکدیگر یکسان می‌باشند و تنها تفاوت موجود، در موقعیت مکانی آن‌ها می‌باشد که در نهایت موجب تشکیل خوشی غیرمحب بزرگ‌تر می‌شوند. مرکز ثقل خوشی غیر محدب، با علامت مربع نشان داده شده است و همان‌گونه که پیداست، در خارج از محدوده‌ی داده‌ها قرار دارد. مرکز ثقل سایر خوشی‌های کوچک‌تر نیز با علامت ضربدر نشان داده شده و البته که کاملاً در محدوده‌ی داده‌های هر کدام از خوشی‌ها واقع شده است. منحنی تراز فاصله‌ی ماهالانوبیس با مقدار مشخص  $\sqrt{2} \times 1$  نیز به ازای هر کدام از خوشی‌ها، چه محدب و چه غیرمحب رسم شده است. پر واضح است که منحنی تراز خوشی ناموزون، نسبت به سایر خوشی‌های کوچک‌تر بسیار عریض‌تر می‌باشد و در نتیجه اگر داده‌ی پرتی در نزدیکی خوشی قرار داشته

باشد، می‌تواند به دلیل قرارگرفتن در محدوده‌ی استحفاظی این خوش، به آن تعلق گرفته و جدای از این که میزان خطا در شناسائی را افزایش می‌دهد، در شکل‌گیری ساختار خوش نیز نقش مؤثری ایفا خواهد نمود. نمونه‌ی چنین داده‌های پرتوی در شکل با نقاط مثلث‌شکل و قرمزنگ نشان داده شده‌اند.

مقدار دترمینان ماتریس کوواریانس خوش‌های غیرمحدب برابر با  $0,3945$  و دترمینان ماتریس کوواریانس هر یک از خوش‌های کوچک‌تر نیز برابر با  $0,0071$  می‌باشد. همان‌طور که قبل از این نیز در مرحله‌ی نمونه‌برداری اشاره گردید، در صورت اضافه‌شدن تعدادی داده‌ی پرت به یک خوش، دترمینان ماتریس کوواریانس آن افزایش قابل توجهی داشته و منحنی‌های تراز ماهالاتوبیس خوش نیز عریض‌تر خواهند شد. رخداد مکرر این مسئله به مرور زمان، می‌تواند سبب شود تا داده‌های پرت بیشتری و یا حتی داده‌های نرمال خوش‌های دیگر به این خوش جذب شوند و سبب افزایش خطای محاسباتی گردند. پس با توجه به همه‌ی این نکات، ضروری می‌نماید که در صورتی که دترمینان ماتریس کوواریانس یک خوش که به مرور آمدن داده‌ها تشکیل شده است، از یک مقدار مشخص و مجاز که در اینجا آن را با علامت  $\delta$  نشان می‌دهیم، تجاوز نماید، اقدام به شکستن این خوش به زیرخوش‌هایی با اندازه‌ی کوچک‌تر نمایم، به گونه‌ای که دترمینان ماتریس کوواریانس هر یک از این زیرخوش‌های، کوچک‌تر و یا مساوی مقدار مجاز دترمینان باشد. شکل ۸.۴ نمونه‌ای از چنین حالتی می‌باشد که در آن می‌توان یک خوش‌های غیرمحدب را به زیرخوش‌های محدب و با اندازه و دترمینان کوچک‌تر شکست تا به دنبال آن، خطر جذب داده‌های پرت محلی کاهش یابد.

با توجه به همه‌ی مطالبی که تا این‌جای کار مطرح گردید، ضروری است تا پس از اعمال الگوریتم DBSCAN بر روی داده‌های معلق در RAM و کسب خوش‌های جدید، به ازای هر یک از این خوش‌های، این مسئله را نیز بررسی نماییم که آیا مقدار دترمینان ماتریس کوواریانس خوش از مقدار مجاز دترمینان تجاوز نموده است یا خیر. در صورتی که این مقدار دترمینان، بیشتر از حد مجاز باشد، باید با استفاده از یک الگوریتم خوش‌بندی دیگر، آن خوش را به زیرخوش‌های کوچک‌تر تا آن‌جایی بشکنیم که دترمینان ماتریس کوواریانس هر کدام از این زیرخوش‌ها از مقدار مجاز دترمینان تجاوز ننماید. در این روش پیشنهادی، ما از الگوریتم Kmeans جهت شکستن چنین خوش‌های ناموزونی به زیرخوش‌های کوچک‌تر استفاده می‌کنیم. لازم به ذکر است که یک شرط لازم و مقدماتی دیگر نیز باید به ازای هر یک از این زیرخوش‌ها برقرار باشد و آن همان است که ماتریس کوواریانس هر یک از این زیرخوش‌ها می‌بایست

ثبت و نیمه قطعی باشد. در صورتی که الگوریتم Kmeans نتواند خوشی کشف شده توسط الگوریتم DBSCAN را به زیرخوشه های کوچک تری که حائز شرایط لازم باشند تقسیم نماید، در آن صورت باز هم مثل سابق عمل کرده و خوشی مربوطه را مردود اعلام می نماییم تا بعدا در کنار سایر داده های معلق موجود در RAM، در مورد آنها تصمیم بگیریم. در غیر این صورت، به ازای هر یک از زیرخوشه های معرفی شده توسط Kmeans، اطلاعات مدل خوشبندی موقت را با توجه به الگوریتم ۲ به روز می نماییم. همین طور در صورتی که خوشی کشف شده توسط DBSCAN نیز حائز همگی شرایط لازم، اعم از ثبت و نیمه قطعی بودن ماتریس کوواریانس آن و نیز کوچک تر بودن مقدار دترمینان این ماتریس کوواریانس از شرط  $\delta$  باشد، دیگر نیازی به استفاده از Kmeans نبوده و کافی است تا مانند قبل، اطلاعات مدل خوشبندی موقت را با توجه به این خوشی به روز نماییم. الگوریتم ۵، تمامی مراحل لازم جهت خوشبندی داده های معلق موجود در RAM را به ترتیب نشان می دهد.

#### **Algorithm 5. Clustering data points retained in RAM buffer**

**Input:** Retained set RS, PCD  $\delta$ , Sampling Epsilon coefficient  $C_\epsilon$ , Sampling MinPts coefficient  $C_\mu$ , DBSCAN sampling Epsilon  $\epsilon_{samp}$ , DBSCAN sampling MinPts  $\mu_{samp}$ .

**Output:** The updated clustering model.

Apply DBSCAN algorithm to cluster RS according to two parameters  $C_\epsilon \times \epsilon_{samp}$  for Epsilon and  $C_\mu \times \mu_{samp}$  for MinPts. Consider the result of this clustering as  $\{X_1, \dots, X_K\}$ .

**for each**  $X_i$

**if** covariance matrix of  $X_i$  is not positive semi-definite

Reject  $X_i$  from being a unique cluster and retain its data points in RAM buffer to be decided on later.

**else if** covariance determinant of  $X_i$  is more than  $\delta$

For the second time, we afford to use a clustering algorithm to break  $X_i$  to smaller clusters with smaller covariance determinants.

Here, we choose the K-means algorithm as it can cluster a set of data points into arbitrary number of clusters. For the acceptable value of K, two conditions must be met as follows:

- $\Sigma_{X_{i,j}}$  shall be positive semiDefinite, for  $j = 1, \dots, K'$ ,
- $|\Sigma_{X_{i,j}}| \leq \delta$ , for  $j = 1, \dots, K'$ .

as  $\Sigma_{X_{i,j}}$  and  $|\Sigma_{X_{i,j}}|$  respectively mean the covariance matrix and the covariance determinant of sub-cluster  $X_{i,j}$ .

**if** such  $K' \geq 2$  could be found

Break the cluster  $X_i$  to sub-clusters and update the array of clusters' information due to partition  $\{X_{i,1}, \dots, X_{i,K'}\}$  according

```

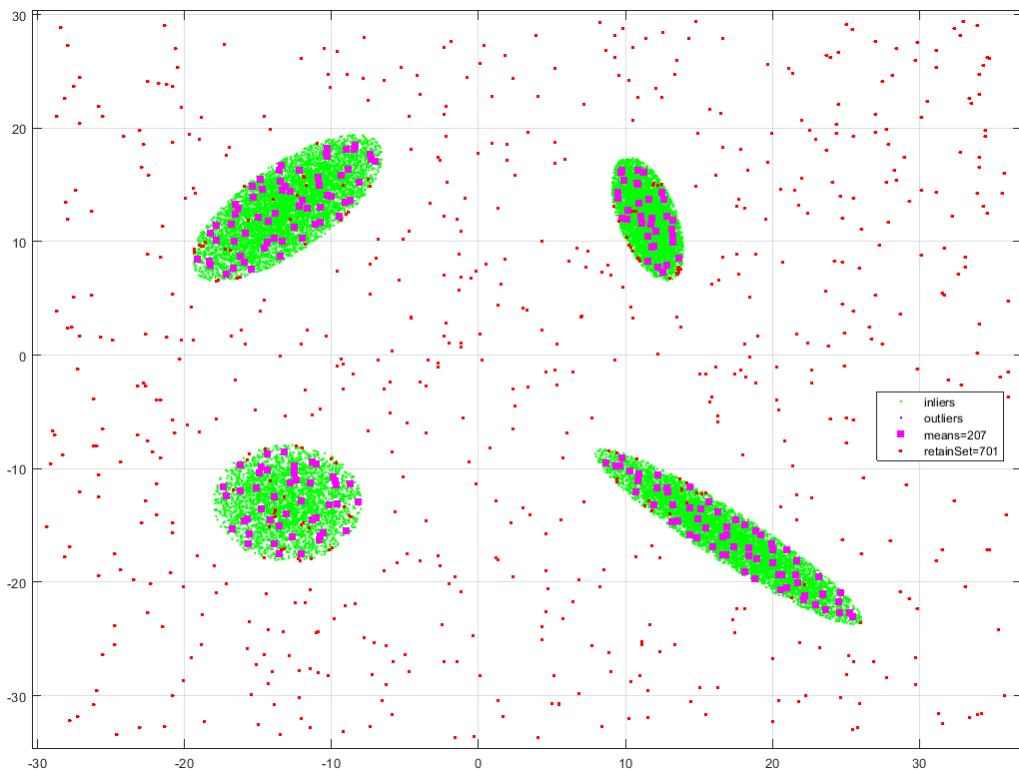
    to Algorithm 2.
else
    Reject  $X_i$  from being a unique cluster and retain its data points
    in RAM buffer to be decided on later.
end if
else
    Update the array of clusters' information according to  $X_i$  according to
    Algorithm 2.
end if
end for

```

## ۵.۱.۲.۲.۴ بررسی امکان تعلق داده‌های آخرین قطعه به خوشه‌های موقت

حال پس از آن که آخرین قطعه از داده‌ها نیز با توجه به مراحل الگوریتم ۳ پردازش شد، یک بار دیگر نیز هر یک از داده‌های معلق موجود در RAM را مورد بررسی قرار می‌دهیم که آیا می‌توانند به یکی از خوشه‌های موقت تعلق بگیرند یا خیر، و در صورت تعلق گرفتن، باز هم اطلاعات خوشه‌ی مربوطه دچار تغییر شده و داده‌ی تعلق گرفته نیز از حافظه پاک خواهد شد، و در صورت عدم تعلق گیری به یک خوشه، به عنوان داده‌ی پرت موقت شناسائی خواهد شد. ذکر این نکته ضروری است که چه داده‌هایی که به خوشه‌های موقت تعلق می‌گیرند و چه داده‌هایی که در نهایت امر و پس از پردازش کلیه‌ی قطعه‌ها، در حافظه بلا تکلیف مانده و به عنوان داده‌های پرت موقت شناسائی می‌شوند، همگی از حافظه‌ی اصلی پاک شده و تنها این اطلاعات خوشه‌های موقت و یا همان مدل خوشه‌بندی موقت است که در حافظه باقی می‌ماند.

شکل ۹.۴ وضعیت نهائی مدل خوشه‌بندی موقت را پس از بررسی کلیه‌ی قطعه‌ها در مورد همان مجموعه داده‌ی شکل ۶.۴ الف را نشان می‌دهد. نقاط مربعی‌شکل، همان میانگین‌های خوشه‌های موقت می‌باشند که همان‌طور که پیداست همگی بر طبق انتظار، در نواحی چگال یا همان خوشه‌ها قرار گرفته‌اند. نقاط سبزرنگ، نمایان‌گر داده‌های نرمال و نقاط قرمزرنگ نیز داده‌هایی می‌باشند که توسط مدل پیشنهادی، چه به درست و چه به اشتباه به عنوان داده‌ی پرت شناسائی شده‌اند. تمامی این نقاط در پایان پردازش قطعه‌ها از حافظه‌ی اصلی پاک خواهند شد. سایر اطلاعات مربوط به خوشه‌های موقت نیز در قالب یک آرایه در حافظه‌ی اصلی ذخیره شده است.



شکل ۹.۴ مدل خوشه‌بندی موقت مربوط به شکل ۶.۴ الف که در قالب میانگین‌ها و با نقاط مربعی‌شکل نشان داده شده است. سایر نقاط سبزرنگ و قرمزرنگ نیز به ترتیب داده‌های نرمال و داده‌های شناسائی شده به عنوان داده‌ی پرت می‌باشند. تمامی این نقاط، پس از پردازش تمامی قطعه‌ها از حافظه‌ی اصلی پاک شده و تنها این مدل خوشه‌بندی موقت خواهد بود که در حافظه باقی می‌ماند.

#### ۲.۲.۲.۴ ساخت مدل خوشه‌بندی نهائی

در ادامه و پس از آن که پردازش قطعه‌ها به پایان رسید، زمان آن رسیده است تا با استفاده از مدل خوشه‌بندی موقت که در حافظه قرار دارد، اقدام به ساخت مدل خوشه‌بندی نهائی نمائیم. به همین منظور مطابق الگوریتم ۶ عمل می‌نماییم. بدین ترتیب که در ابتدا، از مدل خوشه‌بندی موقت که مهم‌ترین اطلاعات آن شامل میانگین‌ها خوشه و ماتریس پراکندگی آن می‌باشد، تنها بردارهای میانگین خوشه‌ها را در قالب یک دادگان جدید در نظر می‌گیریم. حال باید بر روی این دادگان متشكل از میانگین‌ها، خوشه‌بندی انجام دهیم که ما در این قسمت نیز مانند مرحله‌ی نمونه‌برداری از همان الگوریتم خوشه‌بندی مبتنی بر چگالی DBSCAN استفاده خواهیم نمود. اما در اینجا هم می‌بایست در ابتدا، مقادیر بهینه به ازای پارامترهای DBSCAN را به دست آورده و سپس اقدام به خوشه‌بندی نمائیم. در

این قسمت از کار نیز مانند آن‌چه در مورد داده‌های نمونه‌برداری شده انجام شد، از الگوریتم تکاملی PSO جهت یافتن این پارامترهای بهینه استفاده کرده و آن‌ها را به ترتیب  $\varepsilon_{means}$  و  $\mu_{means}$  نامیم.

#### **Algorithm 6. Making the final clustering model**

**Input:** Clusters' information array  $C_{inf}$ , Sampling rate  $\eta$ , k-Sigma pruning  $\beta$ .

**Output:** The final clustering model.

Consider mean vectors of whole temporary clusters as a  $K$  by  $p$  matrix  $\mathcal{M}$  and run an arbitrary algorithm to find the best parameters for the DBSCAN algorithm, for clustering  $\mathcal{M}$ . Here we use the PSO algorithm. Call these two parameters  $\varepsilon_{means}$  and  $\mu_{means}$ . Apply DBSCAN algorithm to cluster  $\mathcal{M}$  regarding mentioned parameters. Assume the result of such clustering as  $\{\mathcal{M}_1, \dots, \mathcal{M}_{K'}\}$ .

Consider the final clustering model as it only includes the mean vector  $\mu_{f,i}$  and the covariance matrix  $\Sigma_{f,i}$  as necessary information per each final cluster, for calculating the Mahalanobis distance. Although one can apply PCA on covariance matrix of each final cluster and calculate the Mahalanobis distance in the transformed and reduced dimension space, just for the sake of lowering the computational load and simplifying the whole calculations.

**for each**  $\mathcal{M}_i$

**if**  $|\mathcal{M}_i| == 1$

Assign the mean vector and the covariance matrix of the isolated mean  $\mathcal{M}_i$ , to  $\mu_{f,i}$  and  $\Sigma_{f,i}$  respectively. We gain the covariance matrix of the isolated mean by normalizing its scatter matrix.

**else**

Let  $\mathcal{M}_{i,j}$  be the  $j$ th temporary cluster, included in cluster  $\mathcal{M}_i$ . The value of  $j$  varies among 1 till  $K$ , the number of temporary clusters. We define the final mean vector of such temporary cluster as follows:

$$\mu_{f,i} = \frac{\sum_j [C_{inf}\{6,j\} \times C_{inf}\{1,j\}]}{\sum_j C_{inf}\{6,j\}}$$

Then, for defining the final covariance matrix of related temporary clusters, here, we use mean vectors and covariance matrices of these clusters, to regenerate some new temporary and small clusters, maintaining the intrinsic distribution of real temporary clusters. The size of a regenerated cluster, representing the  $j$ th temporary cluster, is defined through  $\eta \times C_{inf}\{6,j\}$ .

Finally, after regenerating new defined clusters, consider the whole regenerated data points as a single cluster and then, prune it using  $k$ -Sigma pruning  $\beta$ . Therefore, we disregard any regenerated data point that has a Mahalanobis distance more than  $\beta \times \sqrt{p}$  from the regenerated cluster, regarding its mean and covariance matrix. Now, it is time to calculate the covariance matrix of the pruned regenerated cluster and assign it to  $\Sigma_{f,i}$ , the final covariance matrix.

*Clear RAM buffer of any regenerated data point.*

```

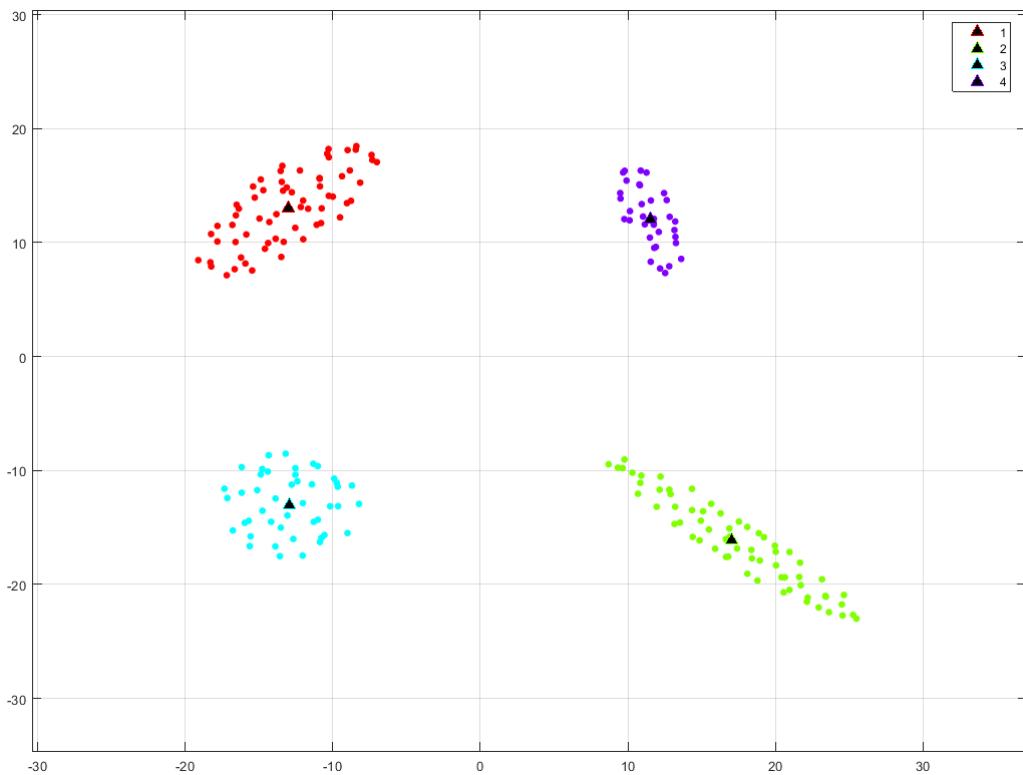
end if
end for

```

پس از اعمال DBSCAN و به دست آوردن خوشه‌های متشكل از بردارهای میانگین، باید اطلاعات خوشه‌های موقت موجود در هر یک از این خوشه‌ها را به نحوی که در ادامه شرح داده خواهد شد، با یکدیگر ترکیب کنیم تا مدل خوشه‌بندی نهائی حاصل گردد. در اینجا فرض می‌آید که اطلاعات مدل خوشه‌بندی نهائی، تنها شامل بردار میانگین و ماتریس کوواریانس به ازای هر خوشه‌ی پایانی می‌باشد. اگرچه که می‌توان مانند آن‌چه که در مورد مدل خوشه‌بندی موقت انجام شد، ماتریس تبدیل و سایر اطلاعات لازم جهت محاسبه‌ی فاصله‌ی ماهالانوبیس در فضای کاوش بُعدیافته را به ازای هر خوشه‌ی نهائی ذخیره نمائیم. اما لوازم همگی این نیازمندی‌ها در ابتدای امر، همان بردار میانگین نهائی و البته ماتریس کوواریانس نهائی می‌باشد. پس برای حصول این دو مهم، باید اطلاعات خوشه‌های موقت موجود در هر خوشه‌ی متشكل از بردارهای میانگین را با یکدیگر تجمعی نمائیم. برای به دست آوردن میانگین نهائی یک خوشه‌ی متشكل از خوشه‌های موقت، می‌توان مطابق الگوریتم ۶ عمل نمود. اگر  $\vec{\mu}_i$  بردار میانگین خوشه‌ی موقت  $i$  و  $n_i$  هم تعداد داده‌های تعلق‌گرفته به این خوشه باشد، بردار مجموع داده‌های این خوشه، برابر با  $\vec{\mu}_i \times n_i$  خواهد بود. حال اگر این مقدار را به ازای تمامی خوشه‌های موقت مربوطه به دست آورده و با یکدیگر جمع نمائیم، بردار مجموع داده‌های متعلق به تمامی خوشه‌های موقت، حاصل شده است. با تقسیم این بردار بر مجموع تعداد داده‌های تعلق‌گرفته به هر یک از خوشه‌های موقت مربوطه، بردار میانگین نهائی به صورت زیر حاصل خواهد شد:

$$\vec{\mu}_f = \frac{\sum_i n_i \times \vec{\mu}_i}{\sum_i n_i} \quad (6.4)$$

شکل ۱۰.۴ نمایی از میانگین‌های موقت و میانگین‌های نهائی را به درستی نشان می‌دهد.

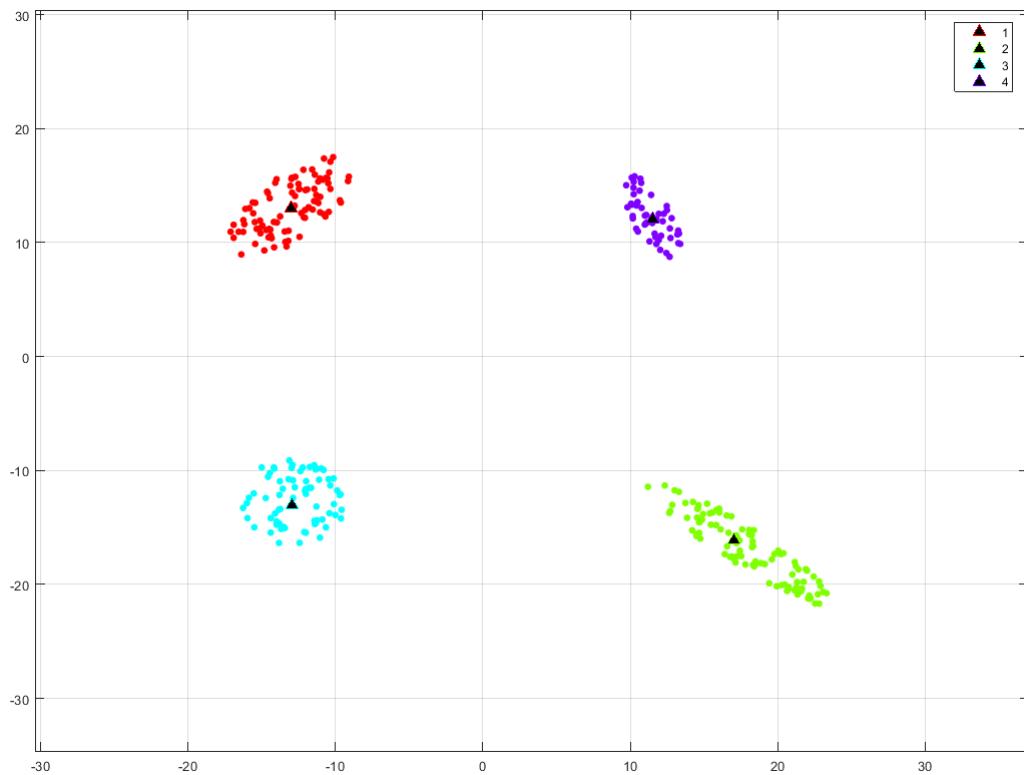


شکل ۱۰.۴ بردارهای میانگین شکل ۹.۴ که با نقاط دایره‌ای شکل نشان داده شده‌اند. این بردارهای میانگین با استفاده از الگوریتم DBSCAN خوشه‌بندی شده و در نهایت با توجه به میانگین‌های موقت تعلق‌گرفته به هر خوشه‌ی نهائی، میانگین نهائی حاصل شده است. میانگین‌های نهائی نیز با نقاط مثلثی شکل نشان داده شده‌اند.

برای به دست آوردن ماتریس کوواریانس یک خوشه‌ی نهائی، نمی‌توانیم مانند آن‌چه در مورد بردار میانگین نهائی انجام شد عمل کرده و ماتریس‌های کوواریانس خوشه‌های موقت را با عملیات جبری با یکدیگر ترکیب نمائیم. لذا به این صورت عمل می‌کنیم که به ازای هر یک از خوشه‌های موقت متعلق به یک خوشه‌ی نهائی و با توجه به میانگین و ماتریس کوواریانس آن خوشه‌ی موقت، اقدام به بازسازی نمونه‌داده‌های آن خوشه می‌نمائیم. اما تعداد این نمونه‌داده‌ها طبعاً برابر با تعداد داده‌های تعلق‌گرفته به هر خوشه‌ی موقت باشد، چرا که در آن صورت ممکن است در طول رویه‌ی ساختن ماتریس کوواریانس خوشه‌ی نهائی، به مشکل محدودیت حافظه برخورده و در نتیجه، ادامه‌ی کار متوقف گردد. لذا تعداد نمونه‌داده‌هایی که به ازای هر خوشه‌ی موقت تولید خواهیم کرد، برابر ضرب نرخ نمونه‌برداری اولیه در تعداد داده‌هایی که تاکنون به این خوشه تعلق گرفته‌اند خواهد بود. حال پس از آن که به ازای هر یک از خوشه‌های موقت متعلق به یک خوشه‌ی نهائی، نمونه‌داده تولید کردیم، همگی این نمونه‌داده‌ها را در کنار یکدیگر در قالب یک خوشه‌ی نهائی موقت در نظر می‌گیریم. اما از آن‌جا که

ممکن است برخی از نمونه‌داده‌ها به دلایلی شدیداً در ناحیه‌ای پرت و دور از این خوشی ترکیبی قرار داشته باشند، باید این خوشی را با توجه به یک معیار، هرس نمائیم. این معیار می‌تواند همان فاصله‌ی ماهالانوبیس داده‌های یک خوشی از میانگین خوشی و با توجه به ماتریس کوواریانس آن خوشی باشد. ما نیز در اینجا آن دسته از داده‌هایی که فاصله‌ی ماهالانوبیس آن‌ها از این خوشی نهائی وقت، بیشتر از یک حد آستانه‌ی مشخص باشد را نادیده می‌گیریم. حال زمان آن است تا این خوشی نهائی وقت هرس‌شده استفاده نموده و ماتریس کوواریانس نهائی را با توجه به بردارهای داده‌های آن به دست آوریم. پس از حصول ماتریس کوواریانس نهائی، باید تمامی نمونه‌داده‌های هرس‌شده متعلق به خوشی نهائی مربوطه را از حافظه‌ی اصلی پاک‌سازی کنیم. این رویه را به ازای هر یک از خوشی‌های نهائی انجام می‌دهیم تا ماتریس کوواریانس‌های نهائی حاصل گرددن. لازم به ذکر است که عمل هرس‌کردن خوشی نهائی وقت، سبب می‌گردد تا ماتریس کوواریانس نهائی به اصطلاح، خوش‌ساخت بوده و در مرحله‌ی نهائی که مرحله‌ی امتیازدهی می‌باشد، تأثیر منفی نداشته باشد. شکل ۱۱.۴ وضعیت داده‌های بازسازی‌شده مربوط به هر یک از چهار خوشی نهائی مدل خوش‌بندی وقت شکل ۹.۴ را نشان می‌دهد که هر خوشی با یک رنگ مجزا نشان داده شده است.

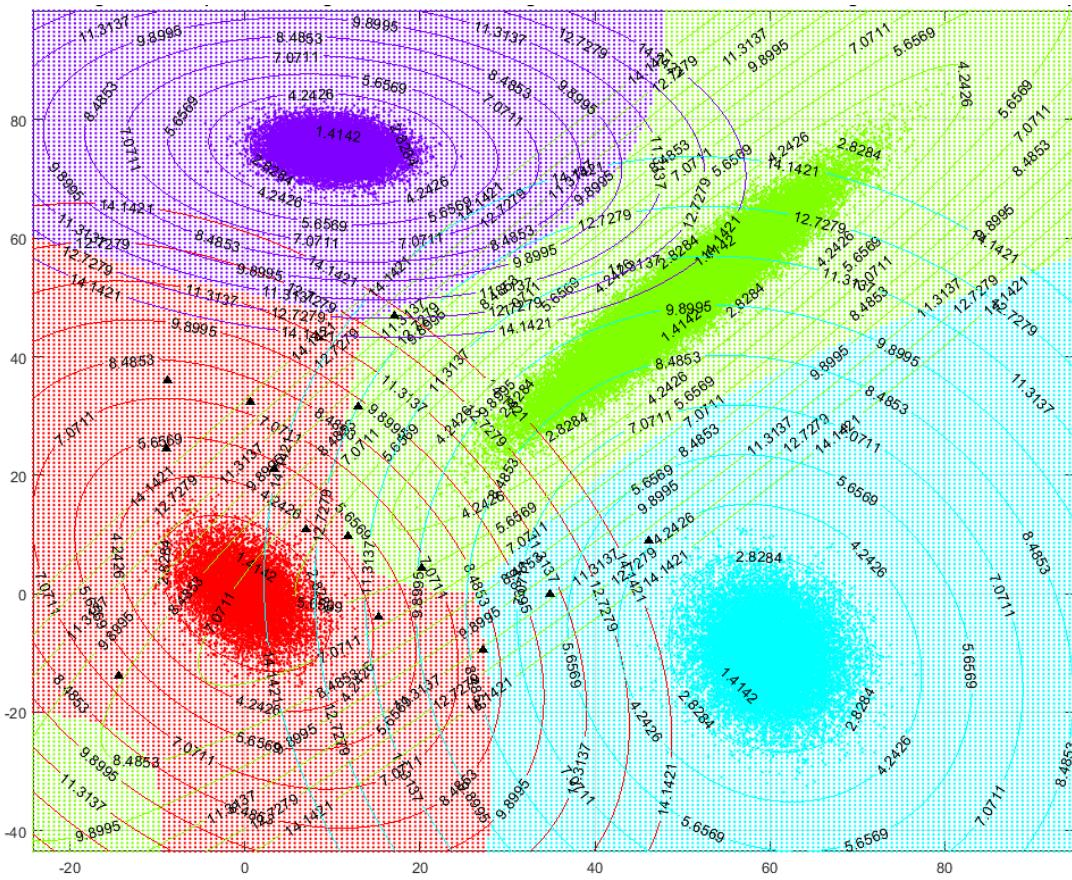
پس تا این جای کار، مدل خوش‌بندی نهائی نیز به دست آمد. در ادامه‌ی الگوریتم پیشنهادی و در مرحله‌ی سوم که مرحله‌ی امتیازدهی می‌باشد، توضیح خواهیم داد که چگونه با استفاده از این مدل خوش‌بندی نهائی، به ازای هر داده، امتیازی مبنی بر میزان پرت‌بودن تعریف خواهیم نمود.



شکل ۱۱.۴ داده‌های بازسازی شده به ازای هر خوش‌های نهائی با توجه به مدل خوش‌بندی موقت شکل ۹.۴. میانگین‌های نهائی نیز به ازای هر خوش‌های با یک مثلث مشکی‌رنگ مشخص شده است. ماتریس کوواریانس یک خوش‌های نهائی از روی داده‌های بازسازی شده‌ی مربوط به آن به دست می‌آید.

### ۳.۲.۴ امتیازدهی

در این مرحله، با استفاده از مدل خوش‌بندی نهائی که از مراحل قبل حاصل شده است، باید به هر کدام از داده‌ها امتیازی مبنی بر میزان پرت‌بودن نسبت دهیم. به عبارت دیگر نیاز خواهد بود تا یک بار دیگر تمامی داده‌های مجموعه‌داده را پردازش نمائیم. بدین ترتیب که باید فاصله‌ی ماهالانوبیس هر داده را تا هر یک از خوش‌های نهائی محاسبه نمائیم و سپس کمترین فاصله را به عنوان امتیاز پرت‌بودن به ازای آن داده در نظر بگیریم. هر داده به آن خوش‌های تعلق می‌گیرد که فاصله‌ی ماهالانوبیس کمتری نسبت به آن خوش‌های دارا باشد.



خوشه‌ها، تحت هیچ شرایطی در محدوده‌ی استحفاظی خوشه‌ی دیگری قرار نگرفته و به عبارتی رنگ دیگر خوشه‌ها را به خود نگرفته‌اند. حتی داده‌های نویزی حاشیه‌ی هر خوشه نیز طبق معیار فاصله‌ی ماهالانوبیس، به همان خوشه تعلق گرفته‌اند. حال اگر داده‌ی پرتی در فضای قرار گرفته باشد (مانند یک سری نقاط که در شکل، در قالب مثلث مشکی نشان داده شده‌اند)، بالاخره به یک کدام از چهار خوشه در فضای تعلق گرفته و در هر صورت، امتیازی بیشتر از امتیاز ماهالانوبیس داده‌های نرمال دریافت خواهد نمود. با این تفاسیر، می‌توان گفت که استفاده از «فاصله‌ی ماهالانوبیس محلی»<sup>۱</sup> به عنوان یک معیار پرتبودن، امری منطقی و مؤثر می‌باشد.

#### ۴.۲.۴ پیچیدگی الگوریتم پیشنهادی

در این قسمت، پیچیدگی زمانی الگوریتم پیشنهادی را به صورت مرحله به مرحله محاسبه می‌نماییم. در ابتدا و در مرحله‌ی نمونه‌برداری، عملیات‌های اصلی، مشتمل بر دو قسمت است. قسمت اول، اعمال الگوریتم خوشه‌بندی DBSCAN بر روی داده‌های نمونه‌برداری شده است که در بدترین حالت، از مرتبه‌ی زمانی  $O(n_B^2)$  می‌باشد، به طوری که  $n_B$ ، معرف تعداد داده‌های یک قطعه‌ی بارگذاری شده در RAM است. قسمت دوم، ساخت مدل اولیه‌ی خوشه‌بندی از روی خوشه‌های به دست آمده می‌باشد که مهم‌ترین بخش آن، اعمال الگوریتم PCA بر روی داده‌های هر یک از خوشه‌های موقت خواهد بود. در اینجا، الگوریتم PCA از مرتبه‌ی زمانی  $O(\min(p^3, n_B^3))$  است، به طوری که  $p$ ، معرف تعداد ابعاد دادگان مربوطه می‌باشد. اما در روش پیشنهادی ما، فرض اولیه بر آن است که دادگان فعلی از نوع کلان‌داده بوده و هر چه هم که ابعاد آن بالا باشد، باز هم  $n_B \ll p$  خواهد بود. در نتیجه، مرتبه‌ی زمانی الگوریتم PCA در اینجا برابر با  $O(p^3)$  است. سایر بخش‌های مربوط به ساخت مدل اولیه‌ی خوشه‌بندی، همگی از مرتبه‌ی زمانی  $O(K \cdot n_B)$  می‌باشند، اگر  $K$  برابر تعداد خوشه‌های موقت موجود باشد. اما از آن‌جا که  $n_B \ll K$  می‌باشد، در نتیجه، مرتبه‌ی زمانی همگی این بخش‌ها نیز برابر با  $O(n_B)$  خواهد بود. لذا مرتبه‌ی زمانی مرحله‌ی اول الگوریتم پیشنهادی برابر با  $O(\max(n_B^2, p^3))$  می‌باشد.

در مرحله‌ی دوم الگوریتم پیشنهادی، در ابتدا می‌بایست به ازای هر قطعه از داده‌ها که مورد بررسی قرار می‌گیرد، مرتبه‌ی زمانی را محاسبه کنیم و سپس این مقدار را در تعداد قطعات مورد بررسی که در

<sup>1</sup> Local Mahalanobis distance

این جا آن را با  $M$  نشان می‌دهیم، ضرب نمائیم. بعد از آن، مرتبه‌ی زمانی را برای فاز نهائی مرحله‌ی دوم که مشتمل بر ترکیب خوش‌های موقع و ساخت مدل خوش‌بندی نهائی می‌باشد، جداگانه حساب خواهیم کرد. در قسمت اول، مرتبه‌ی زمانی بررسی امکان تعلق داده‌های یک قطعه به خوش‌های موجود و سپس به روز کردن اطلاعات خوش‌ها برابر  $O(K \cdot n_B) + O(p^3) \approx O(n_B) + O(p^3)$  خواهد بود. در ادامه، مرتبه‌ی زمانی اعمال الگوریتم‌های DBSCAN و Kmeans بر روی داده‌های معلق موجود در حافظه‌ی اصلی، برابر  $O(n_B^2)$  می‌باشد. افزودن اطلاعات این خوش‌های جدید به مدل خوش‌بندی موقع نیز برابر با همان  $O(n_B) + O(p^3)$  خواهد بود. لذا مرتبه‌ی زمانی قسمت اول از مرحله‌ی دوم روش پیشنهادی، برابر با  $(M \cdot \max(n_B^2, p^3))$  می‌باشد. اما از آنجایی که  $M \ll n_B < \max(n_B^2, p^3)$  می‌باشد، لذا از قید کردن مقدار  $M$  در اینجا صرف نظر می‌نمائیم و در نتیجه مرتبه‌ی زمانی این قسمت برابر با  $\max(n_B^2, p^3)$  خواهد بود.

در مورد فاز نهائی از مرحله‌ی دوم روش پیشنهادی، در ابتدا، اعمال الگوریتم DBSCAN بر روی میانگین‌های خوش‌های موقع، از مرتبه‌ی زمانی  $O(K^2)$  است. محاسبه‌ی میانگین‌های نهائی و نیز تولید نمونه‌داده با توجه به هر یک از خوش‌های موقع، از مرتبه‌ی زمانی  $O(n_B) \approx O(K \cdot n_B)$  خواهد بود. محاسبه‌ی ماتریس‌های کوواریانس نهائی نیز از مرتبه‌ی زمانی  $O(K \cdot n_B \cdot p^2)$  می‌باشد. از آنجا که  $p \ll n_B \ll K$  می‌باشد، مرتبه‌ی زمانی فاز نهائی از مرحله‌ی دوم برابر  $O(n_B)$  خواهد بود. در پایان، مرتبه‌ی زمانی مرحله‌ی دوم از روش پیشنهادی، به صورت  $\max(n_B^2, p^3)$  محاسبه می‌شود.

مرحله‌ی نهائی از روش پیشنهادی ما، با توجه به این‌که تنها شامل بررسی تعلق داده‌های کل مجموعه‌داده یعنی  $n$  به خوش‌های نهائی یا همان  $K$  می‌باشد، از مرتبه‌ی زمانی  $O(n \cdot K)$  خواهد بود. از آنجا که  $n \ll K$  می‌باشد، لذا مرتبه‌ی زمانی این مرحله، برابر با  $O(n)$  می‌باشد.

در نهایت، مرتبه‌ی زمانی الگوریتم پیشنهادی در قالب  $O(\max(n_B^2, p^3)) + O(n)$  محاسبه می‌شود. اما پر واضح است که مقادیر  $n_B$  و  $p$ ، همگی نسبت به  $n$  بسیار ناچیز بوده و قابل چشم‌پوشی می‌باشند. بدین ترتیب، می‌توان گفت که پیچیدگی زمانی الگوریتم پیشنهادی ما خطی و از مرتبه‌ی  $O(n)$  می‌باشد.

در این‌جا، ذکر این نکته ضروری خواهد بود که اگر مدل پیشنهادی ما قرار بود که مانند روش‌های پایه‌ی مبتنی بر چگالی جهت کشف داده‌های پرت، همه‌ی مجموعه‌داده را به یکباره در حافظه بارگذاری نموده و فاصله‌ی دوبه‌دوی داده‌ها را نیز اجباراً در ابتدای امر در اختیار داشته باشد، در آن صورت مرتبه‌ی زمانی

آن برابر  $O(n^2)$  می‌بود که در مورد کلان داده‌ها و با توجه به محدودیت منابع فیزیکی از جمله حافظه‌ی اصلی یا RAM، امری غیرممکن می‌باشد.

۵

## فصل پنجم

# نتایج آزمایشات انجام شده

در این فصل، به بررسی دو نوع آزمایش «اثربخشی»<sup>۱</sup> و «بازدھی»<sup>۲</sup> برای تحلیل عملکرد روش پیشنهادی خواهیم پرداخت. در مورد اثربخشی، عملکرد الگوریتم پیشنهادی و برخی روش‌های رقیب را بر روی مجموعه داده‌های واقعی و مصنوعی، آزمایش خواهیم نمود تا ببینیم که آیا این روش‌ها به درستی، قادر به شناسائی داده‌های پرت از نرمال هستند یا خیر. در مورد بازدھی نیز آزمایشاتی را بر روی مجموعه داده‌های مصنوعی انجام خواهیم داد تا ببینیم که چگونه میزان دقت الگوریتم پیشنهادی در تمیزداندن داده‌های پرت از نرمال، با افزایش تعداد داده‌ها، تعداد ابعاد و تعداد داده‌های پرت تغییر خواهد کرد.

## ۱.۵ روش‌های رقیب و طرح کلی آزمایشات

برای انجام آزمایشات مربوطه، ما روش پیشنهادی خویش را با چند روش مطرح دیگر در زمینه‌ی کشف داده‌های پرت در مجموعه داده‌های اقلیدسی مورد مقایسه قرار می‌دهیم. این روش‌های رقیب، شامل LOF، iLOF و OCSVM و iForest، LoOP یا باشد. روش LOF هم از آنجائی که نتایج آن با نتایج نسخه‌ی پایه‌ی آن یعنی LOF کاملاً تطابق دارد، از قید نتایج آن به صورت جداگانه اجتناب می‌کنیم.

دو روش LOF و LoOP، روش‌های به اصطلاح «پایه»<sup>۳</sup> جهت کشف داده‌های پرت محلی بوده و مبتنی بر چگالی می‌باشند که در ابتدای امر و در آن واحد، نیازمند دانستن فاصله‌ی دوبه‌دروی داده‌های موجود می‌باشند و به عبارتی بر خلاف روش پیشنهادی، نیازمند آن هستند تا کل مجموعه داده را به یکباره در حافظه‌ی اصلی بارگذاری نمایند. این دو روش، در مورد مجموعه داده‌های با اندازه‌ی نرمال، از دقت بسیار بالائی برخوردار می‌باشند، اما مرتبه‌ی زمانی و البته حافظه‌ی مصرفی آن‌ها در مورد کلان داده‌ها شدیداً طاقت‌فرسا می‌باشد و در نتیجه مانیز در آزمایشات، از این روش‌ها تنها در مورد آن دسته از مجموعه داده‌هایی که منابع موجود، اعم از حافظه و پردازش‌گر کفاف خواهند داد، استفاده خواهیم نمود.

آزمایشات بسیاری در این فصل ارائه شده‌اند. جهت بررسی اثربخشی روش پیشنهادی، آن را به همراه روش‌های رقیب مطرح شده، بر روی مجموعه داده‌های واقعی متعددی آزمایش نموده‌ایم. در ادامه‌ی بررسی

<sup>1</sup> Effectiveness

<sup>2</sup> Efficiency

<sup>3</sup> Base methods

اثربخشی، آزمایشاتی نیز بر روی تعداد محدودی مجموعه‌داده‌ی مصنوعی انجام داده‌ایم تا هم کارائی و هم «پایداری»<sup>۱</sup> روش پیشنهادی را در مقایسه با سایر روش‌های رقیب و در مورد مجموعه‌داده‌های با مقیاس بزرگ نشان دهیم. در ادامه، برای آزمایش بازدهی روش پیشنهادی، از مجموعه‌داده‌های مصنوعی متنوع، با تعداد داده‌ها، تعداد ابعاد و تعداد داده‌های پرت متفاوت استفاده خواهیم نمود تا میزان صحت عملکرد الگوریتم را در تشخیص داده‌های پرت از نرمال ارزیابی نمائیم. در ادامه‌ی آزمایشات، همچنان نمودار پیچیدگی زمانی الگوریتم پیشنهادی را در مقایسه با تعداد داده‌های رو به افزایش رسم خواهیم نمود. همین‌طور نمودار تغییرات دترمینان ماتریس کوواریانس داده‌های نمونه‌برداری شده از یک خوش را نسبت به نرخ نمونه‌برداری رسم می‌کنیم تا نشان دهیم که به ازای نرخ‌های نمونه‌برداری خیلی کم نیز می‌توان انتظار دقت بالائی از روش پیشنهادی را داشت.

## ۲.۵ آزمایش اثربخشی

در این قسمت، نتایج آزمایشات انجام شده بر روی مجموعه‌داده‌های واقعی و مصنوعی متنوع را ارائه می‌کنیم تا اثربخشی و کارائی الگوریتم پیشنهادی را در مورد مجموعه‌داده‌های با مقیاس و ابعاد متنوع مورد بررسی قرار دهیم.

## ۱.۲.۵ آزمایش بر روی مجموعه‌داده‌های واقعی

تعداد زیادی از مجموعه‌داده‌های واقعی از زمینه‌های گوناگون که اکثریت قریب به اتفاق آن‌ها از [۳۱] جمع‌آوری شده‌اند، در آزمایشات ما مورد استفاده واقع شده‌اند. همگی این مجموعه‌داده‌های واقعی پیش از استفاده، توسط [۳۲] پیش‌پردازش شده‌اند. به این معنی که برخی ویژگی‌های بی‌مورد از آن‌ها حذف شده و یا هم مقادیر مفقودی در یک دادگان، با مقدار «نما»<sup>۲</sup> در ویژگی مربوطه جایگزین شده است تا رویه‌ی کشف داده‌های پرت، با سهولت بیشتر و خطای کمتری انجام پذیرد. همین‌طور داده‌های هر کدام از این مجموعه‌داده‌های واقعی، به لحاظ نرمال و یا پرت‌بودن برچسب خورده‌اند. برچسب «۰» به معنای

<sup>1</sup> Stability

<sup>2</sup> Mode

نرمال بودن و برچسب «۱» نیز به معنای پرت بودن می‌باشد. داده‌های نرمال، عمدتاً متعلق به کلاس‌های پر جمعیت بوده و داده‌های پرت نیز غالباً از میان داده‌های متعلق به کلاس‌های کوچک انتخاب شده‌اند.

در تمامی این آزمایشات، از معیار «مساحت زیر منحنی (نرخ شناسائی و نرخ هشدار اشتباه)»<sup>۱</sup> یا به اختصار AUC [۱, ۵]، جهت ارزیابی عملکرد الگوریتم‌ها استفاده شده است. معیار ارزیابی AUC، به طور معمول با مقداری در بازه‌ی [۰, ۱] گزارش می‌شود، اما ما در اینجا، برای این‌که بتوانیم نتایج حاصله را با دقیق‌تری با یکدیگر مقایسه کنیم، مقدار این معیار را در ۱۰۰ ضرب نموده و در قالب درصد بیان خواهیم کرد. مقدار نتایج AUC به ازای تمامی مجموعه‌داده‌های مورد آزمایش و نیز برخی از ویژگی‌های این مجموعه‌داده‌ها اعم از تعداد داده‌ها (#n)، تعداد ابعاد (#p) و همین‌طور تعداد داده‌های پرت (#o)، همگی در جدول ۱.۵ قید شده‌اند. درایه‌هایی از جدول که با پیش‌زننده‌ی سبزرنگ نشان داده شده‌اند، نشان‌گر بهترین روش(ها) در مورد مجموعه‌داده‌ی مربوطه می‌باشند.

جدول ۱.۵ نتایج AUC حاصل از آزمایش روش پیشنهادی و روش‌های رقیب، بر روی مجموعه‌داده‌های واقعی و مصنوعی

	Dataset	#n	#p	#o	LOF (IncLOF)	LoOP	iForest	OCSVM	PropMeth
Real Data- sets	Annthyroid	7,200	6	534	75.58	76.34	80.3	54.67	93.44
	Cardio	1,831	21	176	94.94	94.68	92.6	93.52	95.09
	Letter	1,600	32	100	90.14	89.7	63.23	88.96	90.33
	Mammography	11,183	6	260	87.47	88.69	86.65	87.21	88.44
	Musk	3,062	166	97	100	100	100	57.28	100
	Pima	768	8	268	62.94	66.95	68.43	53.93	69.79
	Shuttle	49,097	9	3511	99.47	98.56	99.69	72.49	98.86
	Vowels	1,456	12	50	93.62	92.81	75.06	77.83	93.83
	Yeast	1,484	8	5	98.62	98.58	99.53	98.66	99.89
	real data results average				89.20	89.59	85.05	76.06	92.19
Synth. Data- sets	Data1	21,867	2	600	100	100	~	~	100
	Data2	109,478	20	1000	~	~	~	~	100
	Data3	1,122,601	40	10000	~	~	~	~	100

اکنون می‌خواهیم میان روش پیشنهادی با سایر روش‌های رقیب، مقایسه‌ای انجام دهیم. نتایج جدول ۱.۵ نشان می‌دهند که روش پیشنهادی ما نسبت به LOF، iForest، LoOP و OCSVM اثربخشی بیشتری دارد و البته توانسته است تا در قریب به ۸۰ درصد از مجموعه‌داده‌های موجود، موفق‌تر عمل کند. حتی در

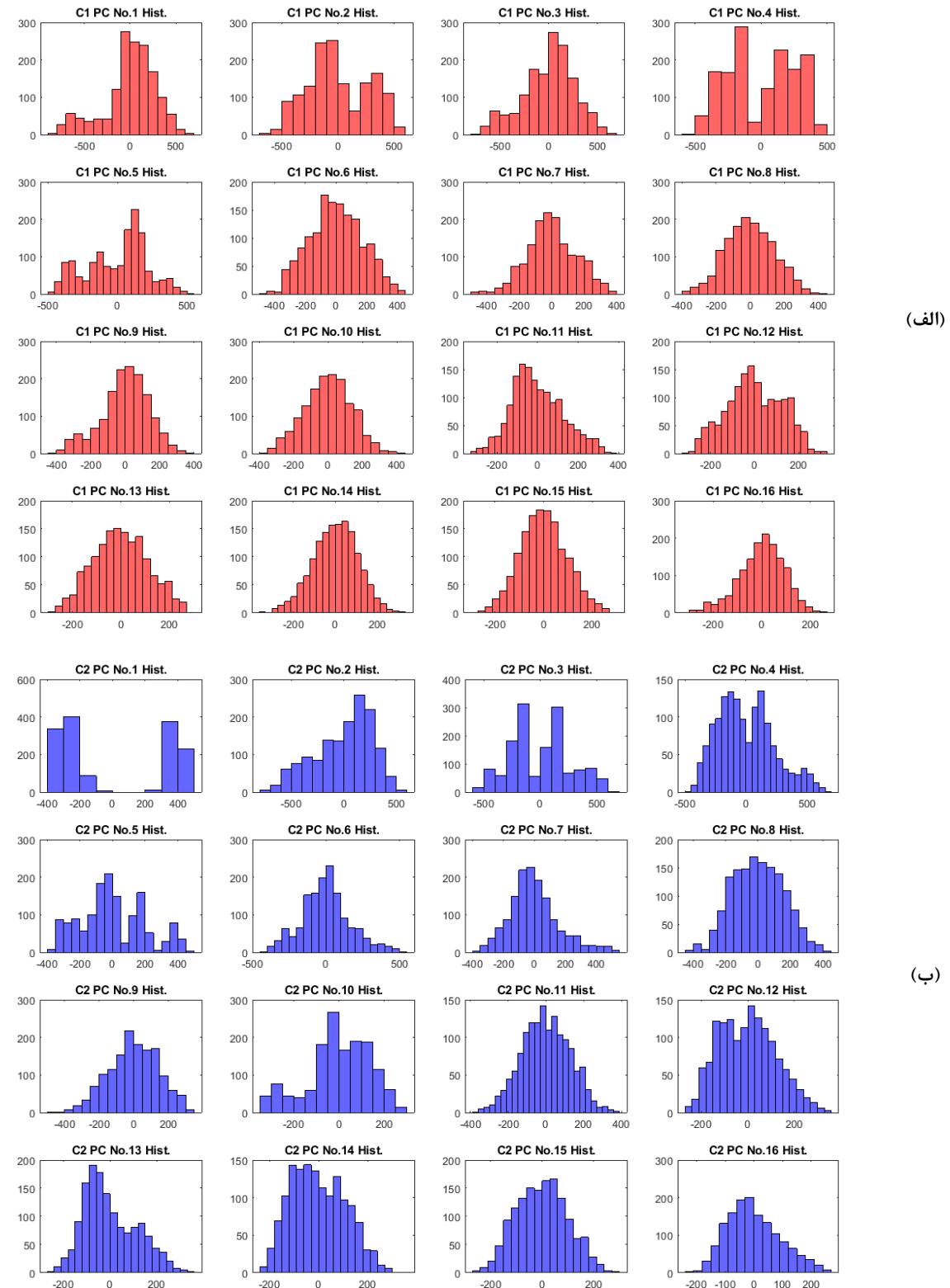
<sup>1</sup> Area Under Curve (of detection rate and false alarm rate)

مواردی که روش پیشنهادی، نتیجه‌ی بهتری را نسبت به بهترین روش گزارش شده به دست نیاورده است، مشاهده می‌کنیم که نتایج مربوط به روش ما، نسبت به بهترین نتایج مربوطه بسیار قرابت داشته و اختلاف آن‌ها به طرز قابل ملاحظه‌ای قابل چشم‌پوشی است. همین‌طور اگر به میانگین نتایج حاصله از کلیه‌ی روش‌ها در مورد مجموعه‌داده‌های واقعی توجه کنیم، پر واضح است که روش پیشنهادی ما، نسبت به سایر روش‌های رقیب، بسیار بهتر عمل نموده است.

آن‌چه در مورد مجموعه‌داده‌های واقعی مورد استفاده باید ذکر گردد، آن است که اکثریت قریب به اتفاق آن‌ها نه تنها کلان‌داده به حساب نمی‌آیند، بلکه در زمرة‌ی مجموعه‌داده‌های با مقیاس بزرگ نیز قرار نمی‌گیرند. لذا در مورد برخی از این مجموعه‌داده‌ها، زمانی که به دنبال یافتن پارامترهای بهینه بودیم، ناگریز از نرخ‌های نمونه‌برداری نسبتاً بالا استفاده کردیم تا به قضیه‌ی «نفرین ابعاد»<sup>۱</sup> ۴ چار نشویم. چرا که در مورد روش پیشنهادی ما، این مسئله باید برقرار باشد که نسبت تعداد داده‌های هر کدام از خوش‌های یافت‌شده به تعداد ابعاد مجموعه‌داده، باید یک نرخ مناسب باشد. در غیر این صورت، ماتریس کوواریانسی که از این خوش، آن هم خصوصاً در زمان نمونه‌برداری و با نرخ نمونه‌برداری خیلی کم به دست می‌آید، به احتمال زیاد، مثبت و نیمه‌قطعی نبوده و در نتیجه معتبر نمی‌باشد. به دنبال این مسئله، روش پیشنهادی ما در مورد چنین مجموعه‌داده‌ای با احتمال بالا با شکست مواجه خواهد شد.

در مورد روش‌های پایه‌ی LOF و LOOP هم ذکر این نکته ضروری است که پارامترهای لازم به ازای این روش‌ها را به صورت تجربی و با آزمایش‌های مکرر به دست آورده‌یم. بدین‌صورت که با تغییر پارامترهای مربوط به هر یک از این روش‌ها، به دنبال کسب بهترین نتیجه‌ی AUC هستیم. این رویه را تا آن‌جا ادامه می‌دهیم که بهترین محدوده برای پارامترهای بهینه را پیدا کنیم.

<sup>1</sup> Curse of dimensionality



شکل ۱.۵ نمودارهای هیستوگرام به ازای تعداد ۱۶ مؤلفه اصلی اول (بعد از اعمال PCA) دو خوشی موجود در مجموعه داده‌ی واقعی (الف) نمودارهای هیستوگرام قرمزرنگ به ازای خوشی اول؛ (ب) نمودارهای هیستوگرام آبی‌رنگ به ازای خوشی دوم.

همان طور که در مورد روش پیشنهادی، پیش از این به ضرورت ذکر گردید که فرض اولیه‌ی قوی آن در مورد نرمال بودن توزیع خوش‌های موجود در مجموعه داده می‌باشد، می‌توان نشان داد که در مورد تمامی مجموعه داده‌های واقعی که روش پیشنهادی ما موفق به حصول نتیجه‌ی عالی گشته است، این فرض اولیه‌ی قوی برقرار می‌باشد. در اینجا، ما تنها در مورد یکی از مجموعه داده‌های واقعی که موفق به کسب نتیجه‌ی AUC برابر با ۱۰۰ درصد با استفاده از روش پیشنهادی گشته‌ایم، نرمال بودن توزیع خوش‌های آن را در هر مؤلفه‌ی اصلی و با استفاده از «نمودار هیستوگرام»<sup>۱</sup> نشان می‌دهیم. مجموعه داده‌ی واقعی مربوطه، *musk* نام دارد که دارای تعداد ۳۰۶۲ داده و ۱۶۶ ویژگی می‌باشد. این مجموعه داده با توجه به اطلاعات اولیه در مورد آن، دارای دو عدد خوش می‌باشد که نمودارهای هیستوگرام به ازای تعداد ۱۶ مؤلفه‌ی اصلی اول این خوش‌ها (بعد از اعمال PCA بر روی هر خوش) در شکل ۱.۵ نمایش داده شده است.

همان طور که از نمودارهای هیستوگرام موجود در شکل ۱.۵ پیداست، توزیع موجود در اکثریت قریب به اتفاق مؤلفه‌های اصلی مربوطه، از نوع گاویین بوده و یا هم شدیداً به این توزیع شباهت دارد. لازم به ذکر است که مجموعه داده‌ی *musk*، دارای تعداد ۱۶۶ ویژگی می‌باشد که ما در اینجا تنها به ذکر تعداد ۱۶ ویژگی اول، بسنده نموده‌ایم.

## ۲.۲.۵ آزمایش بر روی مجموعه داده‌های مصنوعی

ما هم‌چنین آزمایشات اثربخشی را بر روی مجموعه داده‌های مصنوعی نیز انجام داده‌ایم. این مجموعه داده‌های مصنوعی به گونه‌ای ایجاد شده‌اند که اولاً از فرض اولیه‌ی قوی الگوریتم پیشنهادی، یعنی نرمال بودن توزیع خوش‌ها و نیز امکان همبسته بودن ویژگی‌های آن‌ها پیروی می‌نمایند و نیز داده‌های پر تی که اصطلاحاً به این مجموعه داده‌ها «تزریق»<sup>۲</sup> شده‌اند، نسبت به مجموعه داده‌های واقعی، بیشتر قابل تشخیص می‌باشند. تمامی این مجموعه داده‌ها به لحاظ نرمال و یا پرت بودن، برچسب خورده‌اند. تعداد سه آزمایش بر روی مجموعه داده‌های مصنوعی در پایین جدول ۱.۵ ارائه شده‌اند. تعداد داده‌های پرت موجود در هر یک از این مجموعه داده‌ها، نرخی کمتر از ۳ درصد تعداد کل داده‌ها را دارند.

<sup>1</sup> Histogram plot

<sup>2</sup> Inject

در این قسمت، به چگونگی ایجاد خوشه‌های با توزیع گاووسین اشاره می‌کنیم. برای تولید هر خوشه با توزیع گاووسین با  $p$  بُعد، نیازمند یک بردار میانگین و یک ماتریس کوواریانس می‌باشیم. درایه‌های بردار میانگین را به صورت تصادفی تولید می‌کنیم، اما برای تولید ماتریس کوواریانس به این صورت عمل می‌کنیم که در ابتدای امر، یک ماتریس  $[A]_{p \times p}$  را به صورت کاملاً تصادفی تولید می‌نمائیم. به این معنی که هر درایه‌ی این ماتریس، با استفاده از یک «توزیع یکنواخت»<sup>۱</sup> و از اعداد مابین  $0$  و  $1$  انتخاب می‌گردد. سپس به تعداد حدودی نصف درایه‌های ماتریس  $A$  را به صورت تصادفی انتخاب کرده و آن‌ها را منفی می‌کنیم. در نهایت ماتریس کوواریانس مربوطه در قالب  $\Sigma = A^T A$  به دست می‌آید. حال می‌توان با استفاده از بردار میانگین خوشه (مکان خوشه) و ماتریس کوواریانس آن (شاکله‌ی خوشه)، اقدام به تولید نمونه‌داده از خوشه‌ی مربوطه و به تعداد دلخواه نمائیم. همین‌طور جهت این‌که داده‌های نویزی موجود در حاشیه‌ی خوشه را از بین ببریم، می‌توانیم از معیار فاصله‌ی ماهالانوبیس استفاده کرده و آن دسته از داده‌هایی که فاصله‌ی ماهالانوبیس آن‌ها از میانگین خوشه و با توجه به ماتریس کوواریانس آن، از به عنوان مثال  $\sqrt{p} \times 1$  بیشتر باشد را دور ببریزیم. لازم به ذکر است که جهت تولید بردارهای میانگین به ازای خوشه‌های مختلف، تا آن‌جا که ممکن است این بردارها را از یکدیگر دور تولید می‌کنیم تا خطر احتمالی آن‌که خوشه‌های تولیدشده با یکدیگر همپوشانی داشته باشند، تا حد بسیار بالاتی مرتفع گردد.

برای تزریق داده‌های پرت محلی به یک خوشه نیز می‌توان از همان معیار فاصله‌ی ماهالانوبیس بهره برد. بدین صورت که در ابتداء، به ازای هر خوشه، یک محدوده‌ی استحفاظی دلخواه که تمام خوشه و بخشی از فضای خالی اطراف آن را پوشش دهد در نظر می‌گیریم. سپس در این فضا به صورت تصادفی بردار داده تولید کرده و در عین حال، بررسی می‌کنیم که آیا در محدوده‌ی همسایگی ماهالانوبیس مجاز به ازای این خوشه قرار گرفته است یا خیر. این محدوده‌ی همسایگی ماهالانوبیس مجاز، می‌تواند هر بازه‌ای مانند بازه‌ی  $[4\sqrt{p}, 10\sqrt{p}]$  باشد. به عبارتی، داده‌ی تصادفی تولیدشده می‌بایست فاصله‌ی ماهالانوبیس آن از خوشه‌ی مربوطه، در این بازه‌ی مجاز قرار گرفته باشد تا به عنوان یک داده‌ی پرت محلی پذیرفته شود.

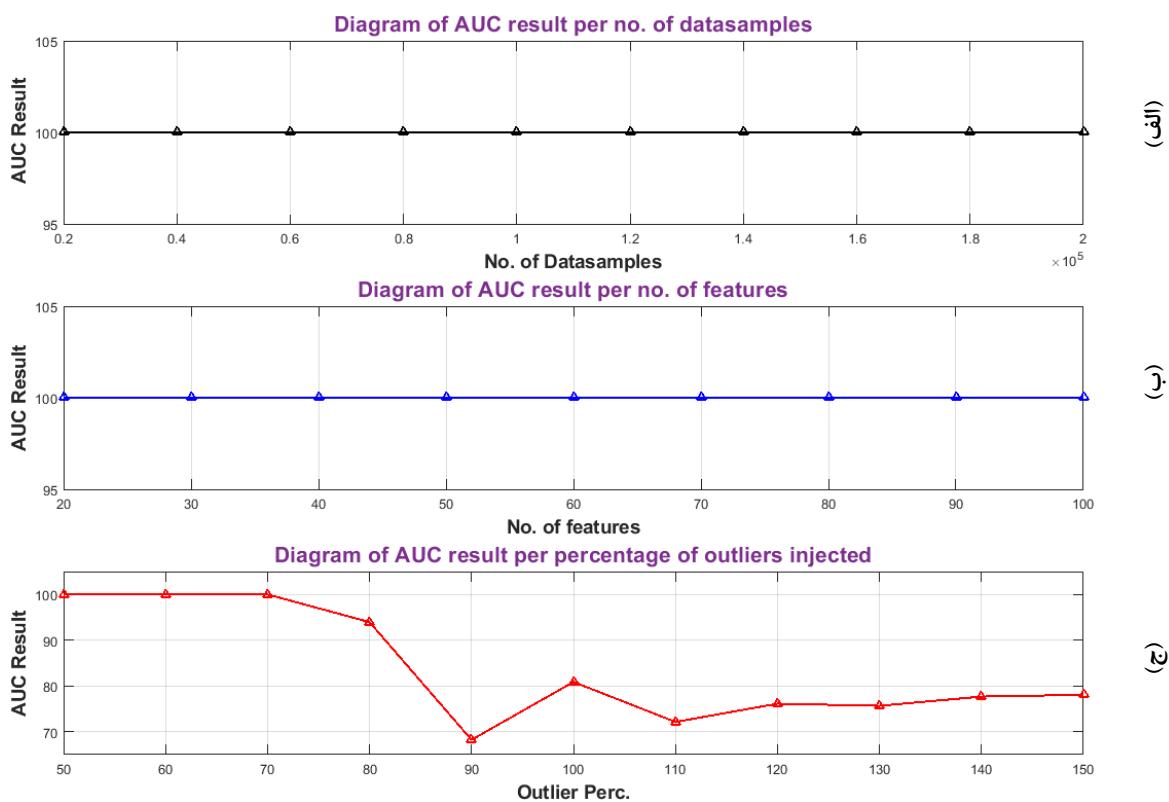
نتایج موجود در جدول ۱.۵، مربوط به مجموعه‌داده‌های مصنوعی تولیدشده به شیوه‌ای که قید شد، نشان می‌دهند که کشف داده‌های پرت در چنین مجموعه‌داده‌هایی، به طور عمومی برای روش پیشنهادی ما

<sup>۱</sup> Uniform distribution

بسیار آسان می‌باشد. چرا که داده‌های نرمال این گونه مجموعه‌داده‌ها در نواحی شدیداً چگال قرار داشته و داده‌های پرت نیز تا حد امکان، در نقاط دورتری از خوش‌ها قرار گرفته‌اند. نتیجه‌ی AUC گزارش شده به ازای روش پیشنهادی و بر روی این سه مجموعه‌داده، همگی عالی و برابر با ۱۰۰ درصد می‌باشد. به ازای دو روش LOOP و LOF هم همان‌طور که پیش از این نیز قید شد، به دلیل پیچیدگی محاسباتی و حافظه‌ی مصرفی بسیار بالائی که نیاز دارند، تنها در مورد یکی از مجموعه‌داده‌های مصنوعی که کوچک‌ترین اندازه را دارد، موفق به حصول نتیجه شدیم. در مورد سایر روش‌های رقیب هم به دلیل پیچیدگی محاسباتی بسیار بالائی که داشتند، موفق به حصول نتیجه نگشتم.

### ۳.۵ آزمایش بازدهی

برای ارزیابی میزان دقت نهائی AUC به ازای افزایش تعداد داده‌ها، تعداد ابعاد و تعداد داده‌های پرت موجود، ما از همان روشی که پیش از این قید شد، جهت تولید مجموعه‌داده‌های مصنوعی متنوع بهره برده‌ایم. در مورد آزمایش بازدهی، زمانی که تعداد داده‌ها رو به افزایش است، تعداد داده‌ها یا  $n$  را از مقدار ۲۰ هزار تا ۲۰۰ هزار و با طول گام ۲۰ هزار تائی افزایش داده‌ایم. تعداد ابعاد و تعداد داده‌های پرت موجود در تمامی این ۱۰ مجموعه‌داده‌ی مصنوعی تولیدشده به ترتیب برابر ۲۰ و ۲۰۰ می‌باشد. در مورد زمانی که تعداد ابعاد یا  $p$  رو به افزایش است، تعداد ابعاد را از ۲۰ تا ۱۰۰ و با طول گام ۱۰ تائی افزایش داده‌ایم. تعداد داده‌ها و تعداد داده‌های پرت موجود نیز در تمامی این مجموعه‌داده‌ها به ترتیب برابر ۱۰۰ هزار و ۱۰۰۰ می‌باشد. در نهایت، در مورد زمانی که تعداد داده‌های پرت تزریق شده یا ۰ رو به افزایش است، نرخ این داده‌ها را از ۵۰ درصد تا ۱۵۰ درصد اندازه‌ی مجموعه‌داده و با طول گام ۱۰ درصد افزایش داده‌ایم. تعداد داده‌ها و تعداد ابعاد هم در تمامی این مجموعه‌داده‌ها به ترتیب برابر ۲۰ هزار و ۲ می‌باشد. نتایج حاصله به تفصیل در ادامه می‌آید.



شکل ۲.۵ نتایج آزمایش بازدهی بر روی مجموعه داده‌های مصنوعی. (الف) زمانی که تعداد داده‌ها را به افزایش باشد؛ (ب) زمانی که تعداد ابعاد را به افزایش باشد؛ (ج) زمانی که تعداد داده‌های پرت تزریق شده را به افزایش باشد.

شکل ۲.۵ (الف)، نتایج آزمایش بازدهی در زمانی را نشان می‌دهد که تعداد داده‌ها را به افزایش باشد. از این نمودار، واضح است که به ازای افزایش تعداد داده‌ها، میزان نتیجه‌ی AUC گزارش شده، همواره برابر با بهترین مقدار ممکن یعنی ۱۰۰ درصد می‌باشد. علت این امر نیز آن است که از آن‌جا که مدل ما مبتنی بر ماتریس کوواریانس خوش‌ها می‌باشد، و در صورتی که نرخ تعداد داده‌ها به تعداد ابعاد یک خوش، پایین‌تر از حد مجاز برای تشکیل یک ماتریس کوواریانس مثبت و نیمه قطعی باشد، آن‌گاه در محاسبه‌ی فاصله‌ی ماهالانوبیس یک داده‌ی دلخواه از چنین خوش‌هایی به مشکل جدی برخواهیم خورد. در آزمایشات انجام شده، متوجه شدیم که در صورت رخداد چنین مسئله‌ای، مقدار فاصله‌ی ماهالانوبیس به احتمال بالائی مختلط خواهد شد و در نتیجه دیگر قابل استفاده به عنوان یک امتیاز پرت‌بودن معتبر نخواهد بود. این قضیه، یادآور همان قضیه‌ی نفرین ابعاد می‌باشد لذا با توجه به آن‌چه در مورد نرخ تعداد داده‌های یک خوش به تعداد ابعاد آن مطرح شد، هر چه که تعداد داده‌ها در عین ثابت‌بودن تعداد ابعاد، افزایش یابد، در آن صورت است که مقدار این نرخ نیز افزایش می‌یابد و در نتیجه خطرات احتمالی مطرح شده در مورد

ماتریس کوواریانس خوش و نیز فاصله‌ی ماهالانوبیس حاصله نیز تا حد بسیار بالائی مرتفع می‌گردد. به دنبال این قضیه، میزان دقت AUC گزارش شده نیز به ازای استفاده از پارامترهای بهینه، همواره برابر با بهترین مقدار ممکن خواهد بود.

شکل ۲.۵ ب، نتایج آزمایش بازدهی را در مورد افزایش تعداد ابعاد نشان می‌دهد. این نمودار نیز نشان می‌دهد که به ازای افزایش تعداد ابعاد مجموعه‌داده، البته تا جایی که دچار قضیه‌ی نفرین ابعاد نگشته‌ایم، مانند آزمایش پیشین در مورد افزایش تعداد داده‌ها، همواره بالاترین دقت AUC ممکن را به دست خواهیم آورد.

شکل ۲.۵ ج، نتایج آزمایش بازدهی را در مورد افزایش تعداد داده‌های پرت تزریق شده نشان می‌دهد. از این نمودار می‌توان فهمید که تحمل الگوریتم پیشنهادی در مورد زمانی که تعداد داده‌های پرت رو به افزایش است، همواره بالا نبوده و از یک مرحله به بعد، دیگر نمی‌توان انتظار دقت AUC بالائی از آن داشت. به عبارتی زمانی که تعداد داده‌های پرت، به تعداد داده‌های نرمال مجموعه‌داده نزدیک می‌شود، روش پیشنهادی در شناسائی این دو از یکدیگر دچار خطأ خواهد شد. علت این مسئله آن است که روش پیشنهادی ما مبتنی بر یک خوش‌بندی سلسله‌مراتبی می‌باشد که بنای آن بر چگالی و تراکم داده‌ها می‌باشد. حال اگر تعداد داده‌های پرت تزریق شده در نواحی غیر از خوش‌ها، به حدی زیاد گردد که این داده‌های پرت بتوانند شروط الگوریتم DBSCAN را برآورده نمایند، آن‌گاه خواهند توانست در این نواحی، تشکیل خوش داده و به دنبال آن و به مرور آمدن قطعه‌های داده، داده‌های پرت و یا حتی داده‌های نرمال را نیز به خود جذب نمایند. اما همان‌طور که از نمودار شکل ۲.۵ ج پیداست، مقدار دقت AUC گزارش شده، با افزایش درصد داده‌های پرت تزریقی خیلی دیر دچار افت می‌شود. به طوری که ما در این آزمایشات، از میزان ۵۰ درصد تعداد داده‌های نرمال شروع کرده و تا میزان ۱۵۰ درصد پیش‌رفته‌ایم و قطعاً در یک مجموعه‌داده‌ی واقعی، تعداد داده‌های پرت به این میزان، بالا نخواهد بود. چرا که عمدتاً نسبت تعداد داده‌های پرت به تعداد داده‌های نرمال یک مجموعه‌داده، یک نرخ بسیار پایین و در بدترین حالت، بالای ۵ درصد می‌باشد. اما علت این مسئله که چرا میزان دقت AUC در چنین شرایطی، این قدر دیر افت می‌کند آن است که بنای تشکیل خوش‌ها در روش پیشنهادی، در ابتدای امر، برآورده شدن شروط الگوریتم DBSCAN می‌باشد و این شروط، همان‌طور که در فصل پیشین به تفصیل قید گردید، در زمان نمونه‌برداری از مجموعه‌داده حاصل می‌شوند. در زمان نمونه‌برداری از یک مجموعه‌داده نیز احتمال حضور

داده‌های پرت نسبت به داده‌های نرمال به مراتب کمتر است، چرا که داده‌های نرمال، متعلق به خوش‌های با توزیع گاویسین بوده و داده‌های پرت نیز با استفاده از یک توزیع یکنواخت در سراسر فضای خالی مجموعه‌داده تزریق شده‌اند. در نتیجه، داده‌های نرمال، در نواحی‌ای بسیار چگال‌تر نسبت به داده‌های پرت حضور دارند. همین مسئله سبب می‌گردد تا پارامترهای  $\epsilon$  و  $\text{m}$  متعلق به الگوریتم خوش‌بندی مبتنی بر چگالی DBSCAN، متناسب با نواحی چگال به دست آمده و به دنبال آن، احتمال تشکیل خوش به مرور آمدن قطعه‌های داده، در نواحی‌ای که داده‌های پرت با توزیع یکنواخت تزریق شده‌اند، به مراتب کمتر از نواحی مربوط به داده‌های نرمال با توزیع گاویسین باشد. اما این مسئله همیشه برقرار نبوده و همان‌طور که از نمودار مربوطه نیز پیداست، از نرخ تزریقی بالای ۷۰ درصد، تحمل روش پیشنهادی در برابر حضور داده‌های پرت، کاهش می‌یابد و داده‌های پرت، دیگر قادر به تشکیل خوش و به دنبال آن، کاهش دقت الگوریتم در شناسائی خواهد شد.

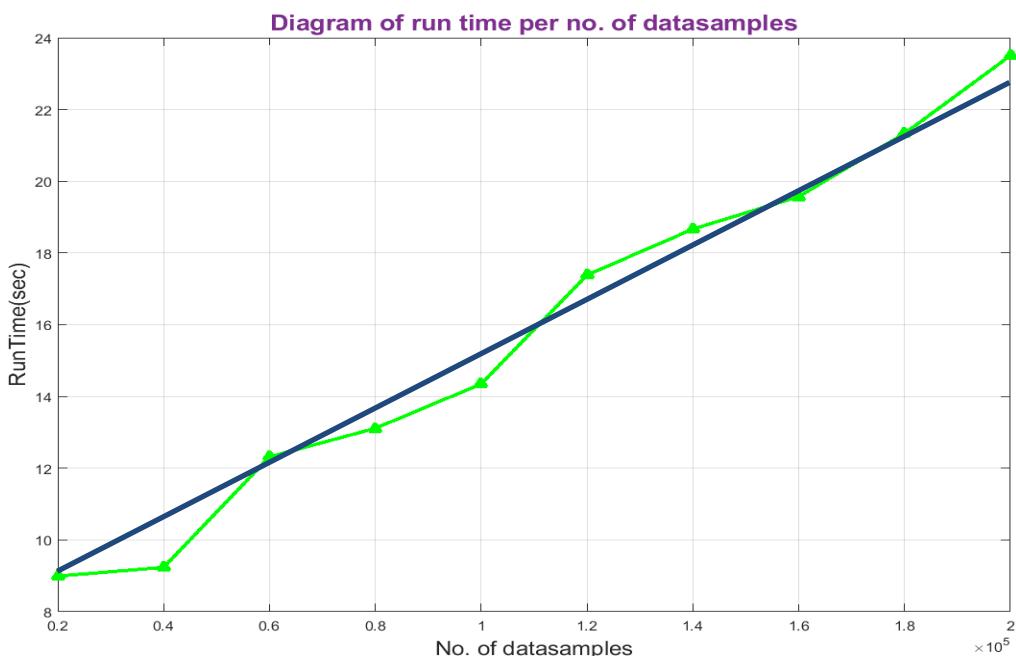
در نهایت به طور خلاصه، از این سه نمودار بازدهی می‌توان این‌گونه برداشت کرد که روش پیشنهادی ما در برابر افزایش تعداد داده‌ها و تعداد ابعاد، عملکرد بهتری را از خود نشان خواهد داد و در برابر افزایش تعداد داده‌های پرت نیز تحمل بالائی را از خود نشان می‌دهد. به طوری که حتی به ازای نرخ تزریقی بسیار بالای ۱۵۰ درصد، مقدار نتیجه‌ی AUC گزارش شده حدوداً برابر با ۸۰ درصد می‌باشد و این مقدار AUC هم‌چنان یک مقدار معتبر و بسیار خوب در زمینه‌ی الگوریتم‌های شناسائی و «دسته‌بندی‌کننده»<sup>۱</sup> محسوب می‌گردد.

## ۴.۵ آزمایش پیچیدگی زمانی

در این قسمت، قصد داریم تا میزان زمان مصرفی الگوریتم پیشنهادی را به ازای افزایش تعداد داده‌ها مورد بررسی قرار دهیم. بدین‌منظور در ابتدا، یک مجموعه‌داده‌ی مصنوعی را مانند قبل، با تعداد ۲۰۰ هزار داده در ۲۰ بعد تهیه می‌کنیم. حال به ازای نرخ‌های نمونه‌برداری ۱۰ درصد تا ۱۰۰ درصد و با طول گام ۱۰ درصد، اقدام به نمونه‌برداری کرده و به هر کدام از مجموعه‌داده‌های حاصل، تعداد ثابت ۲۰۰ داده‌ی پرت تزریق می‌نماییم. علت استفاده از یک مجموعه‌داده‌ی ثابت در ابتدای امر و اقدام به نمونه‌برداری‌های مکرر،

<sup>1</sup> Classifiers

آن است که قصد داریم تا آزمایشات مربوط به پیچیدگی زمانی را با توجه به مجموعه داده های دارای خصوصیات پایه‌ی مشترک، اعم از مکان (میانگین) و شاکله‌ی (ماتریس کوواریانس) یکسان به ازای هر خوش انجام دهیم. اکنون الگوریتم پیشنهادی را بر روی هر یک از مجموعه داده های حاصل اجرا کرده و زمان مصرفی را اندازه می‌گیریم. شکل ۳.۵، نمودار زمان مصرفی نسبت به افزایش تعداد داده ها را نشان می‌دهد.

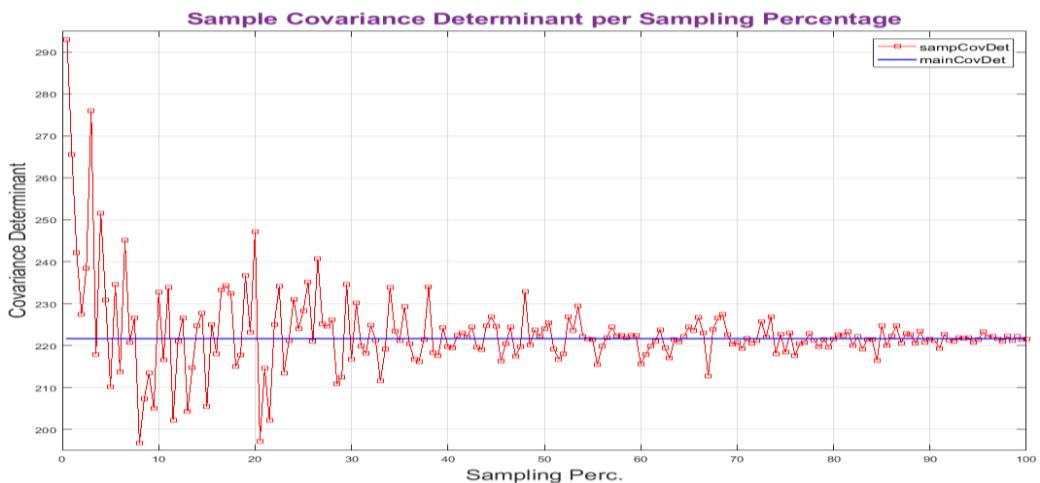


شکل ۳.۵ نمودار زمان مصرفی به ازای افزایش تعداد داده های یک مجموعه داده (منحنی سبزرنگ) و بهترین خطی که می تواند این منحنی را برازش نماید (خط آبی رنگ). از این شکل پیداست که نمودار پیچیدگی زمانی الگوریتم پیشنهادی، خطی و از مرتبه  $O(n)$  می باشد. به طوری که  $n$ ، همان تعداد داده های مجموعه داده است.

از شکل ۳.۵، پر واضح است که زمان مصرفی روش پیشنهادی به ازای افزایش تعداد داده ها، یک رفتار خطی را از خود نشان می دهد و در نتیجه می توان مطابق آن چه در فصل پیشین و در قسمت مربوط به پیچیدگی الگوریتم پیشنهادی مطرح شد، اذعان نمود که پیچیدگی زمانی روش پیشنهادی ما، خطی و از مرتبه  $O(n)$  می باشد.

## ۵.۵ آزمایش نرخ نمونهبرداری

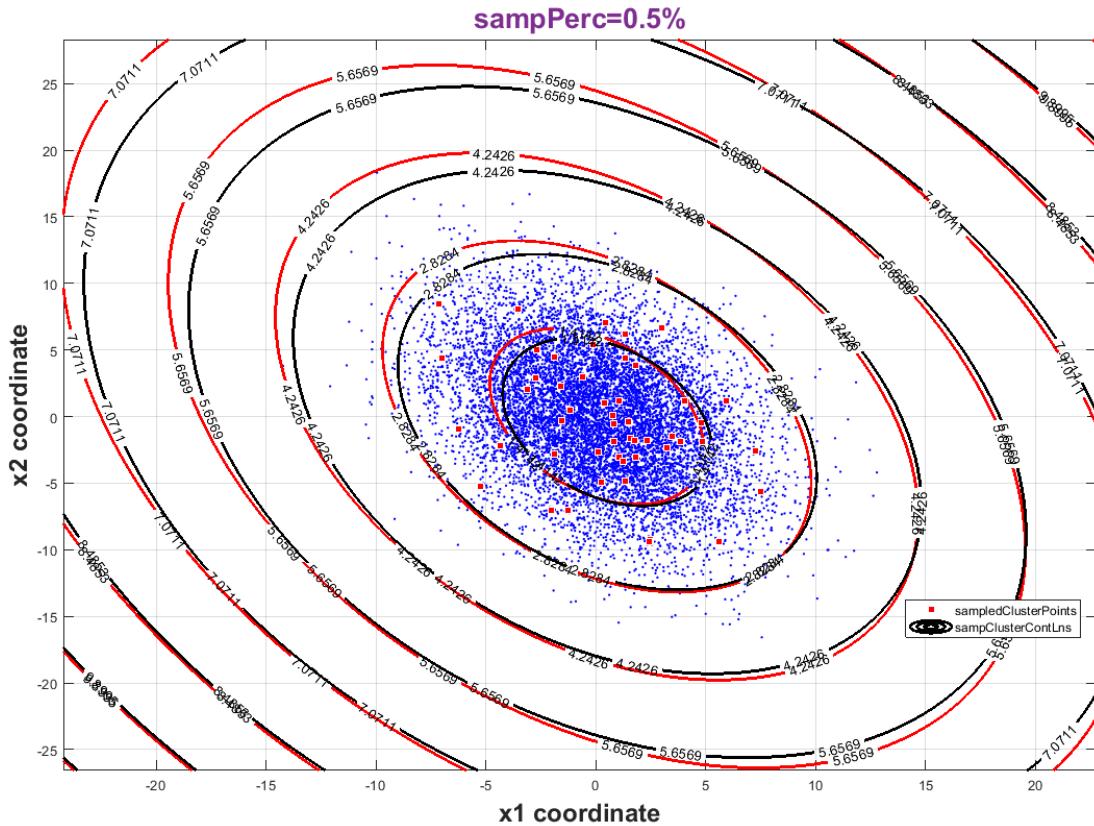
در این قسمت، قصد داریم تا تغییرات دترمینان ماتریس کوواریانس یک خوشه‌ی نمونهبرداری شده را بر حسب نرخ‌های نمونهبرداری متفاوت، مورد بررسی قرار دهیم. نرخ نمونهبرداری از خوشه را از میزان  $0,5$  درصد آغاز نموده و با طول گام  $0,5$  درصدی تا نرخ  $100$  درصد که معادل با همان خوشه‌ی اصلی است، پیش می‌رویم. شکل ۴.۵، نمودار تغییرات دترمینان ماتریس کوواریانس به ازای نمونهبرداری‌های مختلف از یک خوشه با توزیع گاویین را نشان می‌دهد. خط افقی آبی‌رنگ هم نشان‌گر مقدار ثابت دترمینان به ازای ماتریس کوواریانس خوشه‌ی اصلی است که تنها جهت سهولت در مقایسه‌ی مقادیر دترمینان مربوط به نمونهبرداری با دترمینان اصلی ترسیم شده است.



شکل ۴.۵ نمودار تغییرات دترمینان ماتریس کوواریانس یک خوشه‌ی نمونهبرداری شده به ازای نرخ‌های نمونهبرداری از میزان  $0,5$  درصد آغاز شده و با طول گام  $0,5$  درصدی تا میزان نرخ  $100$  درصد که معادل با همان خوشه‌ی اصلی است، پیش می‌رود. خط افقی آبی‌رنگ هم نمایان‌گر مقدار ثابت دترمینان خوشه‌ی اصلی است که تنها جهت سهولت در مقایسه‌ی مقادیر دترمینان نمونهبرداری شده با مقدار دترمینان اصلی رسم شده است.

از نمودار شکل ۴.۵ می‌توان دریافت که با افزایش نرخ نمونهبرداری از یک خوشه، دترمینان ماتریس کوواریانس خوشه‌ی نمونهبرداری شده به دترمینان ماتریس کوواریانس خوشه‌ی اصلی، نزدیک و نزدیک‌تر می‌شود. به عبارت دیگر، میزان تغییرات یا همان واریانس مقادیر دترمینان نمونهبرداری شده، با افزایش نرخ نمونهبرداری کاهش یافته و به سمت صفر می‌کند. اما مشاهده می‌شود که میزان دترمینان به ازای نرخ‌های نمونهبرداری خیلی کم هم چندان نسبت به دترمینان اصلی متفاوت نمی‌باشد. به این معنی که به

ازای نرخ نمونهبرداری مثلاً ۵٪ درصد، مشاهده می‌کنیم که مقدار دترمینان ماتریس کوواریانس خوشهی نمونهبرداری شده حدوداً برابر با ۳۰۰ و مقدار مربوط به خوشهی اصلی هم حدوداً برابر با ۲۲۰ می‌باشد.



شکل ۵.۵ وضعیت منحنی‌های تراز ماهالانوبیس به ازای یک خوشهی دومتغیره با توزیع گاواسین و خوشهی نمونهبرداری شده از آن با نرخ نمونهبرداری برابر با ۵٪ درصد. نقاط ریز آبی رنگ و منحنی‌های قرمزرنگ، به ترتیب نشان‌گر داده‌ها و منحنی‌های تراز ماهالانوبیس خوشهی اصلی می‌باشند. نقاط قرمز مربعی‌شکل و منحنی‌های سیاهرنگ هم به ترتیب نمایان‌گر داده‌های نمونهبرداری شده از خوشهی اصلی و منحنی‌های تراز ماهالانوبیس این خوشهی نمونهبرداری شده می‌باشند.

حال اگر منحنی‌های تراز ماهالانوبیس برابر با  $\sqrt{p} \times k$  به ازای  $k = 1, 2, \dots, 7$  را برای دو خوشهی نمونهبرداری شده و نیز خوشهی اصلی رسم نمائیم (در اینجا  $p$  معادل با تعداد ابعاد دادگان می‌باشد)، مشاهده خواهیم نمود که این منحنی‌ها شدیداً با یکدیگر قربت دارند. شکل ۵.۵ یک خوشهی دومتغیره با توزیع گاواسین را نشان می‌دهد که با نرخ ۵٪ درصد از آن نمونهبرداری شده است. داده‌های خوشهی اصلی در قالب نقاط ریز آبی رنگ و داده‌های مربوط به خوشهی نمونهبرداری شده نیز در قالب مربع‌های کوچک قرمزرنگ نشان داده شده‌اند. منحنی‌های تراز قرمزرنگ، مربوط به خوشهی اصلی و منحنی‌های تراز مشکی نیز متعلق به خوشهی نمونهبرداری شده می‌باشند. همان‌طور که از این شکل پیداست، منحنی‌های تراز دو

خوشی مربوطه، به ازای فاصله‌ی ماهالانوبیس  $\sqrt{2} \times 1$  (که طبق قانون ۳-سیگما، معادل با همان مرز مستحکم ۱-سیگما می‌باشد که حدود ۶۸ درصد داده‌های یک خوشی چندمتغیره با توزیع گاووسین را پوشش می‌دهد) بسیار به هم نزدیک بوده و البته که هر دوی این منحنی‌ها در محدوده‌ی استحفاظی خوشی اصلی قرار گرفته‌اند.

در این جا ذکر این نکته ضروری می‌نماید که ما در تمامی آزمایشات انجام شده، مقدار مرز مستحکم مجاز برای تعلق داده‌ها به یک خوش را برابر با همان مرز ۱-سیگما (که معادل با شعاع ماهالانوبیس  $\sqrt{p} \times 1$  می‌باشد) در نظر گرفته‌ایم. چرا که مقدار این مرز، نه خیلی کم است که به دنبال آن، تعداد ریزخوشه‌هایی که به مرور زمان تشکیل خواهند شد بیشتر از حد ممکن شده و سبب افزایش بار محاسباتی الگوریتم پیشنهادی گردند؛ و نه هم مقدار این مرز خیلی زیاد می‌باشد که به موجب آن، خطر جذب داده‌های پرت به خوشه‌های نرمال در گذر زمان افزایش یابد که در نتیجه‌ی آن، خطای FN<sup>۱</sup> زیاد خواهد شد. به این پدیده که داده‌های پرت در قالب داده‌های نرمال، تغییر چهره می‌دهند، اثر پوشش نیز گویند.

---

<sup>۱</sup> False Negative

۶

## فصل ششم

# جمع‌بندی و نتیجه‌گیری

## ۱.۶ جمع‌بندی و نتیجه‌گیری

در این پایان‌نامه، در ابتدا تعریفی کلی از داده‌ی پرت ارائه نمودیم و مهم‌ترین انواع آن را بر شمردیم و نیز به اهمیت کشف این گونه داده‌ها در کلان‌داده‌ها پرداختیم. سپس به مهم‌ترین انواع مدل‌های کشف داده‌های پرت اشاره نموده و برخی از چالش‌های موجود در این زمینه را بررسی کردیم. بعد از آن، مروری بر کارهای انجام‌شده انجام دادیم و به برخی از مهم‌ترین روش‌های پایه جهت کشف داده‌ی پرت در مجموعه‌داده‌ای با حجم نرمال و مقیاس بزرگ اشاره کردیم.

در ادامه، روشی را جهت کشف داده‌های پرت محلی در کلان‌داده‌ها ارائه نمودیم که بنای آن بر خوش‌بندی مبتنی بر چگالی بود و در یک رویه‌ی مقیاس‌پذیر، قطعه‌های داده را به صورت متوالی بررسی کرده و مدل خوش‌بندی را پیوسته به روز می‌نمود. در نهایت، پس از بررسی کلیه‌ی قطعه‌های داده‌ها، مدل خوش‌بندی نهائی را به دست آورده و با توجه به آن، به هر داده، امتیازی مبنی بر میزان پرت‌بودن نسبت دادیم.

روش پیشنهادی را بر روی مجموعه‌داده‌های واقعی و مصنوعی متنوع، آزمایش نمودیم و با برخی از روش‌های مطرح در زمینه‌ی کشف داده‌های پرت، مقایسه کردیم. در ابتدا، ارزیابی‌های ما بر روی مجموعه‌داده‌های واقعی متنوع و تعداد اندکی از مجموعه‌داده‌های مصنوعی، نشان‌گر آن بودند که الگوریتم پیشنهادی، از اثربخشی بالائی نسبت به روش‌های رقیب برخوردار می‌باشد و در عین این‌که نیازی به این ندارد که تمامی مجموعه‌داده را به یکباره مشاهده نماید و آن را به صورت قطعه‌قطعه و در یک رویه‌ی مقیاس‌پذیر بررسی می‌کند، با دقت بسیار بالائی، قادر به شناسائی داده‌های پرت از نرمال می‌باشد. در ادامه، روش پیشنهادی را بر روی مجموعه‌داده‌های مصنوعی گوناگون بسیاری آزمایش نمودیم تا بررسی نمائیم که میزان دقت آن در تشخیص ناهنجاری، چگونه با افزایش تعداد داده‌ها، تعداد ابعاد و تعداد داده‌های پرت تزریق شده، تغییر می‌نماید. این آزمایشات نیز نشان از آن داشتند که الگوریتم پیشنهادی از بازدهی بالائی در مورد مجموعه‌داده‌های با مقیاس و ابعاد بالا برخوردار می‌باشد و نیز در موقعي که میزان آلوگی دادگان بسیار بالاست هم از تحمل بالائی در تمیزدادن داده‌های پرت از نرمال برخوردار می‌باشد.

نتایج آزمایشات در مورد پیچیدگی الگوریتم نیز نشان می‌دهند که پیچیدگی زمانی روش پیشنهادی، خطی و از مرتبه‌ی  $O(n)$  می‌باشد، به طوری که  $n$ ، معرف تعداد داده‌های مجموعه‌داده است. همین‌طور، آزمایشات مربوط به نرخ نمونه‌برداری نیز تأیید می‌کنند که به ازای نرخ‌های نمونه‌برداری خیلی پایین هم می‌توان انتظار دقت عالی را از الگوریتم پیشنهادی داشت.

در روش پیشنهادی حاضر، ما در هر قسمت از الگوریتم که نیاز به استفاده از الگوریتم DBSCAN داشته‌ایم، جهت جلوگیری از محاسبه‌ی مجدد فاصله‌ها به ازای هر داده‌ی مورد بررسی، از ماتریس فاصله‌ی دوبه‌دوی داده‌ها در ابتدای امر بهره بردۀ‌ایم. لذا در آینده قصد داریم تا این روش را به گونه‌ای بهبود دهیم که دیگر نیازی به دانستن ماتریس فاصله‌ی دوبه‌دوی داده‌ها در ابتدای امر نداشته باشیم. به عبارت دیگر، می‌توانیم از نسخه‌ای از DBSCAN استفاده نمائیم که پیچیدگی حافظه‌ی آن به جای  $O(n^2)$ ، برابر با  $O(n)$  باشد.

## ٧ منابع و مراجع

- [1] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” *ACM Comput. Surv.*, vol. 41, no. 3, p. 15, 2009.
- [2] J. Han, J. Pei, and M. Kamber, *Data mining: concepts and techniques*. Elsevier, 2011.
- [3] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, “LOF: Identifying Density-Based Local Outliers,” *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*. pp. 1–12, 2000.
- [4] C. C. Aggarwal, “Data Mining: The Textbook,” *Springer International Publishing*. p. 746, 2015.
- [5] V. Hodge and J. Austin, “A survey of outlier detection methodologies,” *Artif. Intell. Rev.*, vol. 22, no. 2, pp. 85–126, 2004.
- [6] J. Yin, Q. Ho, and E. P. Xing, “A scalable approach to probabilistic latent space inference of large-scale networks,” in *Advances in neural information processing systems*, 2013, pp. 422–430.
- [7] S.-H. Teng, “Scalable algorithms for data and network analysis,” *Found. Trends® Theor. Comput. Sci.*, vol. 12, no. 1–2, pp. 1–274, 2016.
- [8] P. S. Bradley, U. M. Fayyad, and C. Reina, “Scaling Clustering Algorithms to Large Databases.,” in *KDD*, 1998, pp. 9–15.
- [9] D. M. Hawkins, *Identification of outliers*, vol. 11. Springer, 1980.
- [10] C. C. Aggarwal, *Outlier analysis*, vol. 9781461463. 2013.
- [11] B. Tang and H. He, “A local density-based approach for outlier detection,” *Neurocomputing*, vol. 241, pp. 171–180, 2017.
- [12] S. Wu and S. Wang, “Information-theoretic outlier detection for large-scale categorical data,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 3. pp. 589–602, 2013.
- [13] X. H. Dang, B. Micenková, I. Assent, and R. T. Ng, “Local outlier detection with interpretation,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8190 LNAI, no. PART 3. pp. 304–320, 2013.
- [14] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, “LoOP: local outlier probabilities,” *Proceedings of the 18th ACM conference on Information and knowledge management*. pp. 1649–1652, 2009.
- [15] D. Pokrajac, A. Lazarevic, and L. J. Latecki, “Incremental local outlier detection for data streams,” in *Computational Intelligence and Data Mining, 2007. CIDM 2007. IEEE Symposium on*, 2007, pp. 504–515.

- 
- 
- [16] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation forest,” in *Data Mining, 2008. ICDM’08. Eighth IEEE International Conference on*, 2008, pp. 413–422.
  - [17] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, “Estimating the support of a high-dimensional distribution,” *Neural Comput.*, vol. 13, no. 7, pp. 1443–1471, 2001.
  - [18] T. de Vries, S. Chawla, and M. E. Houle, “Density-preserving projections for large-scale local anomaly detection,” *Knowledge and Information Systems*, vol. 32, no. 1, pp. 25–52, 2012.
  - [19] Q. Cai, H. He, and H. Man, “Spatial outlier detection based on iterative self-organizing learning model,” *Neurocomputing*, vol. 117, pp. 161–172, 2013.
  - [20] E. W. Forgy, “Cluster analysis of multivariate data: Efficiency vs. interpretability of classifications,” *Biometrics*, vol. 21, pp. 768–769, 1965.
  - [21] 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.
  - [22] J. MacQueen, “Some methods for classification and analysis of multivariate observations,” in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, 1967, vol. 1, no. 14, pp. 281–297.
  - [23] P. C. Mahalanobis, “On the generalized distance in statistics,” 1936.
  - [24] J. Leskovec, A. Rajaraman, and J. D. Ullman, *Mining of massive datasets*. Cambridge university press, 2014.
  - [25] “DBSCAN.” [Online]. Available: <https://en.wikipedia.org/wiki/DBSCAN>, last edited on 25 March 2018.
  - [26] L. KPFRS, “On Lines and Planes of Closest Fit to Systems of Points in Space,” in *Proceedings of the 17th ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems (SIGMOD)*, 1901.
  - [27] “Principal Component Analysis.” [Online]. Available: [https://en.wikipedia.org/wiki/Principal\\_component\\_analysis](https://en.wikipedia.org/wiki/Principal_component_analysis), last edited on 8 April 2018.
  - [28] R. C. Eberhart and J. Kennedy, “Particle swarm optimization,” in *Proceedings of the IEEE international conference on neural networks*, 1995, vol. 4, pp. 1942–1948.
  - [29] P. J. Rousseeuw, “Least Median Squares Regression.” journal of the American Statistical Association, 1984.
  - [30] M. Hubert and M. Debruyne, “Minimum covariance determinant,” *Wiley Interdiscip. Rev. Comput. Stat.*, vol. 2, no. 1, pp. 36–43, 2010.
  - [31] “UCI Machine Learning Repository.” [Online]. Available: <http://www.ics.uci.edu/mlearn/mlrepository.html>.
  - [32] “ODDS.” [Online]. Available: <http://odds.cs.stonybrook.edu/>.

## **Abstract**

---

---

# **Abstract**

According to the fact that nowadays, the volume of data collected and stored in different ways is increasing so rapidly, thus commonly used software methods for processing and managing this very large amount of data are not efficient and, as such, this type of data is categorized as Big Data. The huge size and high dimensionality of Big Data make it more necessary to use novel techniques and technologies that can deal with such amount of data. One of the major challenges in the field of Big Data is Outlier detection. Detecting and removing outliers from data could be considered as a preprocessing step for data analysis, which attempts to find those rare data points that are behaving in an unacceptable manner than the rest of the common data points.

In this thesis, we are trying to provide a method for detecting local outliers in Big Data, which is based on scalable density-based clustering. Considering the fact that a Big Data does not have the capability to be fit into the RAM at once, so we have to process it chunk by chunk, as each chunk can be both inserted and processed in the main memory at the same time. Then, for each chunk, we update the information of the clustering model. Throughout the clustering procedure, our effort is to not let the outliers play any role in formation and updating clusters. At the end of scalable clustering, we obtain the structure of the final clusters. Subsequently, using a suitable criterion, we will give each data point an outlierness score. Our evaluations on real-life and synthetic datasets exhibit that our proposed method has a low linear time complexity and is effective in comparison with those traditional methods that need to observe the whole data at once, not chunk by chunk, and also is efficient in severe conditions.

**Key Words:** Big Data, Outlier detection, Scalable, Density-based.



**Amirkabir University of Technology  
(Tehran Polytechnic)**

**Department of Computer Engineering and Information Technology**

**MSc Thesis**

**Local Outlier Detection in Big Data by  
a Density-based Method**

**By  
Sayyed Ahmad Naghavi Nozad**

**Supervisor  
Dr. Maryam AmirHaeri**

**March 2018**