یک روش شناسایی بخش مجزای سریع بر پایه KNN و الگو گرفته از MST

چکیده

پایگاه های داده ی دنیای واقعی امروزی معمولا شامل میلیون ها مورد با هزاران حوزه می شوند. به عنوان یک نتیجه، روش های شناسایی بخش جدای سنتی توزیع بنیان دارای توانایی های محدود شده ی بسیاری هستند و رویکردهای جدید همسایه های نزدیکترین K بنیان، محبوب تر شده اند. اما، مشکل با این روش های همسایه های نزدیکترین K بنیان این است که آنها بسیار به مقدار K حساس هستند(می توانند رتبه بندی متفاوتی برای بخش های مجزای برتر n داشته باشند)، از نظر محاسباتی برای مجموعه های داده بسیار پر هزینه هستند و در کل در اینکه آیا آنها برای مجموعه های ابعاد زیاد به خوبی کار می کنند یا نه شک وجو دارد. در این مقاله برای تا حدی دور زدن این مشکلات،یک فاکتور جدید بخش مجزای سراسری و یک فاکتور جدیدی بخش مجزای محلی و یک الگوریتم شناسایی بخش مجزای کارآمد بر مبنای این دو فاکتور مطرح کردیم که به راحتی پیاده سازی می شود و با راه حل های موجود می تواند عملکردهای رقابتی را بهبود ببخشد.آزمایشات انجام شده روی هر دو مجموعه های داده ی ترکیبی و واقعی، کارآمدی روش ما را نشان می دهند.

کلید واژه ها: تشخیص نقاط پرت بر اساس فاصله .تشخیص بیرونی بر اساس چگالی .تشخیص نقاط پرت مبتنی بر خوشه .حداقل خوشه بندی مبتنی بر درخت .جستجوی تقریبی k-نزدیکترین همسایگان

1. مقدمه

شناسایی بخش مجزا با هدف گذاری برای کشف مشاهدات بسیار دور شده از سایر مشاهدات (به میزانی که شک هایی به وجود آید مبنی بر این که این مشاهات بوسیله ی مکانیزم متفادتی ایجاد می شوند) تبدیل به یک کار داده کاوی مهم شده است. شناسایی بخش مجزا با مورد استفاده قرار گرفتن در حوزه های متعدد مختلفی مانند شناسایی نفوذ برای امنیت سایبری، شناسایی تقلب برای کارت های اعتباری، بیمه و مالیات، شناسایی سریع شیوع بیماری در حوزه ی پزشکی، شناسایی خطا در شبکه ی سنسور برای نظارت سلامت، ترافیک، وضعیت ماشین، هوا، آلودگی و مراقبت و غیره ، منافع بسیار زیادی را تولید کرده و در سال های اخیر روش های بسیاری برای این هدف ایجاد شده است، که از میان آنها می توان رویکردهای توزیع بنیان، عمق بنیان، فاصله بنیان، تراکم بنیان و خوشه بندی بنیان را نام برد.

آخرین الگوریتم شناسایی بخش مجزا بر پایه ی نزدیک ترین k همسایه(مانند تعدادی از روش های فاصله بنیان و تراکم بنیان) راه های مختلفی برای فیلتر کردن داده ی عادی و موقعیت یابی تعداد کم بخش های مجزا نشان داده اند. در حالیکه این روش ها برای پیاده سازی ساده هستند اما سایر جنبه های مربوط به الگوریتم ها نیز ارزش تحقیقات بیشتر را دارند. اولا، این روش ها معمولا تنها n بخش مجزای برتر را با دو مقدار نشان می دهند. یکی از این مقادیر فاکتور بخش مجزا (به عنوان امتیاز نیز در این مقاله به آن اشاره شده) و دیگری رتبه بندی نقاط براساس امتیازات. بنابراین، روش های مختلف می توانند برای n بخش مجزای برتر، رتبه بندی های مختلفی داشته باشند. دوما، مشاهده شده که روش های شناسایی بخش مجزا بر مبنای نزدیک ترین k همسایه به پارامتر k حساس هستند و تغییری کوچک در k می تواند منجر به تغییرات در امتیازات و متقابلا رتبه بندی شود. به عنوان یک نتیجه، به جز برای بخش های مجزای خیلی قوی که امتیازات در آنها متمایز هستند، رتبه بندی نیز به k حساس است. سوما، برای مجموعه داده های بزرگ مدرن با N واحد داده، زمان اجرای $O(N \log N)$، که برای جستجوی دقیق همسایه های k بوده، باید به طور معناداری بهبود داده شود. در نهایت، در فضا با بعد بالا، نقاط داده به طور برابر به یکدیگر

نزدیک می شوند و مشکلی که "نفرین ابعاد" نامیده می شود را به وجود می آورند و این مسئله وجود دارد، چه ایده ی این تعاریف بخش مجزا برای داده ی بعد زیاد هنوز معنادار باشند و چه نباشد.

در این مقاله یک روش شناسایی بخش مجزای الهام گرفته از درخت پوشای کمینه و KNNبنیان برای بر طرف کردن این چالش ها و برای تحقیق در مورد اثر ابعاد روی این الگوریتم های شناسایی بخش مجزای K-NN بنیان مطرح کرده ایم که کار ارائه شده در 9 را ادامه می دهد و هم از نظر محاسباتی کارآمد است و به شایستگی آخرین روش های شناسایی بخش مجزا می باشد.اساسا، روش ما با یافتن نزدیک ترین k همسایه برای هر نقطه ی داده شروع می کند. در ادامه یک MST کوچک برای هر نقطه ی داده و نزذیک ترین همسایه های k آن ساخته می شود. در نهایت تعداد کمی از بخش های مجزای با استفاده از امتیازهای بخش مجزای پیشنهاد شده ی ما، نسبتا در داخل MST کوچک شناسایی می شوند. اینکه یک نقطه ی خوب از روش های شناسایی بخش مجزای خوشه بندی بنیان MST هنگام خوشه بندی در حساب کردن فاصله ی بین نقاط داده در MST و از همین رو کم کردن حساسیت امتیازات به k قرار دارد (به جای فاصله تا نزدیکترین همسایه (ها)به k یا k امین همسایه(ها))، بر مبنای مشاهدات است. اما، ساخت یک MST دقیق، نیازمند زمان اجرای درجه ی دو است و از نظر محاسباتی برای مجموعه داده های بزرگ بسیار گران است. بنابراین، هدف این پیوندزنی (برای مثال شناسایی بخش مجزای خوشه بندی بنیان و شناسایی بخش مجزای KNN بنیان) افزایش مقاومت و ثبات نتایج شناسایی و کاهش چشمگیر زمان محاسباتی برای روند شناسایی بخش مجزا بر پایه ی خوشه بندی MST است. اولین مشارکت ما در این مقاله، دو امتیاز ارائه شده ی جدید بخش مجزای KNN بنیان و با الهام از MST (یک سراسری و یک محلی) و یک روش شناسایی بخش مجزای ایجاد شده بر مبنای آنها است. دومین مشارکت بازدهی چشمگیر گام روش ارائه شده از طریق کاربرد یک تخمین کارآمدِ چارچوب جستجوی نزدیکترین همسایه های k برای روش های شناسایی بخش مجزای KNN بنیان ما است. این موضوع مزیتی برای کاری اولیه برای این تحقیق ارائه شده در 9 است که نیازمند ساخت یک زیر گراف کمینه ی تقریبی که به نمونه ی اصلی آن بسیار نزدیک باشد، است. سومین مشارکت ما، مطالعه ای در مورد مجموعه ای از الگوریتم های شناسایی بخش مجزای کنونی در زمانی است که برای مجموعه داده های با ابعاد بالا استفاده شده اند.

در نهایت، الگوریتم ما برای اینکه تا جای ممکن جامع باشد، ملزومات خاصی در مورد ابعاد مجموعه های داده ندارد و می تواند در شناسایی بخش های مجزای مجموعه داده هایی با ابعاد بالا به کار گرفته شود. تعدادی آزمایش روی هر دو مجموعه داده ی مصنوعی و واقعی، استقامت و کارآمدی رویکرد ارائه شده را در مقایسه با برخی از الگوریتم های شناسایی بخش مجزای مدرن نشان داده است.

باقی این مقاله به صورت پیش رو سازمان دهی شده است. در بخش 2، برخی از کارهای موجود در مورد رویکردهای شناسایی بخش های مجزای مدرن را بررسی کرده ایم. در ادامه رویکردهای پیشنهاد شده ی خودمان را در بخش 3 ارائه می کنیم. در بخش 4 یک مطالعه ی تجربی انجام داده می شود. در نهایت نتیجه گیری در بخش 5 گرفته شده است.

2. کارهای مرتبط

کارهای مرتبط در این مقاله، به سه دسته ی اصلی تقسیم بندی می شود: روش های شناسایی بخش مجزای تراکم و فاصله بنیان، روش های شناسایی بخش مجزا بر مبنای خوشه بندی MST و روش های شناسایی بخش مجزا برای داده با ابعاد بالا.

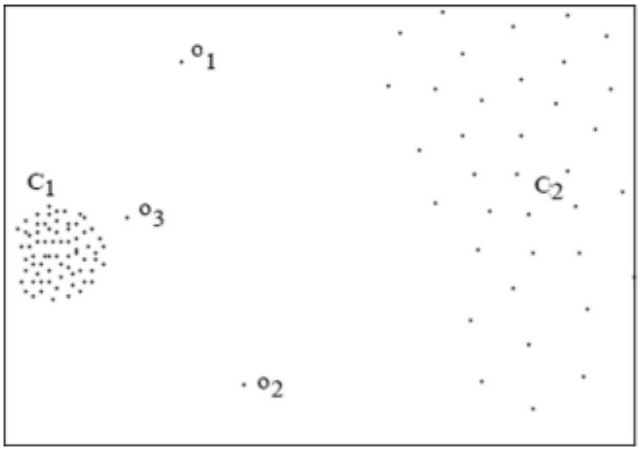
2.1. روش های شناسایی بخش مجزای تراکم و فاصله بنیان

تلاش های زیادی به شناسایی بخش های مجزا تخصیص داده شده است. روش شناسایی بخش مجزای فاصله بنیان در اصل بوسیله ی Knorr and Ng در سال 1998 به عنوان یک پیشرفت در روش های توزیع پایه، ارائه شد. با توجه به مقیاس فاصله ی تعریف شده در یک فضای ویژگی، "یک شیء O در یک مجموعه داده ی T یک بخش مجزای DB(p,D) است اگر حداقل یک شکاف p اشیاء در T بزرگتر از فاصله ی D از O باشد"، که اصطلاح بخش مجزای DB(p,D) یک نماد مختصر برای بخش مجزای فاصله بنیان (بخش مجزای DB) شناسایی شده با استفاده از پارامترهای p و D است. KNN بنیان های گوناگونی برای متناسب بودن با اهداف مختلف عملی و تئوری ایجاد شده

اند. "با توجه به دو عدد صحیح (kوn)، بخش های مجزای فاصله بنیان، واحدهایی از داده است که فاصله یآنها تا نزدیک ترین همسایه ی k ام آنها در میان n فاصله ی بزرگ است" (که در ادامه به عنوان "DB-MAX" ذکر شده است) و "با توجه به دو عدد صحیح (kوn)، بخش های مجزای فاصله بنیان، واحدهای داده ای هستند که میانگین فاصله ی آنها به نزدیک ترین k همسایهدر میان n فاصله ی بزرگ است" (که در ادامه به عنوان "DB" ذکر شده است).

اگر چه روش های شناسایی بخش مجزای فاصله بنیان به صورت ساده و ظریف و خوب برای مجموعه داده هایی که شامل یک خوشه یا بیشتر و با تراکم های مشابه کار می کنند و می توانند بخش های مجزای سراسری محور را شناسایی کنند اما برخی از مجموعه های داده ی دنیای واقعی معمولا ساختارهای پیچیده ای دارند. یک وضعیت کلاسیک که این کمبود را نشان می دهد، در شکل 1 نشان داده می شود که o1 و o2 بخش های مجزای سراسری هستند و می توانند به سادگی بوسیله ی روش های فاصله بنیان شناسایی شوند در حالیکه o3 بخش مجزای محلی است و نمی توانند به سادگی بوسیله ی روش های فاصله بنیان شناسایی شوند.

در سال 2000 برای اداره کردن این وضعیت، Breunig et al بوسیله ی نشان دادن یک نشانگر برای هر واحد داده (فاکتور بخش مجزا (LOF) نامیده می شد) که نسبت بین تراکم محلی یک شیء و میانگین اشیاء نزدیک ترین k همسایه ام آنها می باشد، در تحقیق شناسایی بخش مجزای تراکم بنیان پیش گام شدند. روش LOF به این طریق کار می کند که در ابتدا LOF را برای هر شیء محاسبه می کند، سپس رتبه بندی نقاط داده بر اساس مقادیر LOF آنها مشخص می شود و در نهایت اشیاء با n مقدار بالای LOF به عنوان بخش مجزا بازگردانده می شوند



شکل 1:یک مثال کلاسیک از بخش مجزای محلی

به دنبال ایده ی فاکتور بخش مجزای محلی، چندین بسط و اصلاحیه برای مدل پایه ی LOF ارائه شده است. در سال 2002، Tang et al یک فاکتور بخش مجزای اتصال بنیان (COF) را به منظور اداره کردن خصوصیت "جدا بودن" بخش های مجزا پیشنهاد کردند. "جدا بودن" به تراکم کم دلالت دارد اما تراکم کم همیشه به "جدا بودن" دلالت نمی کند. با توجه به نقطه ی داده ی o ، و نزدیک ترین k همسایه، اولین هزینه در تعریف هزینه فاصله از o تا نزدیک ترین همسایه ی آن است. به طور کلی، هزینه ی i ام ($i \le k$) برابر با کوچکترین فاصله از o و نزدیک ترین (i-1) شیء آن تا بهقی k-i شیء در همسایگی است. در نهایت، COF نسبت تعریف هزینه ی نقاط داده به میانگین هزینه ی نقاط داده ی نزدیک ترین k همسایه، می باشد. در سال 2003، et al Papadimitriou طرح شناسایی بخش مجزای محلی دیگری با نام انتگرال بخش مجزای محلی (LOCI) را بر مبنای مفهوم فاکتور انحراف چند دانه بودن (MDEF) پیشنهاد کرد. تفاوت اصلی میان LOF و LOCI این است که MDEF در LOCI از همسایگی های ε به جای نزدیک ترین k همسایه، استفاده می کنند. در سال 2004، Sun and Chawla یک معیار بخش مجزای محلی فضایی با نام SLOM مطرح کرد. در سال 2006، Jin et al روش INFLO که اجتماع داده ی نقاط نزدیکترین k همسایه و معکوس نزدیک ترین همسایه های آن را به منظور بدست آوردن میزان جدایی لحاظ میکرد را ارائه کرد. نزدیک ترین همسایگی معکوس برای نقطه ی داده ی p تعریف می شود تا شامل همسایگیِ نزدیکترین k همسایه ای شود که p برای آنها در میان نزدیکترین k همسایه است. به همین روش، INFLO تراکم p را با

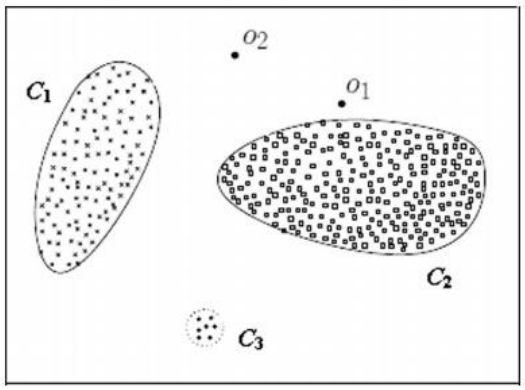
میانگین تراکم های اشیاء در اجتماع داده را به عنوان میزان جدایی مقایسه می کند..Zhang et al با توجه به اینکه داده ی دنیای واقعی معمولا دارای توزیعی پراکنده است، یک تعریف بخش مجزای جدید با نام فاکتور بخش مجزای فاصله بنیان محلی (LDOF) در جهت شناسایی بخش های مجزا در مجموعه داده های پراکنده مطرح کرد. LDOF نسبت بینِ میانگین فواصل از یک نقطه ی داده تا نزدیک ترین k همسایه ی آن و میانگین فواصل دو به دو در میان این k+1 نقطه ی داده است، و به همین طریق، در جه ای که یک شیء از سیستم همسایگی اش منحرف شده را بدست می آورد. در سال 2013، Huang et al، رویکردی جدید برای شناسایی بخش مجزا با نام RBDA و بر مبنای یک معیار رتبه بندی که روی این سوال که آیا یک نقطه نسبت به نزدیک ترین همسایه های خود نزدیک نرین است یا نه متمرکز شده، مطرح کرد. RBDA با حذف مشکل محاسبه ی تراکم در همسایگی یک نقطه، بخش های مجزا را بر پایه ی محاسبه ی مرتبه ی نقطه ی در میان همه ی نزدیک ترین k همسایه ی آن مشخص می کند. اما متاسفانه چنین کاری دارای هزینه ی محاسباتی بالایی است.

2.2. الگوریتم هایی بر مبنای خوشه بندی MST

امتیازات بخش مجزای فاصل بنیان و نیز تراکم بنیان نسبت به تنظیمات پارامترهایشان حساس هستند. این موضوع بوسیله ی مجموعه داده ی دو بعدی نشان داده شده در شکل 2 می تواند نمایش داده شود. برای روش های شناسایی بخش های مجزای فاصله بنیان، اگر نزدیک ترین k=6 همسایه در نظر گرفته شود، همه ی نقاط داده در خوشه ی c3 به عنوان بخش مجزا شناسایی نخواهند شد در حالیکه اگر k=7 باشد، همه ی نقاط داده در خوشه ی c3 به عنوان بخش مجزا لحاظ می شوند. مشکلاتی مشابه برای روش های شناسایی بخش مجزای تراکم بنیان وجود دارد. وضعیت برای شناسایی بخش های مجزا در فضای ویژگی با ابعاد بالا می تواند بدتر باشد چون نقاط داده در آنجا به سادگی پدیدار نمی شوند.

اینجا محلی است که الگوریتم های خوشه بندی بنیان، می توانند معنای بیشتری داشته باشند. نگرانی اصلی الگوریتم های خوشه بندی به عنوان یک ابزار بسیار مهم داده کاوی، یافتن خوشه ها از طریق بهینه سازی برخی معیارها مانند

کمینه کردن فاصله ی خوشه ی درونی و بیشینه کردن فاصله ی خوشه ی داخلی است. واحد های داده در گروه های کوچک، به عنوان یک محصول فرعی معمولا می توانند به عنوان بخش های مجزا (دماغه) ای ذکر شوند که باید برای افزایش اطمینان خوشه حذف گردند. الگوریتم های کلاسیک خوشه بندی مانند الگوریتم میانگین های k و PAM، به گروه بندی نقاط داده در پیرامون برخی "مراکز" وابسته هستند و در زمانی که مرز های خوشه ها غیر عادی باشند به خوبی کار نمی کنند. به عنوان یک جایگزین، روش هایی با بنیان تئوری گراف که از طریق الگوریتم های خوشه بندی بر مبنای MST به عنوان نمونه داده شده اند، می توانند خوشه هایی با مرزهای غیر عادی را بیابند.



شکل 2:خوشه های نمونه در مجموعه داده ی دو بعدی

یک درخت پوشای کمینه دارای حداقل وزن کل است در حالیکه یک گراف وزن دار متصل روی مجموعه ای از نقاط داده اما بدون مسیر بسته شده است. اگر یک وزن که دلالت بر فاصله ی بین دو نقطه ی پایانی دارد، به هر لبه تخصیصی داده شود، هر لبه در یک MST کوتاه ترین فاصله بین دو زیر شاخه ای خواهد بود که بوسیله ی آن لبه متصل می شوند. این نکته به عنوان دارایی برشِ MST استناد می شود. بنابراین، حذف بلندترین لبه ها متناسب با انتخاب انفصال ها در جهت شکل دهی خوشه ها است. درخت پوشای کمینه (MST) خوشه بندی بنیان، برای اولین بار به وسیله ی Zahn در سال 1971 ارائه شد و تا کنون به طور گسترده ای مورد مطالعه قرار گرفته است. با توجه به این موضوع، نقاط داده در کوچکترین خوشه هایی که بوسیله ی بریدن بلندترین لبه ها در یک MST شکل گرفته اند به احتمال زیاد می توانند بخش مجزا باشند. چندین روش شناسایی بخش مجزای MST بنیان پیشنهاد شده

است. اما برای مجموعه های داده ی بزرگ و با ابعاد بالای مدرن که در آنها تنها یک مجموعه از N نقطه ی داده ارائه می شود، این الگوریتم های شناسایی بخش مجزای MST بنیان از زمان اجرای درجه دوی ملزوم برای ساخت یک MST رنج می برند.

برای دقت بیشتر محاسباتی، Wang و دیگران یک روش شناسایی بخش مجزای سه مرحله ای کارآمد پیشنهاد دادند (در ادامه به صورت MST+LOF ذکر می شود). در ابتدا، یک ساختار کارآمد از یک درخت پوشای بسیار نزدیک به دخت پوشای کمینه ی مجموعه داده ساخته می شود. دوم، بلندترین لبه ها در درخت پوشای بدست آمده حذف می شوند تا خوشه ها را شکل دهند. براساس این یافته که نقاط داده در خوشه های کوچک به احتمال زیاد همگی می توانند بخش مجزا باشند، این خوشه های کوچک انتخاب می شوند و به عنوان کاندیداهای بالقوه ی بخش مجزا در نظر گرفته می شوند. در نهایت، عوامل مجزا بودن تراکم بنیان، LOF، برای کاندیداهای بالقوه ی بخش مجزا محاسبه و به منظور مشخص کردن بخش مجزای محلی ارزش گذاری می شوند. مزیت اصلی این الگوریتم، بازده محاسباتی آن است.

2.3. بخش مجزا کاوی با ابعاد بالا بر مبنای طرح

الگوریتم های شناسایی بخش مجزایی که تا کنون ارائه شدند، فرضیه هایی ضمنی از داده ی نسبتا با ابعاد کم ایجاد می کنند و فواصل در فضای ابعاد کامل را در جهت یافتن بخش مجزا استفاده می کنند اما، در فضای با بعد بالا، داده نامتراکم است و ایده ی یافتن بخش های مجزای معنادار به طور اساسی پیچیده تر و غیرواضح می شود. جدا از در نظر گرفتن رفتار داده در ابعاد کامل، انحرافات غیر عادی می توانند در برخی زیر فضاها با ابعاد کمتر قرار گیرند. طراحی روش هایی هدفمندتر برای یافتن بخش های مجزایی که مختص به زیرفضای خاص در سوال با بررسی رفتار داده در زیرفضا، ممکن می شود. در حقیقت، در 11 گزارش شده است که بخش های مجزای جذاب تر در پایگاه داده ی آماری بسکتبال NBA98 با استفاده از ویژگی های کمتری بدست آمدند. براساس این مشاهدات، Aggarwal and Yu روش های جدیدی برای شناسایی بخش مجزا ارائه کردند (در ادامه به صورت شناسایی بخش مجزا بر

مبنای طرح (PBOD) ذکر می شود) که اگر در یک طرح با ابعاد کمتر، یک نقطه ی داده در یک منطقه ی محلیِ تراکمِ کمِ غیرعادی حضور داشته باشد،آن را بخش مجزا تعریف می کند. برای مشخص کردن و یافتن چنین طرح هایی، در ابتدا یک گسسته سازی شبکه ی داده اجرا می شود. هر یک از ویژگی های d داده به محدوده های ϕ تقسیم شده اند و از این رو هر محدوده شامل یک کسر f=1/ ϕ از سوابق می شود. برای یک مکعب K بعدی که بوسیله ی برداشتن محدوده های شبکه از ابعاد مختلف k<=d ایجاد می شود، ضریب پراکندگی S(c) برای مکعب C به صورت زیر محاسبه می شود:

$$S(C) = \frac{n(C) - N \cdot f^k}{\sqrt{N \cdot f^k \cdot (1 - f^k)}} \qquad (1)$$

که N در آن تعداد نقاط داده است و n(C) به تعداد نقاط در مکعب k بعدی دلالت دارد. مکعب هایی را که حضور نقطه ها در آن به طور معناداری کمتر از میزان مورد انتظار است، تنها از طریق ضرایب پراکندگی منفی نشان داده می شوند. وقتی چنین الگوهایی مشخص شد، بخش های مجزا به عنوان آن سوابقی که چنین الگوهایی در آنها وجود دارد، تعریف می شوند. یک مشاهده ی جالب این است که چنین طرح هایی با ابعاد کمتر می توانند حتی در مجموعه های داده ای که مقادیر ویژگی از دست رفته دارند نیز کاویده شوند. افزایش نمایی فضای جستجوی طرح های ممکن با ابعاد، مشکل PBOD است. این الگوریتم برای چند صد بعد عملی نیست.

3. یک الگوریتم شناسایی بخش مجزا ی الگو گرفته از MST

از مطالعه ی ما در مورد شناسایی بخش مجزای فاصله بنیان، تراکم بنیان و خوشه بندی بنیان، چندین مشاهده بدست آورده ایم:

اولا، روش های فاصله بنیان، به صورت تئوری بوسیله ی محاسبه ی KNN برای هر نقطه ی داده، محاسبه ی امتیازات بخش مجزای فاصله بنیان برای آنها، رتبه بندی همه ی اشیاء برمبنای امتیازاتشان و در نهایت برگرداندن نقاط داده با n امتیاز بزرگتر به عنوان بخش های مجزا، کار می کند اما هیچ دلیلی وجود ندارد تا فرض کنیم که موضوع همین است. برای مثال، در شکل 3، اگرچه امتیازات بخش مجزای فاصله بنیان می تواند از منظر یک MST،

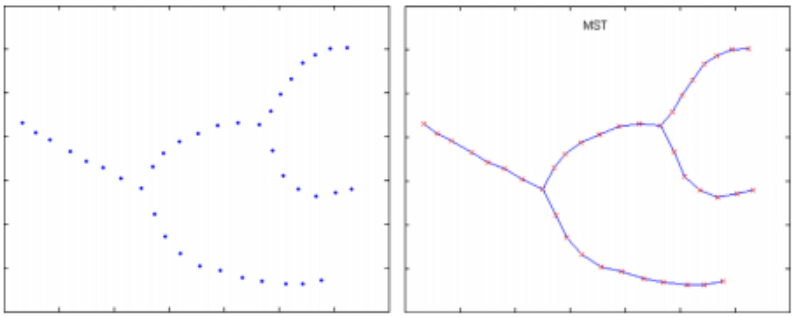
برای این مجموعه داده محاسبه شوداما هیچ بخش مجزای برجسته ای وجود ندارد. متاسفانه، معیار خوشه بندی بر پایه ی MST تمایل به بریدن مجموعه های کوچکِ گره های جدا شده در یک گراف دارد و می تواند در زمانی که هیچ بخش مجزایی وجود ندارد، قسمت بندی بدی را ارائه دهد. این حقیقت می تواند به سادگی از مجموعه داده های نشان داده شده در شکل 3 نشان داده شود. بنابراین، باید راهی برای قضاوت در مورد اینکه آیا بخش های مجزا وجود دارد یا نه وجود داشته باشد. برای انجام دادن چنین کاری، به عنوان تخمین درجه اول، وزن های لبه (برای مثال فواصل لبه) در خوشه ی یک MST می تواند فرض شود که یک توزیع متحدالشکل میانگین متناظر را دنبال می کند و از این رو انحراف استاندارد می تواند محاسبه شود. تناسب انحراف استاندارد روی میانگین می تواند استعمال شود تا در سطحی قضاوت کند که آیا بخش های مجزا وجود دارند یا نه.

دوما، الگوریتم های شناسایی بخش مجزای تراکم بنیان، نمی تواند در شناسایی بخش مجزای سراسری، به خوبی کار کند. برای مثال، مجموعه داده در شکل 4 را در نظر بگیرید. ظاهرا، نقطه ی داده ی A دورترین نقطه از نزدیکترین شش همسایه ی خود است و از این رو باید به عنوان یک بخش مجزای سراسری مشخص شود اما برای k=6، الگوریتم LOF ، بالاترین امتیاز بخش مجزا را به جای نقطه ی A به نقطه ی داده ی B تخصیص می دهد. این مسئله به این علت است که LOF به عنوان یک تناسب محاسبه می شود و نقطه ی داده ی B بالاترین تناسب را برای این k می گیرد. به عنوان یک نتیجه، الگوریتم های LOF بنیان، در شناسایی A به عنوان چشمگیرترین بخش مجزا شکست می خورند. از منظر یک MST، همان طور که در سمت راست شکل 4 نشان داده شده، اگر وزن لبه ی درخت (برای مثال فاصله) به عنوان یک عامل استفاده شود، این موضوع اتفاق نمی افتد و امتیاز داده ی A برای k های مختلف، ثابت می ماند.

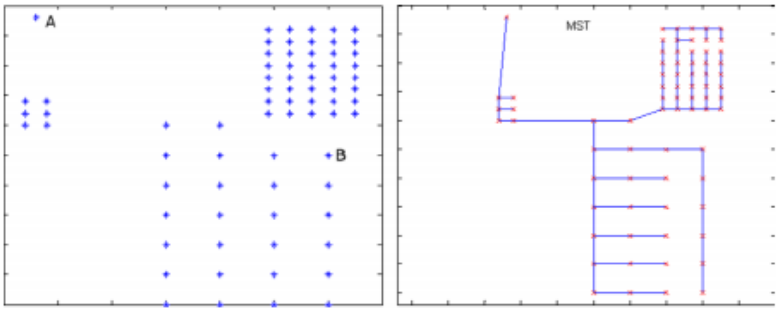
سوما، براساس دارایی برش، الگوریتم های خوشه بندیِ MST بنیان می توانند برای شناسایی خوشه های کوچکی که از طریق لبه های بلند به نقاط داده ی معمولی متصل می شوند، مفید باشند و همان طور که در شکل 4 نشان داده شده، می توانند به عنوان بخش مجزا انتخاب و لحاظ شوند. اگرچه، الگوریتم های شناسایی بخش مجزا بر مبنای

خوشه بندیِ MST استاندارد معمولا زمان اجرای درجه دویی را برای تضمین ارضا ی ویژگی های MST دارند اما از دیدگاه ما، این موضوع نه کارآمد است و نه لازم.

براساس این مشاهدات، ایده های کارکردن در پشت الگوریتم کتارآمد شناسایی بخش مجزای الگو گرفته از MST مادر زیرگروه پیش رو رسمیت پیدا می کند.



شکل 3: یک مجموعه داده ی دو بعدی که یک کمبود مجموعه داده (سمت راست) ی شناسایی بخش مجزای فاصله بنیان (سمت چپ) را نشان می دهد.



شکل 4:یک مجموعه داده ی دو بعدی که یک کمبود مجموعه داده (سمت راست) ی شناسایی بخش مجزای LOF بنیان (سمت چپ) را نشان می دهد.

3.1. دو عامل جدید بخش مجزا

تعریف 1: (بخش مجزای Hawkins) یک بخش مجزا، یک مشاهده ای است که از دیگر مشاهدات به قدری منحرف می شود که شکی مبنی بر اینکه بوسیله ی مکانیزمی متفاوت تولید شده است را به وجود می آورد.

تعریف 2: (خوشه بندی) با توجه به مجموعه داده ی T، یک الگوریتم خوشه بندی سراسری قصد می کند تا T را به k خوشه تقسیم بندی کند. $C_1, C_2,\ldots, C_K, C_i \neq \emptyset, C_i \cap C_j = \emptyset, T = C_1 \cup C_2 \cup \ldots \cup C_K, ij = 1{:}K, i \neq j,$ به طوری که فاصله ی بین هر کدام از نزدیک ترین نقاط در یک خوشه کمتر از فاصله ی بین هر کدام از نقاط در همان خوشه و هر نقطه ای که در خوشه (یا، در نزدیکترین خوشه ی آن) نیست، می باشد.

تعریف 3: (فاصله ی بین دو خوشه) اجازه دهید C_i, C_j خوشه های مجموعه داده ی T باشند، فاصله ی بین Ci و Cj به صورت $\rho(C_i,C_j) = \min \{\rho(x_i,x_j) | x_i, \in C_i, x_j \in C_j\}$ تعریف می شود.

تعریف 4: (درخت پوشای کمینه) با توجه به گراف متصل شده و وزن گذاری شده، $G(T) = (V,E)$ با یک تنظیم راس $V = T$ (برای مثال، مجموعه ای از T نقطه ی داده) و تنظیم لبه ی $E = \{e_{ij} = (x_i, x_j) | x_i, \ x_j \in T, \ i \neq j\}$. هر لبه ی eij دارای وزن $w(x_i, \ x_j)$ است، یک درخت پوشای کمینه (MST) ی گراف G(T) ، یک زیرگروه فاقد چرخه[1] از E است که همه ی رئوس در V را با وزن مجموع حداقل $W(\text{MST}) = \min \{\Sigma_{xi,xj \in V, eij \in E} w(x_i, x_j)\}$ متصل می کند و دو شرط زیر را ارضا می کند:

1) دارایی برش: یک لبه با کوچکترین وزن که دو بخش از مجموعه ی رئوس را قطع می کند باید به یک MST تعلق داشته باشد و

2) دارایی چرخه: یک لبه با بزرگترین وزن در هر چرخه ای در یک گراف نمی تواند در یک MST باشد.

اگر یک وزن (برای مثال w) که دلالت به یک فاصله ی (برای مثال p) بین دو تقطه ی انتهایی دارد، به هر لبه تخصیص داده شود، هر لبه در یک MST کوتاهترین فاصله بین دو زیردرختی است که بوسیله ی آن لبه متصل می شوند. بنابراین، حذف بلندترین لبه ها (برای مثال، لبه های ناسازگار) به صورت تئوری می تواند خوشه ها را نتیجه دهد.

[1] acyclic

تعریف5: (پیوند) بخش بندی گره های گراف G، قسمتی در دو زیرمجموعه ی (C_i, C_j) ناتهی منفصل است. یک پیوند، لبه ای در G است که وزن آن برابر با فاصله ی $\rho(C_i, C_j)$. است.

قضیه ی 1: همه ی لبه های MST پیوندهایی از چند بخش در گراف G هستند. (دلیل در 20 داده می شود)

تعریف6: (خوشه بندی بر مبنای درخت پوشای کمینه) با توجه به کجکوعه داده ی T و یک MST آن، اجازه دهید Cm و Cn دو زیر گروه T باشند به طوریکه $C_m,\ C_n \subseteq T,\ C_m \cap C_n = \emptyset \text{ and } C_m \neq \emptyset,\ C_n \neq \emptyset$. Cm و Cn خوشه هایی بر مبنای درخت پوشای کمینه هستند اگر بوسیله ی حذف پیوندی که وزن آن (برای مثال فاصله) به طور چشمگیری بزرگتر از میانگین وزن لبه های نزدیک در هر دو طرف پیوند باشد، شکل بگیرند. بدین معنا که $\rho(C_m, C_n) \gg \max\{\rho(C_m), \rho(C_n)\}$ که در آن $\rho(C_m) = \max\{\rho(x_i, x_j) | x_i, x_j \in C_m, e_{ij} \in MST_T,\ i \neq j\}$ and $\rho(C_n) = \max\{\rho(x_i, x_j) | x_i, x_j \in C_n,\ e_{ij} \in MST_T,\ i \neq j\}$.

تعریف 7: (بخش های مجزا بر مبنای خوشه بندی درخت پوشای کمینه) اجازه دهید $C_1, C_2, \ldots, C_K$ خوشه های مجموعه داده ی T باشند که بوسیله ی خوشه بندی MST بنیان در بخش شناخته شده اند، به طوریکه $|C_1| \geq |C_2| \geq \cdots \geq |C_K|$. با توجه به پارامترهای α و β، بخش های مجزا برپایه ی خوشه بندی، خوشه هایی از Ck در Ci هستند به طوریکه

that: $|C_1| + |C_2| + \cdots + |C_{i-1}| \geq |T| * \alpha$ and $|C_1| + |C_2| + \cdots + |C_{i-2}| \leq |T| * \alpha$. and $C_{i-1}/C_i > \beta$.

براساس توزیع نرمال، α می تواند مقدار 0.3٪ را بدون از دست دادن عمومیت بگیرد، β می تواند حداقل مقدار 3 را بگیرد.

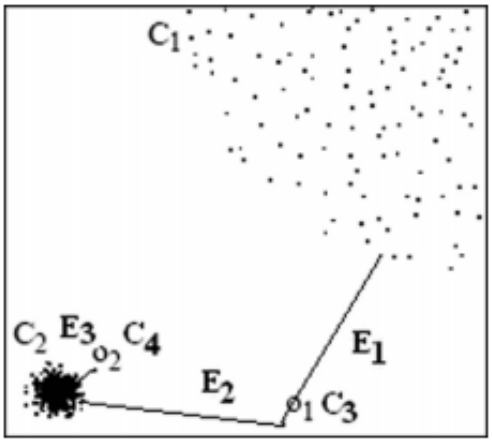
تعریف 8: (بخش های مجزای سراسری بر مبنای خوشه بندی MST) اجازه دهید $C_1, C_2, \ldots, C_K$ خوشه های مجوعه داده ی T که بوسیله ی خوشه بندی براساس MST یافته شده اند، باسند. و ρ_{MAX} بزرگترین وزن لبه ی

داخل خوشه باشد. اگر $\rho(C_i,\ C_j)\gg\rho_{MAX},$ باشد که در آن Cj نزدیک ترین خوشه ی همسایه ی Ci است، خوشه ی Ci یک خوشه ی بخش مجزای سراسری بر مبنای خوشه بندی MST است.

تعریف 9: (بخش های مجزای محلی برمبنای خوشه بندی MST) اجازه دهید $C_1,C_2,\ldots,C_K$ خوشه های مجوعه داده ی T که بوسیله ی خوشه بندی براساس MST یافته شده اند، باسند. و ρ_{MAX} بزرگترین وزن لبه ی داخل خوشه باشد. اگر $\rho(C_i,\ C_j)<\rho_{MAX},$ باشد اما $\rho(C_i,\ C_j)\gg\rho(C_j),$ باشد که در آن Cj نزدیک ترین خوشه ی همسایه ی Ci است، ، خوشه ی Ci یک خوشه ی بخش مجزای محلی بر مبنای خوشه بندی MST است.

برای مجموعه داده ی نمونه که در شکل 5 نشان داده شده است، می توان دید که o1 (برای مثال خوشه ی C3) یک بخش مجزای سراسری بر مبنای MST است چون $E_1\gg\rho_{MAX}\,(\text{i.e., }\rho(C_1))\text{ and }E_1<E_2$ در حالیکه O2 (برای مثال خوشه ی C4) بخش مجزای محلی برمبنای MST است چون $E_3<\rho_{MAX}\text{ and }E_3\gg\rho(C_2)$ که در آن $E_3=\rho(C_2,\ C_4).$

تعریف 10: (عامل بخش مجزای سراسری بر مبنای خوشه بندی MST) اجازه دهید C1 یک گروه بخش مجزای سراسری بر مبنای خوشه بندی MST باشد و C2 نزدیکترین خوشه ی C1 باشد. عامل بخش مجزای سراسری بر مبنای خوشه بندی MST به صورت $\rho(C_1,C_2)$ تعریف می شود.



شکل 5: خوشه های نمونه در مجموعه داده ی دو بعدی

تعریف 11: (عامل بخش مجزای محلی بر مبنای خوشه بندی MST) اجازه دهید C1 یک گروه بخش مجزای محلی بر مبنای خوشه بندی MST باشد و C2 نزدیکترین خوشه ی C1 باشد. عامل بخش مجزای محلی بر مبنای خوشه بندی MST به صورت $\rho(C_1,C_2)/\rho(C_2)$ تعریف می شود.

از آنجایی که تنها تعداد اندکی از بخش های مجزا در یک مجموعه داده وجود دارد، الگوریتم های شناسایی بخش مجزا بر پایه ی خوشه بندی می توانند کارآمد تر باشند اگر بلندترین لبه ها که خوشه های بخش های مجزا را به داده ی نرمال متصل می کنند، می تواند به درستی شناسایی و به سرعت موقعیت یابی شود. به عبارت دیگر، برخی از بلندترین لبه ها با هیچ کدام از بخش های خوشه تطابق ندارند اما مرتبط با بخش های مجزا هستند. اساسا، این مشاهده در مورد طراحی طرح کارآمدتر در قضیه ی پیش رو شکل دهی می شود.

قضیه ی 2: با توجه به مجموعه داده و درخت پوشای کمینه ای که از روی ان ساخته شده است، برای یک گروه دورافتاده ی نقاط داده ی k، لبه ای که این نقاط داده ی k را به باقی مجموعه داده متصل می کند، همانند آن لبه ای است که در یک MST کوچک ساخته شده از روی این نقاط داده ی k و نقطه ی داده در دیگر انتهای این لبه در درخت پوشای کمینه ای که از روی این مجموعه داده ساخته شده است، قرار دارد.

مدرک: برای اثبات این موضوع، از برهان خلف استفاده کرده ایم. فرض کنید که در آنجا لبه ی کوچکتری وجود دارد که به این نقاط داده ی دورافتاده را به باقی مجموعه داده متصل می کند. این اتفاق غیر ممکن است چون در غیر اینصورت دارایی برش مورد تخطی قرار می گیرد. بنابراین، آنها باید یکی باشند. این موضوع برای هر دو بخش مجزای محلی و سراسری صدق می کند.

براساس قضیه ی 2، موقعیت یابی بلندترین لبه ها در یک MST که بخش های مجزا را به باقی داده متصل می کند، برابر با موقعیت یابی بلندترین لبه ها در MST های کوچکی است که از روی هر نقطه داده و نزدیکترین k همسایه ی آن ساخته شده است. این موضوع زمان کمتری می برد چون بسیاری از روش های محاسبه ی kNN کارآمد ارائه شده اند. بنابراین، می توانیم یک فرآیند شناسایی بخش مجزا (برای مثال، یک فرآیند بر مبنای خوشه بندی MST) در هر نقطه ی داده و نزدیکترین k همسایه ی آن به کار ببریم تا بخش های مجزای برجسته را شناسایی کنیم. از

سوی دیگر، از آنجایی که خوشه بندی بر مبنای MST یک محدوده ی تحقیق پویا بوده است و الگوریتم های پیچیده بسیاری برای آن ایجاد شده اند، برای مجموعه های داده ی کوچک، الگوریتم های خوشه بندی برمبنای MST می توانند روی آن اقامت کنند و برای پتانسیل آنها برای شناسایی بخش مجزا در یک MST کوچک انشعاب ایجاد کنند. اینجا محلی است که تجمیع ساختار kNN و دارایی برش می تواند مورد استفاده قرار گیرد. در همین مفهوم، اگر k برای روند شناسایی بخش مجزا ، اندازه ی بزرگترین خوشه ی دورافتاده بعلاوه ی 1 تعریف شود دارای معنای بیشتری خواهد بود. در ابتدا به منظور شناسایی تناسب تعداد کم لبه های بلند که گروه های بخش مجزای کوچک را به اکثریت داده ی نرمالِ یک MST متصل می کنند، یک MST کوچک برای هر نقطه و kNN ان ساخته می شود. به منظور یافتن بخش های مجزای سراسری، در ابتدا به جستجوی لبه در این MST های کوچک بودیم. این MST ها دارای بزرگترین مقدار فاصله ی لبه هستند. اگر این مقدار به صورت چشمگیری بزرگتر از مقادیر میانگین لبه های همسایگی باشد، این بزرگترین لبه در میان MST های کوچک می توانند به عنوان امتیاز بخش خارجی سراسریِ آن واحدهای داده ای که در همان بخشی هستند که خود نقطه ی داده است، لحاظ شوند. همان طور که در شکل 4 نشان داده شده است، مشخص است که این امتیازهای سراسری در مقایسه با DB و DB-MAX کمتر به k حساس هستند.

در ادامه برای شناسایی بخش های مجزای سراسری، بیشتر به آن MST های کوچکی علاقه مند هستیم که وزن های بزرگترین لبه ها در آن به طور چشمگیری بزرگتر از مقادیر میانگینِ وزن های لبه های همسایگی است. برای کمیت سنجی بیشترِ اهمیت وزن یک لبه که بزرگتر از مقادیر میانگینِ لبه های همسایگی برای شناسایی بخش مجزای سراسری، از توزیع متحدالشکل به عنوان تخمین درجه اول برای وزن های لبه های درخت در MST های کوچک استفاده می کنیم.

تعریف 12: (شاخص بخش مجزای سراسری برپایه ی MST) اجازه دهید یک MST کوچک از روی یک MST ام برای چنین i دلالت بر وزن لبه ی dist[i] همسایه ی آن ساخته شود، kنقطه ی داده و نزدیکترین کوچک MSTکوچکی که در در ان نقطه شروع می شود دارد. شاخص بخش مجزای سراسری بر مبنای

) این وزن MeanMST) و میانگین (StdMST) تعریف می شود که تناسبی از انحراف استاندارد (SOMMST(
های لبه (برای مثال فواصل) است به طوریکه :

$$Mean_{MST} = \frac{1}{k} \sum_{i=1}^{k} dist[i] \qquad (2)$$

$$Std_{MST} = \sqrt{\frac{1}{k} \sum_{i=1}^{k} (dist[i] - Mean_{MST})^2} \qquad (3)$$

$$SOM_{MST} = \frac{Std_{MST}}{Mean_{MST}} \qquad (4)$$

SOMMST یک معیار کمیت سنج انحراف از حالت نرمال است و می تواند به عنوان یک آستانه برای غیر محتمل شمردن بخش بزرگی از داده ی نرمال مورد استعمال قرار گیرد. برمبنای این ایده ها، امتیاز بخش مجزای سراسریِ kNN بنیان در تعریف پیش رو داده شده است.

تعریف 13: (عامل بخش مجزای سراسریِ kNN بنیان و الگو گرفته از MST) اجازه دهید یک MST کوچک از روی یک نقطه ی داده و نزدیکترین K همسایه ی آن ساخته شده باشد، dist[i] دلالت بروزن لبه ی i امِ چنین MST کوچک دارد و بند-برش یک آستانه ی تامین شده برای SOMMST باشد که احتمال وجود بخش مجزا را اندازه گیری می کند. یک عامل بخش مجزای سراسری برپایه ی kNN و الگو گرفته از MST به صورت زیر تعریف می شود:

$$MST - MAX = \max_{i=1:k} \{dist[i]\} \qquad (5)$$

$$SOM_{MST} \geq cut - thred \qquad (6)$$

برای بخش های مجزای محلی، چون به طور چشمگیری تنها از نزدیکترین خوشه های همسایگی آنها بسیار دور هستند، امتیاز بخش مجزای محلی بهعنوان تناسبی از مقدار وزن بزرگترین لبه روی مقدار وزن کوچکترین لبه در هر MST کوچک تعریف می شود.

تعریف 14: (عامل بخش مجزای محلی برپایه ی kNN و الگو گرفته از MST) اجازه دهید یک MST کوچک از روی یک نقطه ی داده و نزدیکترین K همسایه ی آن ساخته شده باشد، dist[i] دلالت بروزن لبه ی i امِ چنین MST کوچک دارد، عامل بخش مجزای محلی برپایه ی kNN و الگو گرفته از MST به صورت زیر تعریف می شود:

$$MST-MAX-MIN=\frac{\max\limits_{i=1:k}\{dist[i]\}}{\min\limits_{i=1:k}\{dist[i]\}} \qquad (7)$$

شناسایی بخش مجزای محلی می تواند در زمانی که تناسب به کمتر از آستانه (مثلا 3) می رسد، پایان یافته فرض شود. در نهایت، امتیازهای بخش خروجی برای تخصیص درجه ای از بخش مجزا بودن به نقاط داده ی بازگشت داده شده استعمال می شود.

3.2. ساختار جستجوی نزدیک ترین همسایه تخمینی

از آنجایی که نقاط داده، نرمال هستند، هدفِ یافتن n بخش مجزای برتر می تواند بوسیله ی در ابتدا سریعا یافتن تخمینی خوب از امتیاز دورافتادگی برای هر واحد داده ئ در ادامه تمرکز روی موارد برتر m>=n محقق شود. با حذف همه ی پنجره ها در میان آنها، n بخش مجزای مورد نیاز پدیدار می شوند. برای محقق کردن این هدف، به طور خاصی به امکان جست وجوی نزدیکترین همسایه ی تخمینی که الگوریتم خوشه بندیِ سلسله مراتبیِ تقسیم کننده (DHCA) نامیده می شود، علاقه مند شده ایم. اساسا، برای شروع DHCA، مراکز k در سطح بالا به صورت تصادفی از کل مجموعه ی داده انتخاب می شوند. سپس هر نقطه ی داده به نزدیک ترین مرکز آن تخصیص داده می شود و k بخش ایجاد می کند. در هر سطحی که بیش از حد تکرار شده اند، برای هر کدام از این k بخش، k مرکز تصادفی به طور بازگشتی در هر بخش انتخاب می شوند و روند خوشه بندی ادامه پیدا می کند تا حداکثر kn بخش در مرحله ی n ام ایجاد کند. این فرآیند ادامه دارد تا اینکه تعداد مولفه ها در یک بخش کمتر از k+2 باشد، در این زمان، جستجوی نزدیک ترین همسایه در میان همه ی واحدهای داده در آن بخش اجرا می شود. چنین استراتژی تضمین می کند که نقاطی که در فضا به یکدیگر نزدیک تر هستند، احتمال بیشتری دارند تا در بخشی یکسان جمع

شوند و چندین اجرا از DHCA چنین احتمالاتی را به شدت تقویت می کند. ارائه ای با جزییات بیشتر و اثبات کارآمدیِ DHCA در جستجوی نزدیکترین همسایه ی تخمینی در 21 داده شده است و در اینجا تکرار نخواهد شد. بعد از چندین تکرار، kNN های دقیق و امتیازهای متناظر برای بخش های مجزای برتر محاسبه می شوند. تعداد بخش های مجزای برتر کوچک است و از این رو زمان محاسبات سریع است. با در نظر داشتن این مشاهدات، یک روش شناسایی بخش مجزای ساده در ادامه ایجاد می شود.

3.3. الگوریتم شناسایی بخش مجزا ی الهام گرفته از MST ما

ما سه عامل بالا را برای ایجاد الگوریتم شناسایی بخش مجزا برپایه ی kNN و الگو گرفته از MST مان تجمیع کردیم:

1. k را به عنوان اندازه ی بزرگترین خوشه ی دورافتاده بعلاوه ی یک تنظیم کنید

2. مجموعه داده را به طور پی در پی بخوانید و k همسایه ی هر واخد داده را از پیشینیان فوری آن یا جانشینانش روی پرش، مقدار دهی اولیه کنید (به مقداردهی پی در پی (SI) لستناد می شود)؛

3. DHCA را چندین بار راه اندازی کنید، و ، برای هر تکرار، یک آرایه ی یک بعدی از فاصله ی میانگینِ هر واحد داده تا kNN آن محاسبه کنید و در ادامه میانگین آرایه را بدست آورید؛ این مرحله را در زمانی که درصد دیفرانسیلِ میانگین بین دو تکرار متوالی کمتر از 6-10 باشد، متوقف کنید.

4. MST کوچک را از روی هر واحد داده و نزدیکترین k همسایه ی آن بسازید

5. سه آرایه ی یک بعدی از امتیازات دورافتادگی تخمین زده شده برای iNN (که در آن i=2:k) محاسبه کنید و آنها را در ترتیبی که افزایش نمی یابد، دسته بندی کنید.

6. برای واحدهای داده با امتیازات دورافتادگی سراسری بالا، iNN واقعی آنها را بیابید و امتیازات دورافتادگی واقعی آنها را محاسبه کنید، SOMMST آن را کنترل کنید، آن را بازگردانید اگر SOMMST بزرگتر از آستانه (مثلا 1.5) باشد، آنگاه اگر i=k باشد، شناسایی بخش مجزای سراسری پایان می یابد و در غیر اینصورت i=i+1 و به 5 برو.

7. برای واحدهای داده با امتیازات دورافتادگی محلی بالا، hNN واقعی آنها را پیدا کنید و امتیازات دورافتادگی آنها را محاسبه کنید، آن را بازگردانید اگر آن بزرگتر از آستانه (مثلا 3) باشد، آنگاه اگر i=k باشد، شناسایی بخش مجزای محلی پایان می یابد و در غیر اینصورت i=i+1 و به 5 برو.

8. مرحله ی 5 و 6 و 7 را تا زمانی که بخش های مجزا (برای مثال، آنهایی که امتیازشان بالاتر از آستانه های سراسری یا محلی باشد) در نظر گرفته می شوند، ادامه دهید.

از آنجایی که انتظار می رود تا تعداد بخش های مجزا نسبتا کم باشد، تعداد محاسبات مصافت استفاده شده نیز انتظار می رود تا کم باشند.

از نظر فیزیکی، منبع استفاده شده از طرف الگوریتم ما شامل فضای نگهداری همه ی مجموعه داده در حافظه، فضای ذخیره سازی k همسایه ی آنها، و برخی فضاهای موقت برای MTS های کوچک می شود. نیازهای پارامترهای عددی الگوریتم ما از کاربر شامل تعداد نزدیک ترین همسایه ها (برای مثال k)، آستانه های بخش های مجزای محلی و سراسری می شود در حالیکه خروجی ها شامل مجموعه ای از بخش های مجزای رتبه بندی شده از مجموعه داده ها می شوند. الگوریتم شناسایی بخش مجزای ما برای بهبود قابلیت خواندن در یک شکل شبه کد، در جدول 1 ارائه می شود.

3.4. تحلیل پیچیدگی زمان

از تعریف داده شده در زیربخش های قبلی می توان دید که الگوریتم ما عمدتا شانل سه مرحله می شود، یک مقدار دهی اولیه ی مکرر، یک به روز رسانی DHCA و در نهایت کاویدن بخش های مجزای برتر. پیچیدگی زمان برای مقداردهی اولیه ی مکرر dNk است. از آنجایی که تعداد بخش های مجزا کم است، مرحله ی سوم تقریبا به طور میانگین زمان نسبتا خطی ای می برد. ما برای راه اندازی های DHCA از یک ساختار درخت استفاده کردیم تا پیچیدگی زمان آن را محاسبه کنیم.

در فضای d بعدی، پیچیدگی زمان محاسبه ی kNN بالاتر است که بوسیله ی $O(dN^2)$ مرزبندی شده است. اما اگر خوشبختانه، مانند موردی که در سمت چپ شکل 6 نشان داده شده، یک مجموعه داده بتواند به طور مساوی به چندین خوشه ی به خوبی بخش بندی شده، تقسیم بندی شود، هزینه ی محاسبات می تواند تا $dN^2/4$ کاهش یابد. در ادامه این مسئله که چگونه kNN را برای نقطه ی داده سریعا پیدا کنیم تبدیل به این موضوع می شود که داده چگونه سریعا به اندازه ی کاهش یافته ی خوشه های مساوی توزیع شده ی به خوبی بخش بندی شده، تقسیم بندی شود. این موضوع که DHCA روشی بسیار خوب برای تقسیم بندی سریع داده فراهم می کند، در این مفهوم است.

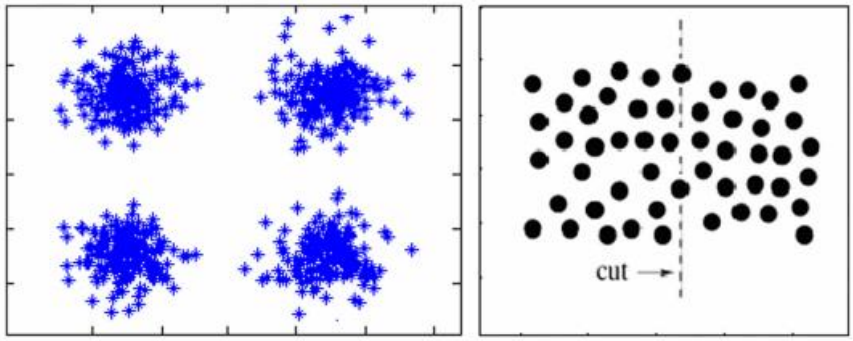
در سطح بالای DHCA، تعداد محاسبات فاصله dNk است که در آن k تعداد زیرمجموعه ها برای DHCA است. برای هر سطح متعاقب l، اگر یک درخت متعادل شده باشد، kl زیر مجموعه به اندازه ی N/K^l وجود دارد و هزینه ی تقسیم بندی هنوز dNk است. از آنجایی که k+2 عدد از نقاط داده بسیار کوچک است، محاسبه ی فاصله ی دو به دو در کمترین دسته، تقریبا زمان ثابتی می برد. برای یک درخت متعادل شده، ارتفاع درخت l برابر با $\log_K N$ است. بهترین مورد برای پیچیدگی زمان DHCA برابر با $O(dNK \log_K N + N)$ است.

Table 1
Our MST-inspired outlier detection algorithm. الگوریتم شناسایی بخش مجزای الهام گرفته از ام اس تی ما

Input:
data — a set of *N* data points — مجموعه ای از ان نقطه ی داده
k — the number of *NN*s of a data item — تعداد ان ان های هر واحد داده
K — the number of clusters at each step — تعداد خوشه ها در هر مرحله
SOM-TH — global outlier detection termination threshold — آستانه ی حذف شناسایی بخش مجزای سراسری
MAX-MIN-TH — local outlier detection termination threshold — آستانه ی حذف شناسایی بخش مجزای محلی
Output:
a set of ranked global outliers *GO*, a set of ranked local outliers *LO* — LO مجموعه ای از بخش های مجزای سراسری رتبه بندی شده ی جی او، مجموعه ای از بخش های مجزای محلی رتبه بندی شده ی
Begin شروع
set k to be the largest outlying cluster size plus 1 — کی را اندازه ی بزرگترین خوشه ی دور افتاده بعلاوه ی یک قرار دهید
perform a sequential initialization (SI) — مقداردهی پی در پی را اجرا کنید
run DHCA multiple times until the percentage difference between two consecutively updated *kNN* is below 10^{-6}; — دی اچ سی ای را چندین بار راه اندازی کنید تا اختلاف درصد بین دو کی ان ان به روز رسانی شده ی متوالی کمتر از ۱۰ است
find miniMST for each data point — ام اس تی کوچک را برای هر نقطه ی داده پیدا کن
for each $i=2{:}k$
{ compute three 1-dimensional arrays of the estimated outlying scores for *iNN*, namely, *MST-MAX*, *MST-MAX-MIN* and SOM_{MST}, and sort the first two in a non-increasing order and the orders are remembered in *MST-MAX-INDEX* and *MST-MAX-MIN-INDEX*, respectively;
GO_T=[];
for $j=1{:}N$
{
find true *iNN* for *MST-MAX-INDEX*[*j*];
recomputed the global score and SOM_{MST};
if (SOM_{MST}[*MST-MAX-INDEX*[*j*]] > *SOM-TH*)
GO_T=[*GO_T MST-MAX-INDEX*[*j*]];
end
}
GO=[*GO GO_T*];
LO_T=[];
for $j=1{:}N$
{
find *iNN* for *MST-MAX-MIN-INDEX*[*j*];
recomputed the local score; — امتیاز محلی مجدد محاسبه شده
if(*MST-MAX-MIN*[*j*]] > *MAX-MIN-TH*)
LO_T=[*LO_T MST-MAX-MIN-INDEX*[*j*]];
end
}
LO=[*LO LO_T*];
}
End

سه آرایه ی تک بعدیِ امتیازات دور افتادگی تخمین زده شده برای آی ان ان را محاسبه کن و دو تای اول را در ترتیبی که افزایش نمی یابد دسته بندی کنید و این ترتیب در ـ به خاطر سپرده می شود

GO_T and *LO_T* are temporary arrays to hold ranked outliers for each *i*.
آرایه های موقت هستند تا بخش های مجزای رتبه بندی شده را برای هر آی نگه دارد



شکل 6: یک تصویر از تاثیر DHCA

اما (در رویه) ممکن است درخت متعادل نداشته باشیم و تقسیم بندی می تواند منجر به دو زیرگروه کاملا نامتعادل شود که یک مجموعه در آنها شامل اکثر نقاط می شود. بنابراین، بدترین پیچیدگی زمان برای DHCA می تواند $O(dN^2)$ باشد. خوشبختانه، از طریق گزینش تصادفیِ مراکز دسته بندی، احتمال جدا کردن یک نقطه از همسایه

هایش کم است و بدترین مورد در زمانی که همه ی واحدهای داده در یک قسمت خوشه بندی می شوند، کم ارزش خواهد شد. پیچیدگی زمان میانگین برای DHCA به سختی می تواند به عنوان مورد درخت متعادل تقریب زده شود. از سوی دیگر، همان طور که در سمت راست شکل 6 نشان داده شده، هر نقطه ی داده در یک قسمت به مرکز خوشه اش نزدیک تر (نه نزدیک ترین همسایه ی آن) از مرکز هر قسمت دیگر است. نقاط داده در مرزهای خوشه ها می تواند به اشتباه در بخشی اشتباه دسته بندی شوند. خوشبختانه، MST ها در برابر این اثر تقریبا حساس نیستند که می تواند به سادگی از شکل 6 بدست آید. بنابراین، تعداد کمی DHCA نیازهای ما را محقق می کند.

Table 2
The node class. کلاس گره

Name اسم	Explanation توضیح
public data members: اعضای داده ی عمومی	
sampleNumbers; شماره ی نمونه ها	An array holding the indices of all samples in the cluster یک آرایه که شاخص های همه ی نمونه در خوشه را نگه می دارد
centroid; مرکز ثقل	An array holding the indices of the randomly chosen cluster centers یک آرایه که شاخص های مراکز خوشه ی تصادفی انتخاب شده را نگه می دارد
childNodes; گره فرزند	An array holding the indices of its child Nodes in the Node array یک آرایه که شاخص های گره فرزند را در آرایه ی گره نگه می دارد
parent; والد	An integer holding the index of its parent Node in the Node array یک عدد صحیح که شاخص های گره والدش را در آرایه ی گره نگه می دارد
vecindex;	An integer holding the index of the current Node in the Node array یک عدد صحیح که شاخص های گره کنونی را در آرایه ی گره نگه می دارد
public Methods: روش های عمومی	
void DHCA تهی	Our DHCA Procedure فرایند ما

Table 3
The DHCA member function. تابع عضویت

Procedure name نام فرایند	DHCA
Input:	آرایه های کمکی برای نزدیک ترین کی همسایه ی هر واحد داده
dist_knn, edge_knn	The auxiliary arrays to remember *k*-nearest neighbors(*kNN*) for each data item
k	The number of *NN*s of a data item تعداد ان ان های یک واحد داده
nodeArray	An array of the *Node* structures یک آرایه از ساختارهای گره
currentNode	The current *Node* in the *Node* array گره کنونی در آرایه ی گره
K	The number of clusters at each step تعداد خوشه ها در هر مرحله
data	The input data set مجموعه داده ی ورودی
clustersize	The maximum size of each clusters بیشترین اندازه ی هر خوشه
Output:	
updated *dist_knn, edge_knn*, and newly generated $\leq K$ Nodes	

```
Begin
    randomly select K centers from sampleNumbers of currentNode;   مراکز کی را به صورت تصادفی از شماره های نمونه ی گره کنونی انتخاب کن
    generate K newNodes;   کی گره جدید تولید کن
    for each sample i in sampleNumbers of currentNode that is not a center   برای هر نمونه ی آی در شماره های نمونه ی گره کنونی که مرکز نیست
    {   find its nearest center j out of K;   نزدیک ترین مرکز جی در خارج از کی را پیدا کن
        if (dist_knn[i].max > distance(i,j))
            update dist_knn, edge_knn;
        end
        assign sampleNumbers[i] to the group of center j;   شماره های نمونه را به مرکز گروه جی تخصیص بده
    }
    for each newNode j=1: K
    {   if (newNode[j].sampleNumbers.size > clustersize)
            push newNode[j] to the end of nodeArray;
            assign values to data members parent and vecindex;
        end                                   مقادیر را به والد یا وسیتدکس اعضای داده تخصیص دهید
    }
End
```

dist_knn[*i*].*max* is the *kNN*th nearest neighbor of data item *i*.

به عنوان یک نتیجه، انتظار داریم که پیچیدگی زمانِ الگوریتم ما $O(dfNK \log_K N)$ باشد که در آن f دلالت بر تعداد راه اندازی های DHCA دارد.

3.5. شبه کد برای DHCA

پیاده سازی یک DHCA در رویکرد ما از طریق طراحی یک ساختار داده ی C++ است که گره نامیده می شود و چندین متغیر عضو برای دفترداری و تابع عضویت اصلی دارد. لازم به ذکر است که این تابع عضویت، مجموعه ی خودش را به k زیرخوشه، خوشه بندی می کند. خروجی های ساختار داده ی گره در بیشتر k ها، گرهی جدید به عنوان زاده ای از گره کنونی است.

روند خوشه بندی سلسله مراتبی تقسیم کننده با یک نمونه ی گره (به نام گره برتر) شروع می کند که همه ی داده های داده را در مجموعه ی داده به عنوان نمونه هایش دارد و k زیرمجموعه ی داده به شکل k گره تولید می کند. تنها در زمانی که تعداد نمونه ها در گره بزرگتر از اندازه ی خوشه ی از پیش تعریف شده شود، آن گره به پشت گره برتر هل داده می شود. این عمل باعث شکل گیری آرایه ای از گره ها می شود. این روند به صورت بازگشتی ادامه پیدا می کند تا اینکه گره جدید تولید نمی شود و به انتهای آرایه ی گره موجود می رسد. کلاس گره در جدول 2 خلاصه می شود و فرآیند DHCA در جدول 3 داده می شود.

4. یک مطالعه عملکرد

در این بخش، نتایج سه مجموعه از آزمایشات انجام شده در جهت ارزیابی الگوریتم شناسایی بخش مجزای الگو گرفته از MST را ارائه می کنیم. در آزمایش 1، سه مجموعه داده ی ترکیبیِ دو بعدی استفاده می شود تا نشان دهد که روش شناسایی بخش مجزای الگو گرفته از MST ما می تواند از الگوریتم های شناسایی بخش مجزای کلاسیک در دقت دسته بندی بهتر عمل می کند. در آزمایش 2، پنج مجموعه داده ی واقعی که از مخزن یادگیری ماشین

UCI بدست آمده اند، استعمال می شوند تا استقامت فنیِ این مطالعه را کنترل کند و کارآمدی روش ما را در وضعیت دنیای واقعی نشان دهد.

Table 4
مجموعه های داده the sets of data.

Data set مجموعه داده	اندازه Size (N)	ابعاد Dimensionality
مجموعه داده ی یک Dataset1	515	2
Dataset2	78	2
Dataset3	473	2
شاتل Shuttle	14 500	9
لیمفوگرافی Lymphography	148	18
یونوسفر Ionosphere	351	34
ارقام نوری Optical digits	5620	64
ویژگی های چندگانه Multiple features	2000	649
IPUMS	88 443	68
نوع پوشش Covertype	581 012	55
آمار ایالات متحده UScensus	2 458 285	61

در آزمایش 3، عملکرد زمان اجرای الگوریتم ارائه شده را در سه مجموعه داده ی بزرگ واقعی ارزیابی می کنیم و آن را با الگوریتم شناسایی بخش مجزای MST+LOF مقایسه می کنیم تا اثر اندازه های داده ی مختلف و ورودی k در الگوریتم ما را نشان دهیم. همه ی الگوریتم ها در c/c++ پیاده سازی شده اند و روی کامپیوتری با پردازنده ی Intel Core 2 Duo E6550 2.33 GHz CPU و 2 GB RAM راه اندازی شده اند. سیستم عاملی که روی این کامپیوتر کار می کند ویندوز XP است. ما از ابزار های تایمر که در کتابخانه ی استاندارد C تعریف شده اند استفاده می کنیم تا زمان CPU را گزارش دهیم. نتایج نشان می دهد که به طور کلی، الگوریتم شناسایی بخش مجزای الگو گرفته از MST نسبت به دیگر الگوریتم های شناسایی بخش مجزای مدرن در دو بخش دقت دسته بندی و زمان اجرا برتر است. برای ثبات، تنها از پارامتر k برای ارائه ی اندازه ی همسایگی در بررسی این روش ها استفاده می کنیم.

4.1. عملکرد الگوریتم ما در مجموعه داده های ترکیبی

در این زیر بخش، از سه مجموعه داده ی ترکیبی استفاده می کنیم تا نشان دهیم که روش پیشنهاد شده می تواند به طور کارآمدی بخش های مجزای محلی و سراسری را در سناریوهای مختلف شناسایی کند. در هر مجموعه داده، چندین خوشه و شش بخش مجزا وجود دارد (A,B,C,D,E,F) که در همسایگی خوشه ها قرار داده می شوند. یک ویژگی چالش برانگیز خاص درباره ی این مجموعه داده ها این است که خوشه ها در اندازه های مختلف هستند و تراکم های متفاوتی دارند. به طور خاصی، از مجموعه داده های دو بعدی استفاده کرده ایم. این کار به علت ثبات آنها برای جستجوی تصویری است.

در ابتدا برای مطالعه ی تاثیرات نسبیِ الگوریتم شناسایی بخش مجزای ارائه شده ی ما روی پارامتر k، برای مقایسه با DB, DB-MAX, LOF, COF, INFLO, LDOF, RBDA, and MST+LOF، بزرگترین تعداد نقاط داده یک خوشه ی دورافتاده را مشخص می کنیم، در مرحله ی بعد، امتیازات بخش مجزای داده شده از طریق تعریف هر بخش مجزا محاسبه می شود و آنها را در ترتیب هایی که کاهش پیدا نمی کنند، مرتب سازی کرده و موارد بالای آن را برگشت می دهیم.

مجموعه داده ی ترکیبی 1 شامل 515 نمونه می شود. این نمونه ها شامل بخش مجزای قرار داده شده، یک خوشه ی به صورت نرمال توزیع شده و دو خوشه ی متحدالشکل کوچک می شوند. این موضوع یک کار شناسایی بخش مجزای سراسری است. برای رویکرد ما، اگر اندازه ی بزرگترین گروه دور افتاده روی 2 قرار داده شده باشد، k برای همه ی الگوریتم ها در این مورد 3 است. نتایج تجربی در شکل 7 نشان داده شده اند که شش بخش مجزای برتر به ترتیب با بعلاوه ی قرمز، ستاره ی قرمز، ضربدر قرمز، مثلث قرمز، دایره ی قرمز و مربع قرمز علامت گذاری شده اند.

از اشکال می توان دید که DB, DB-MAX, LOF, و INFLO MST+LOF و روش ما (برای مثال MST-MAX) به درستی همه ی بخش های مجزا را شناسایی می کنند، اما رتبه بندی برای روش ها اندکی متفاوت است. متاسفانه، COF به خوبی کار نمی کند و A,D,F را از دست می دهد. از آنجایی که میانگین نزدیکترین k فاصله و میانگین برای k(k-1) فاصله ی داخلی دو به دو ، هر دو برای B,D,F بزرگ هستند، نسبتشان نزدیک به یک است و

بوسیله ی LDOF از دست داده می شوند. RBDA یک بخش مجزای سراسری را در شکل 7 از دست می دهد (برای مثال A). این پدیده ناشی از این حقیقت است که رتبه ی A در نزدیکترین سه همسایه ی خود نسبتا کمتر از رتبه ی H در نزدیکترین سه همسایه ی H است. PBOD سه حق برای شش بخش مجزای برتر شناسایی می کند.

تا کنون، به طور ضمنی فرض شد که تعداد بخش های مجزا از قبل داده شده است و در زمانی که روند شناسایی باید به یک انتها نزدیک شود، مشخص نیست. به عنوان یک بهبود، SOMMST می تواند به عنوان یک تخمین درجه ی اول برای چنین هدفی استعمال شود.

برای آزمایش مکانیزم حذف مان، هفت بخش مجزای برتر که با استفاده از الگوریتم ما و برای k=3 در میان طرح روی آخرین خط شکل 7شناسایی شدند را نشان می دهیم. از شکل می توانیم محدودیت مولفه ی بخش مجزای سراسری را مشاهده می کنیم. بدین معنا که نقطه ی داده ی k به اشتباه از طرف تعریف بخش مجزای سراسریمان شناسایی شد. خوشبختانه، مکانیزم حذف ما برای شناسایی بخش مجزای سراسری، شیء داده ی k مشخص می شود که مقدار SOMMST بسیار کمی (که در حقیقت صفر است) دارد و پایان شناسایی بخش مجزای سراسری را اعلام می کند.

علاوه بر شش بخش مجزای قرار داده شده، سه خوشه نیز وجود دارد. به همین تناسب، اگر اندازه ی بزرگترین گروه دورافتاده روی 9 تنظیم شده باشد، مجموعه داده تنها یک الگوی داده ی غالب دارد. نتایج تجربی برای هر روش در زمانی که k روی 10 تنظیم شده، در شکل 8 نشان داده شده که در این مورد، مجموعا 24 بخش مجزا وجود دارد. برای ساده سازی تصویر، مجموعا 24 بخش مجزای برتر وجود دارد که با بعلاوه ی قرمز علامت گذاری می شود.

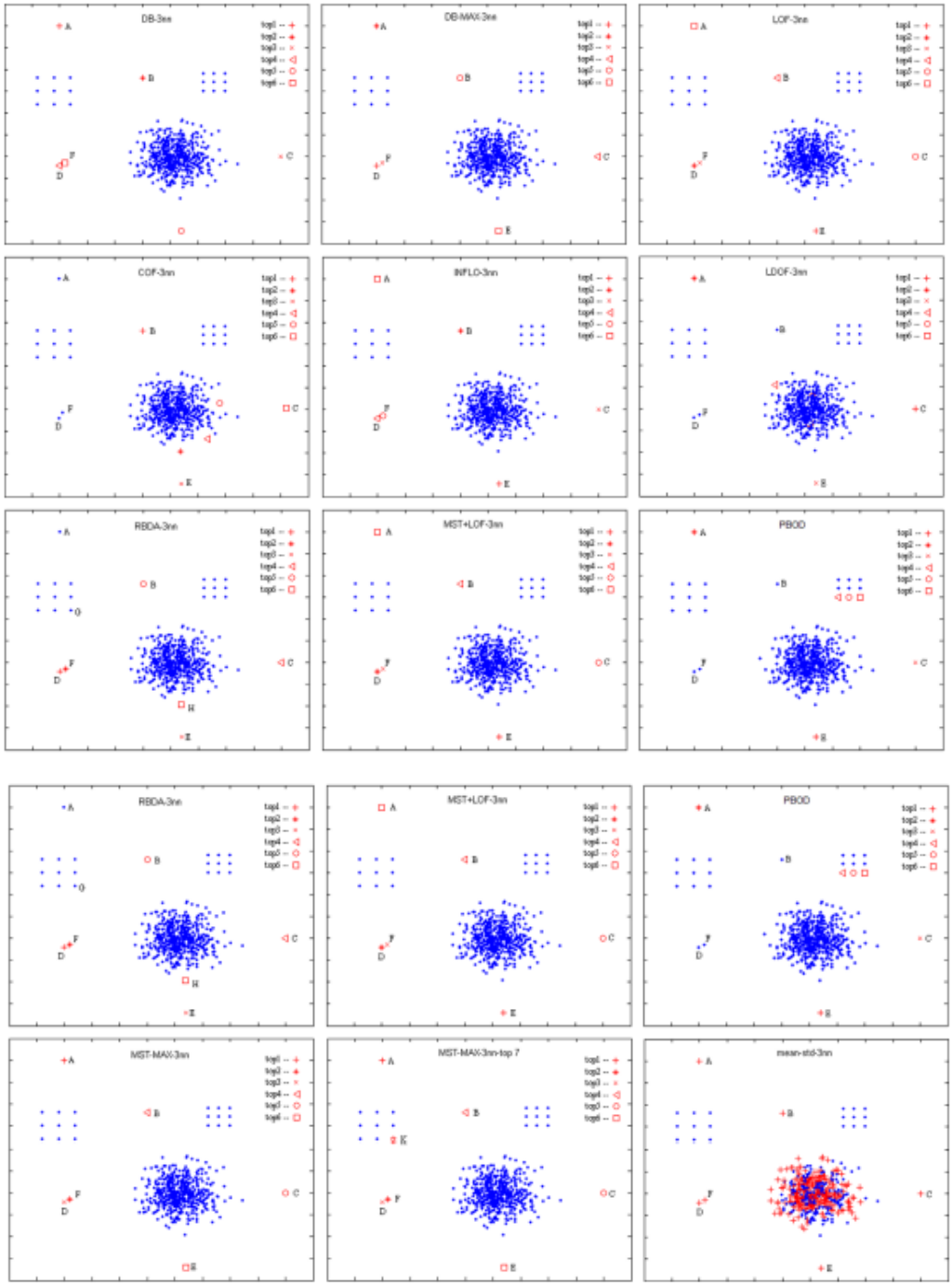
می توان از شکل دید که DB برای k=10 یکی را از دست می دهد. LOF تنها ده تا را از بیست و چهارمورد به درستی شناسایی می کند. COF نیز چندان خوب کار نمی کند و 19 مورد از 24 مورد را به اشتباه دسته بندی می کند. INFLO تنها 9 مورد از 24 موردرا می گیرد. LDOF دو خوشه با اندازه ی 9 را شناسایی نمی کند و از این رو 18 مورد از 24 مورد را از دست می دهد. RBDA در یکی از دو خوشه ی اندازه ی 9، 9 نقطه را از دست می دهد. PBOD در این مورد اندکی بهتر کار می کند و تنها 2 تا از 24 بخش مجزای برتر را ازدست می دهد. LOCI دو تا را

از دست می دهد اما با برخی اشتباهات قطعی. DB-MAX, MST+LOF و روش ما (برای مثال MST-MAX) همه ی بخش های خروجی را به درستی تشخیص می دهند.

برای آزمایش در این موضوع که آیا رویکرد ما می تواند به طور موثر بخش های مجزای معنادار را در یک مجموعه داده ی اندکی پیچیده تر بیابد یا نه، از مجموعه داده ی ترکیبی 2 استفاده می کنیم که شامل 78 نمونه می شود. این نمونه ها شامل پنج بخش مجزای سراسری قرار داده شده، یک بخش مجزای محلی قرار داده شده (B) و چهار خوشه با غلظت های مختلف از جمله 36 و 8 و 12 و 16 نمونه ی متحدالشکل توزیع شده می شوند.

برای نشان دادن کارآمدی رویکرد ما در یافتن بخش های مجزای محلی و سراسری، کارآمدی روش های فاصله بنیان و فاصله بنیان روی این مجموعه داده را مقایسه می کنیم. برای این روش هایی که بر پایه ی kNN هستند در ابتدا k=3 را تنظیم کرده ایم و تصاویر در شکل 9 نتایج کاویدن شش بخش مجزای برتر را نشان می دهند. برای این موضوع، DB و DB-MAX، هر دو B و C را از دست می دهند و برای A,D,E,F رتبه ای یکسان دارند. LDOF و MST-MAX و MST+LOF همگی یک مورد را از دست می دهنددر حالیکه PBOD 5 مورد را از دست می دهد. RBDA, INFLO, COF, LOF و مولفه ی بخش مجزای محلی ما MST-MAX-MIN همه ی شش بخش مجزا را به درستی شناسایی می کنند. تصویر در گوشه ی پایین سمت راست، نقاط داده ای را نشان می دهد که مقدار SOMMST آن بالای 1.5 باشد که به درستی شش بخش مجزا را شناسایی می کند.

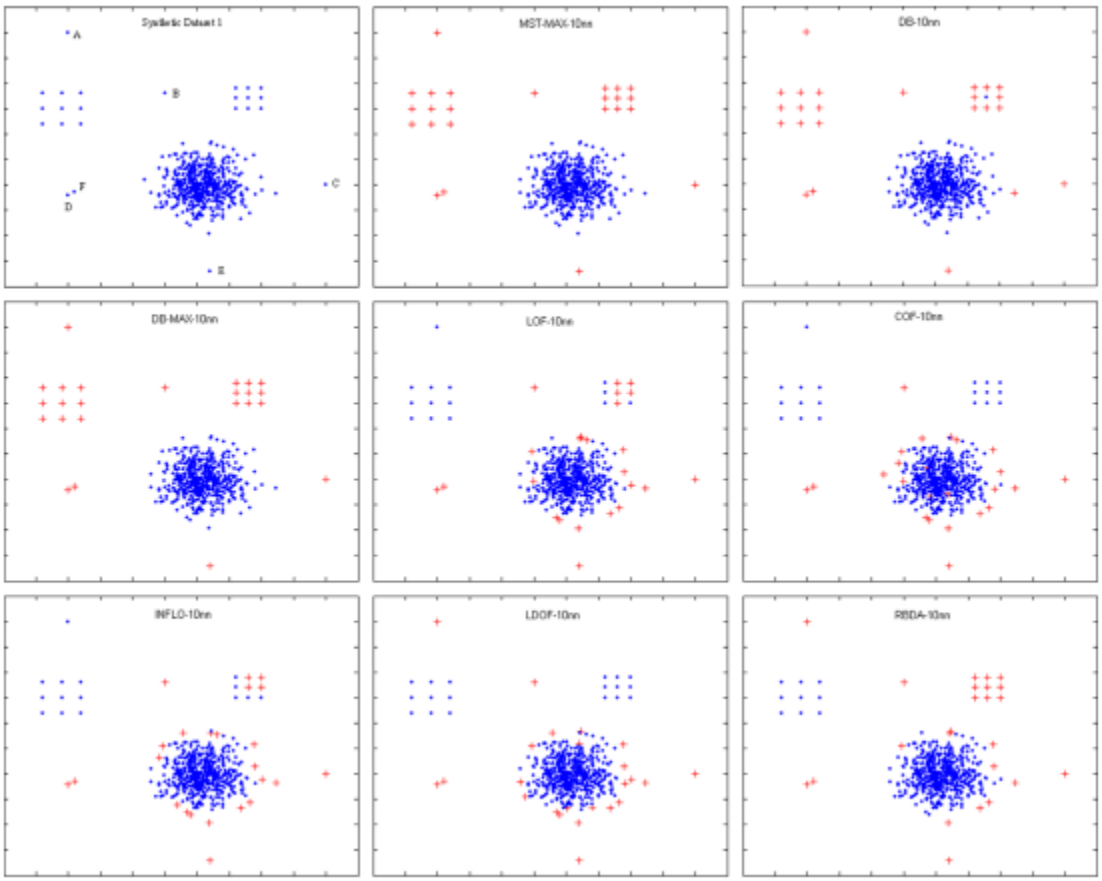
علاوه بر شش بخش مجزای قرار داده شده، یک خوشه ی کوک که شامل 8 نقطه می شود نیز وجود دارد. اگر این خوشه نیز به عنوان یک گروه دورافتاده لحاظ شود، k را روی 9 تنظیم می کنیم. برای این مورد،14 بخش مجزا وجود دارد. نتایج شناسایی در شکل 10 نشان داده می شوند. از شکل می توانیم مشاهده کنیم DB-MAX در زمانی که بخش مجزای محلی وجود داشته باشد، به خوبی کار نمی کند. DB 10 مورد، DB-MAX 11 مورد، LOF 3 مورد، COF 6 مورد، INFLO 4 مورد، LDOF 8 مورد، RDBA 6 مورد، PBOD 12 مورد را از دست می دهند. LOCI 11 مورد، MST+LOF 1 مورد را از دست می دهند و شناساگر بخش مجزای محلی ما (MST-MAX-MIN) همه ی بخش های مجزا را به درستی تشخیص می دهد.

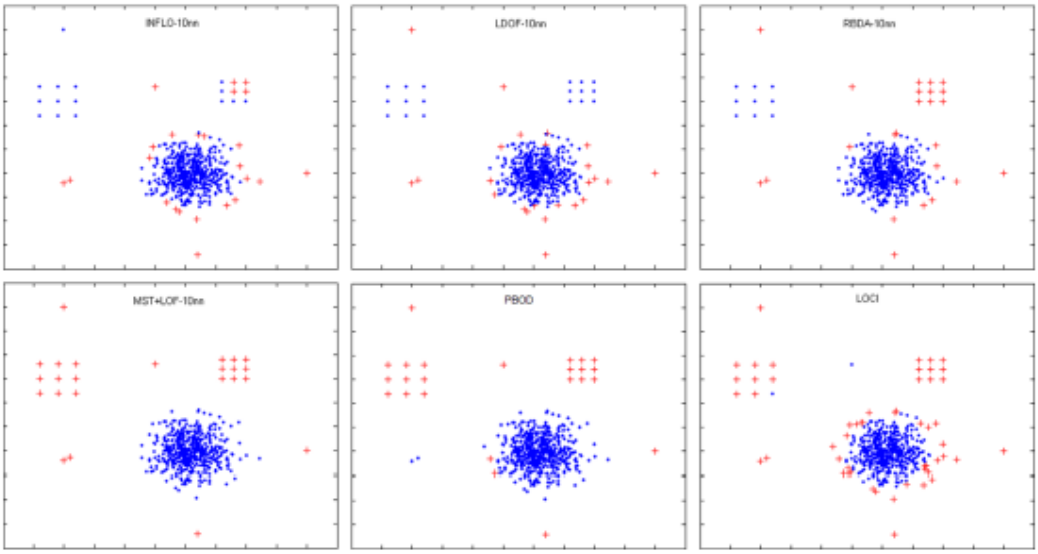


شکل 7: نتایج شناسایی بخش مجزا برای مجموعه داده ی ترکیبی 1 برای k=3

با توجه به شکل های 9 و 10، مزیت مولفه ی شناسایی بخش مجزای محلی ما (MST-MAX-MIN) در این مجموعه داده با ابعاد نسبتا کم که نه تنها بخش های مجزای سراسری بلکه بخش های مجزای محای نیز دارد، مشهود است. آزمایش نشان می دهد که روش ما برای نوع ترکیبی داده و نیز این مفهوم مطلوب است.

برای آزمودن روش ما به عنوان یک تمامیت، مجموعه داده ی 3 که دارای 473 نقطه ی داده می باشد و شامل پنج خوشه با تراکم های مختلف و شش بخش مجزا است، مورد استعمال قرار می گیرد.یک ویژگی چالش برانگیز خاص برای این مجموعه داده این است که سه خوشه ی متراکم تر در یک خوشه ی پراکنده در گوشه ی بالا، سمت راست قرار داده می شوند. از آنجایی که بزرگترین گروه دورافتاده دارای دو نقطه ی داده است (برای مثال E و F)، k در ابتدا روی سه تنظیم می شود. عملکرد روش شناسایی بخش مجزا در شکل 11 نشان داده می شود. قرار دادی مشابه برای رسیم ها نیز به کار برده می شود به سادگی از شکل دیده می شود که این موضوع نیز یک وضعیت شناسایی بخش مجزای سراسری با تفاوتی نسبت مجموعه داده ی 1 است.این گونه است که روند شناسایی بوسیله ی اتصال فوری خوشه ها با تراکم های مختلف توزیع می شود.

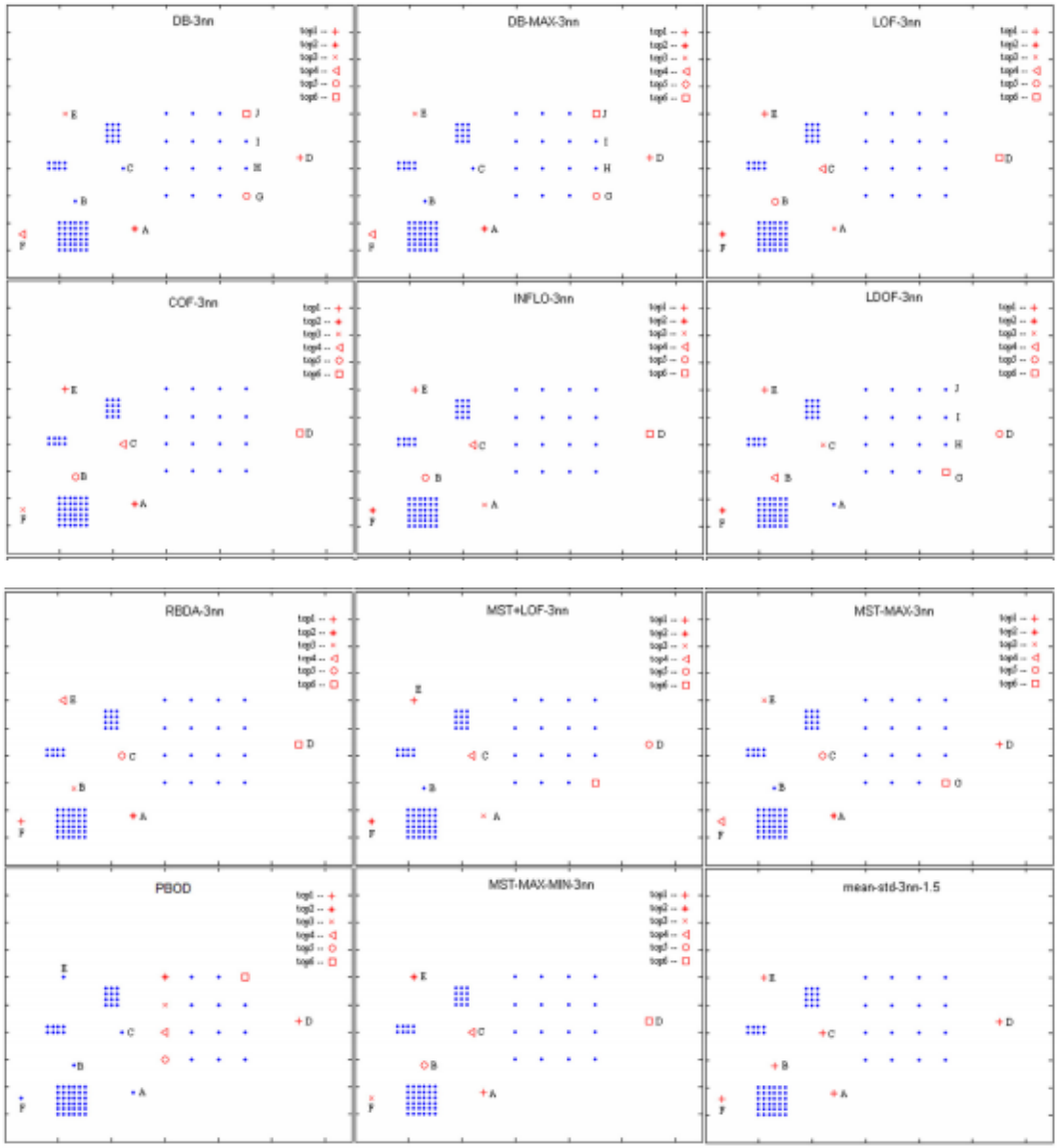




شکل 8: نتایج شناسایی بخش مجزا برای مجموعه داده ی ترکیبی 1 برای k=10

برای شناسایی شش بخش مجزای برتر، COF ، B و C را از دست می دهد، LDOF ، E و F را از دست می دهد، RBDA ، C را و PBOD همه را از دست می دهد. خبر خوب این است که DB, DB-MAX, LOF, INFLO, MST+LOF و روش ما هر شش بخش مجزا را به درستی اما با رتبه بندی های مختلف شناسایی می کنند. اگر خوشه ی پراکنده (خوشه ای که دارای هفت نقطه ی داده است) نیز به عنوان یک گروه دورافتاده لحاظ شود، k روی 8 تنظیم می شود. همان طور که در شکل 12 نشان داده شده، برای این وضعیت، DB-MAX و LOF هر دو یک مورد را از دست می دهند، COF، 7 مورد، INFLO ، دو مورد، LDOF 8 مورد، RBDA، 6 مورد، PBOD، 6 مورد، LOCI، 8 مورد را از دست می دهند. در حالیکه DB، MST+LOF و شناساگر بخش مجزای ما همه را به درستی تشخیص می دهند.

به طور خلاصه، می توان از اشکال 7 تا 12 دید که روش ما هیچ مشکلی در شناسایی همه ی بخش های مجزا ندارد و به وضوح بهترین رتبه بندی را در سه مجموعه داده ی ترکیبی ارائه می کند در حالیکه همه ی روش های دیگر، در شناسایی همه ی بخش های مجزا برای دو k مختلف به طور شایسته ای عمل نمی کنند. به عبارت دیگر، مزیت مولفه های شناسایی بخش مجزای ما روی این مجموعه داده های دو بعدی بسیار مشهود است.



شکل 9: نتایج شناسایی بخش مجزا برای مجموعه داده ی ترکیبی 2 برای k=3

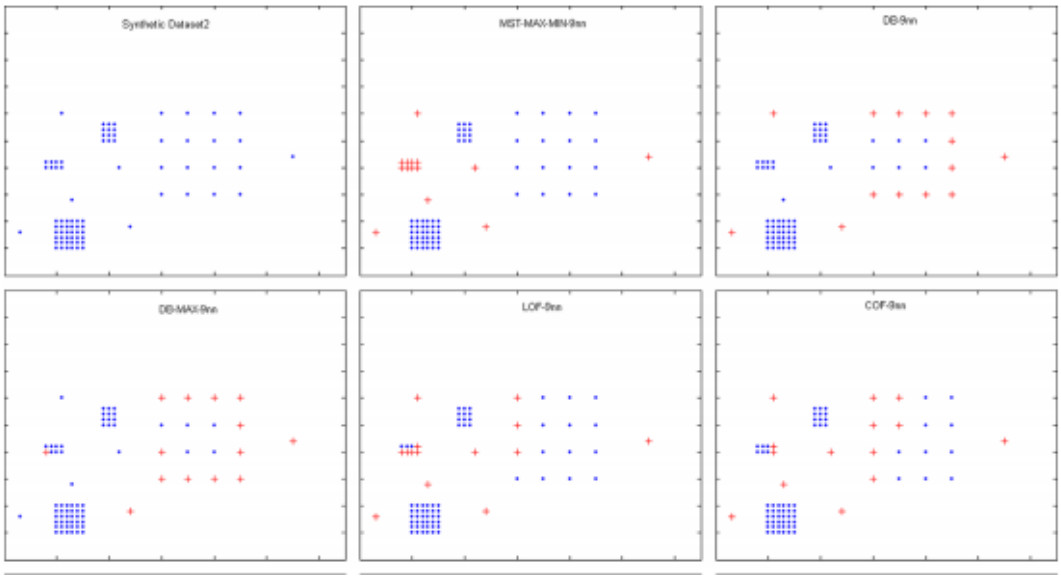
4.2. عملکرد روی مجمو عه های داده ی واقعی

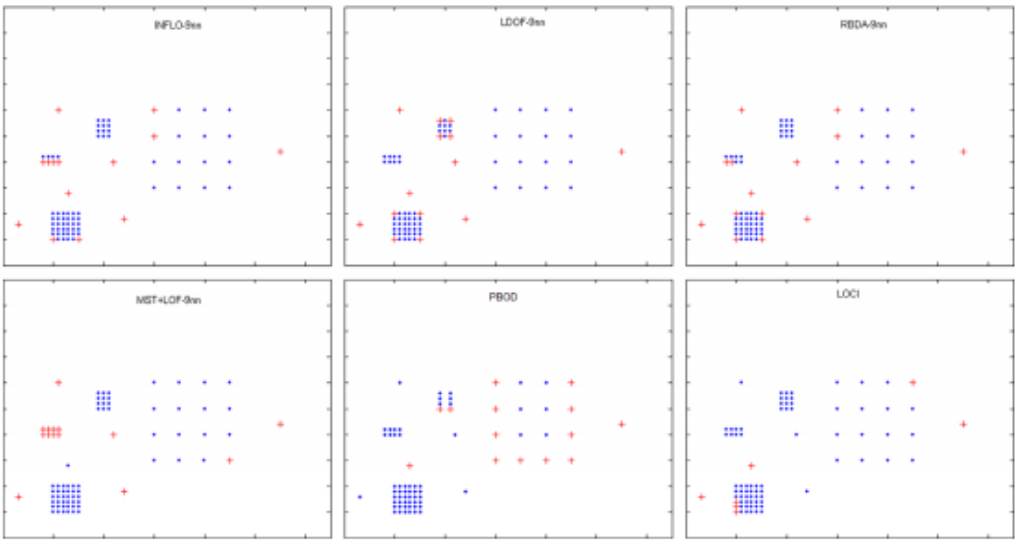
همان طور که بوسیله ی Aggarwal and Yu نشان داده شد، یک راه برای آزمودن این مسئله که الگوریتم شناسایی بخش مجزا چقدر خوب کار می کند، راه اندازی روش در مجموعه داده و آزمودن درصد نقاطی که متعلق به کلاس های نادر هستند، می باشد. برای ارزیابی کارآمدی و دقت روش پیشنهادی ما روی داده ی واقعی، الگوریتم

ها را بوسیله ی عملکردشان در شناسایی کلاس های نادر در پنج مجموعه داده ی واقعی از جمله مجموعه داده های لیمفوگرافی، شاتل، یونوسفر، ارقام نوری و چند ویژگی مقایسه کرده ایم که از UCI دانلود می شوند.

برای اندازه گیری کمیِ عملکرد یک طرح شناسایی بخش مجزا، سه سنجش محبوب به نام های دقت، یادآوری و توان رتبه در اینجا به کار برده می شوند. با فرض اینکه یک مجموعه داده ی $D=Do\cup Dn$ که در ان D0 دلالت بر مجموعه ی همه ی بخش های مجزا دارد و Dn دلالت بر مجموعه ی همه ی داده های نرمال دارد. با توجه به هر عدد صحیح m>=1، اگر Om دلالت بر مجموعه ی بخش های مجزا در میان اشیاء در m موقعیت برتر بازگشت داده شده بوسیله ی یک طرح شناسایی بخش مجزا داشته باشد، دقت، یادآوری به صورت زیر تعریف می شوند:

$$precision=\frac{|O_m|}{m} \qquad (8)$$





شکل 10: نتایج شناسایی بخش مجزا برای مجموعه داده ی ترکیبی 2 برای k=9

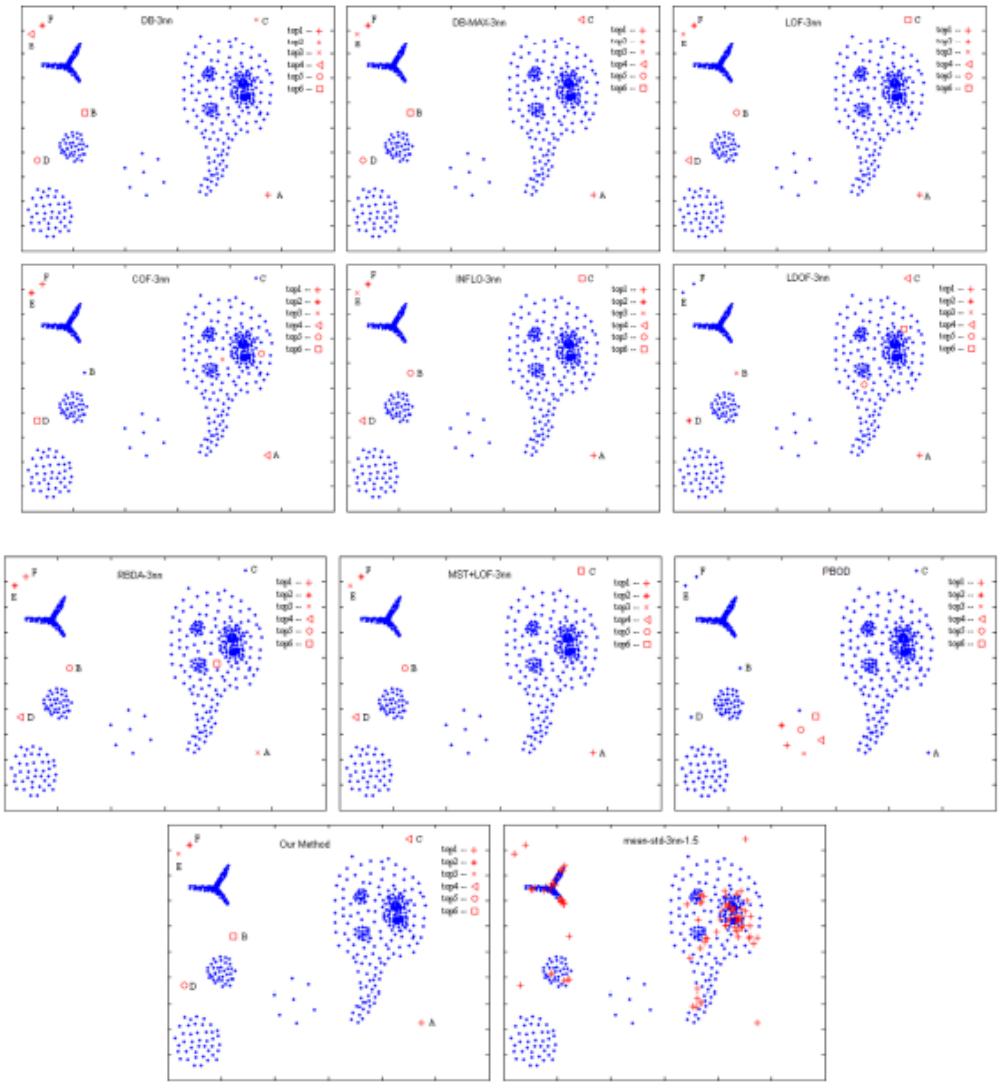
$$recall = \frac{|O_m|}{|D_o|} \qquad (9)$$

بنابراین، دقت، درصد بخش های مجزا را در میان m شیءرتبه بندی شده ی برتر که بوسیله ی یک روش برگشت داده شده اند را اندازه گیری می کند، در حالیکه یادآوری، درصد مجموعه ی بخش مجزای کل را که در m شیء رتبه بندی شده ی برتر مشمول می شود را اندازه گیری می کند. معمولا، کاربران تنها علاقه مند به اینکه چه تعداد بخش مجزای صحیح بوسیله ی یک روش برگشت داده شده اند، نیستند بلکه همچنین به محل قرار گرفتن انها نیز علاقه مندند. توان رتبه یک سنجش است که جایگذاری و تعداد نتلیج برگشت داده شده بوسیله ی یک روش را لحاظ می کند. در اینجا، توان رتبه ی داده شده در 35 مورد استعمال قرار گرفته است. فرض کنید که یک روش m شیء را برگشت دهند که n تا از آنها بخش مجزای صحیح هستند. برای $1 \le i \le n$ اگر Li دلالت بر موقعیت بخش مجزای i ام باشد، توان رتبه ی روش با توجه به m می تواند به صورت زیر تعریف شود

$$Rank\ Power = \frac{n(n+1)}{2\sum_{i=1}^{n} L_i} \qquad (10)$$

همان طور که در معادله ی 10 می توان دید، توان رتبه، جایگذاری بخش های مجزای برگشت داده شده را به سنگینی وزن گذاری می کند. یک بخش مجزایی که پیش از این در لیستِ برگشت داده شده قرار داده شود، نسبت

به آنهایی که بعدا در لیست قرار داده شوند،کمتر به مخرج توان رتبه اضافه می کند (و از این رو در سنجش توان رتبه بیشتر مشارکت می کند). مقدار 1 بهترین عملکرد و صفر بدترین عملکرد را نشان می دهد.



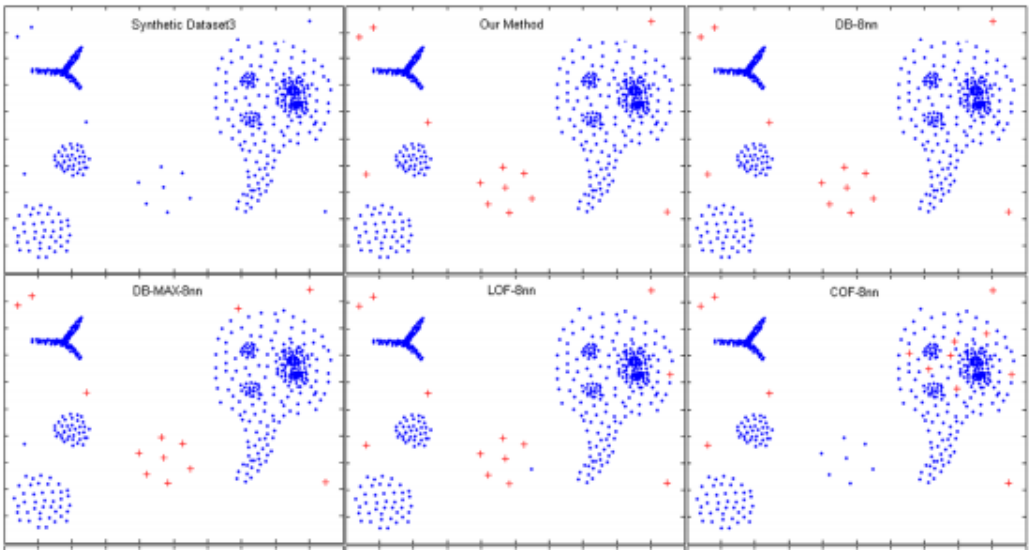
شکل 11: نتایج شناسایی بخش مجزا برای مجموعه داده ی ترکیبی 3 با استفاده از k=3

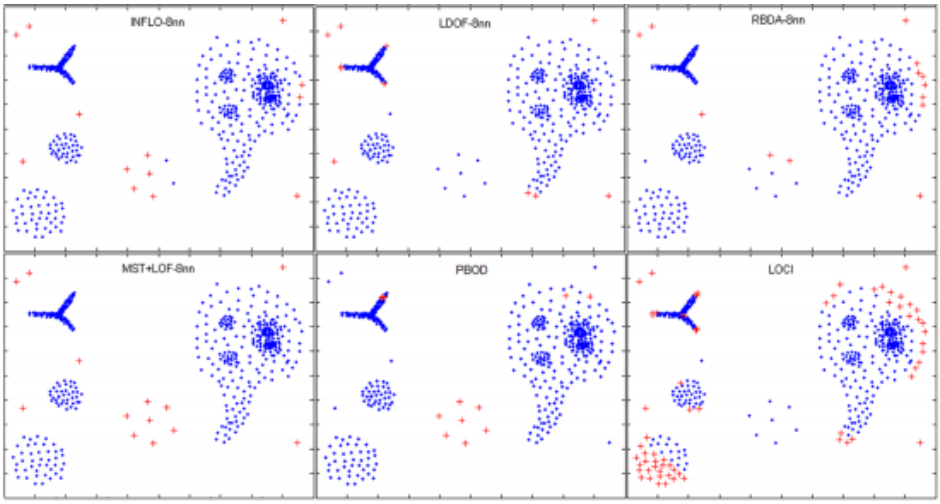
در ادامه برای اندازه گیری توانایی هر الگوریتم به منظور بازیابی محتمل ترین بخش های مجزا و برای مقایسه ی کیفیت رتبه بندی های فراهم شده از طریق هر الگوریتم، عملکرد با سه سنجش یادآوری، دقت و توان رتبه اندازه گیری می شود. این سه سنجش به ترتیب با p و r و rp نشان گذاری شدند. مقدار m ، نشانگر m سابقه ی برگشت داده شده ی برتر بوسیله ی طرح ما است. برای همه ی آزمایش ها در این زیربخش که روش ما را استفاده می کنند، SOM-TH روی 1.5 و MAX-MIN-TH روی 3 تنظیم شده است.

4.2.1. داده لیمفوگرافی

مجموعه داده ی لیمفوگرافی دارای 148 نمونه با 18 ویژگی است و در مجموع شامل 4 کلاس می شود. کلاس های 2 و 3 دارای بیشترین نمونه ها هستند (به ترتیب 81 و 61). دو کلاس باقی مانده در مجموع دارای 6 نمونه هستند (به ترتیب 2 و 4) و به عنوان بخش های مجزا لحاظ می شوند (کلاس های نادر) چون از نظر اندازه کوچک هستند. ما نتایج شناسایی متناظرِ روشمان، روش PBOD و سه مورد از بهترین موارد از باقی روش ها را در جدول 5 برای 4 مقدار k و 5 مقدار m، گزارش می دهیم.

برای زمانی که k برابر با 7 باشد، PBOD اولین روشی می باشد که همه ی شش بخش مجزا را در 10 نمونه ی رتبه گذاری شده ی برتر می کاود. LOF و روش ما دومین ها هستند. برای k برابر با 30، LOF و INFLO و RBDA بهتر از PDOB عمل می کنند. عملکرد های شناسایی روش ما مشابه با چهار روش دیگر است و برای همه ی k ها یکسان باقی می ماند، بدین معنا که روش ما نسبت به پارامتر k کمتر حساس است که با انتظارات ما تطابق دارد.





شکل 12: نتایج شناسایی بخش مجزا برای مجموعه داده ی ترکیبی 3 برای k=8

4.2.2. داده شاتل فضایی

مجموعه داده ی شاتل شامل 14500 شیء می شود که هر شیء در ان دارای 9 ویژگی با ارزش گذاری واقعی و یک برچسب عدد صحیح است (1-7) و 7 خوشه دارد. ما این اشیاء را با برچسب های 2 و 6 و 7 (به ترتیب با اشیاء 13 و 4 و 2) به عنوان بخش مجزا و 4 کلاس باقی مانده را به عنوان داده ی نرمال لحاظ می کنیم (کلاس های 1 و 3 و 4 و 5 با به ترتیب نمونه های 11478 و 39 و 2155 و 809). در جدول 6 نتایج تجربیِ روشمان، روش PBOD و سه مورد از بهترین روش های باقی مانده را برای چهار مقدار مختلف k و هفت مقدار مختلف m نشان دادیم. همان طور که در جدول نشان داده شده، این مسئله کاری ساده برای همه ی تکنیک ها ی دیگر نیست. روش ما به طور چشمگیری در مقایسه با دیگر روش ها برای همه ی k ها و همه ی m ها بهتر عمل می کند. به طور خاص، روش ما همانند مورد داده ی لیمفوگرافی، دارای بالاترین دقت و یادگیری است و حساسیت کمتری نسبت به k دارد که مجددا شهودی را بیان می کند با این عنوان که استفاده از وزن های لبه ی درخت در MST کوچک یا MST به عنوان شاخص درجه ی دورافتادگی نسبت به مقادیر مختلف k حساسیت کمتری و در نتیجه قابلیت اطمینان بیشتری دارد.

Table 5
Lymphography, $\kappa=7,10,20,30$.

m	LOF				INFLO				RBDA				Our method				PBOD			
	n	p	r	rp	n	p	r	rp	n	p	r	rp	n	p	r	rp	n	p	r	rp
k=7																				
5	4	0.80	0.67	0.83	3	0.60	0.50	0.60	3	0.60	0.50	0.86	4	0.80	0.67	0.71	2	0.40	0.33	**1.00**
10	5	0.50	0.83	0.75	5	0.50	0.83	0.58	4	0.40	0.67	0.71	5	0.50	0.83	0.68	**6**	**0.60**	**1.00**	0.88
15	**6**	**0.40**	**1.00**	0.64	5	0.33	0.83	0.58	5	0.33	0.83	0.60	**6**	**0.40**	**1.00**	0.57	**6**	**0.40**	**1.00**	0.88
20	**6**	**0.30**	**1.00**	0.64	5	0.25	0.83	0.58	5	0.25	0.83	0.60	**6**	**0.30**	**1.00**	0.45	**6**	**0.30**	**1.00**	0.88
30	**6**	**0.20**	**1.00**	0.64	**6**	**0.20**	**1.00**	0.40	**6**	**0.20**	**1.00**	0.42	**6**	**0.20**	**1.00**	0.37	**6**	**0.20**	**1.00**	0.88
k=10																				
5	4	0.80	0.67	0.83	3	0.60	0.50	0.60	3	0.60	0.50	1.00	4	0.80	0.67	0.71	–	–	–	–
10	5	0.50	0.83	0.79	5	0.50	0.83	0.65	5	0.50	0.83	0.65	5	0.50	0.83	0.68	–	–	–	–
15	5	0.33	0.83	0.79	5	0.33	0.83	0.65	5	0.33	0.83	0.65	**6**	**0.40**	**1.00**	0.49	–	–	–	–
20	**6**	**0.30**	**1.00**	0.58	5	0.25	0.83	0.65	5	0.25	0.83	0.65	**6**	**0.30**	**1.00**	0.49	–	–	–	–
30	**6**	**0.20**	**1.00**	0.58	5	0.17	0.83	0.65	5	0.17	0.83	0.65	**6**	**0.20**	**1.00**	0.40	–	–	–	–
k=20																	–	–	–	–
5	4	0.80	0.67	**1.00**	4	0.80	0.67	**1.00**	4	0.80	0.67	**1.00**	4	0.80	0.67	0.77	–	–	–	–
10	5	0.50	0.83	0.94	5	0.50	0.83	0.94	5	0.50	0.83	0.94	5	0.50	0.83	0.60	–	–	–	–
15	5	0.33	0.83	0.94	**6**	**0.40**	**1.00**	0.68	5	0.33	0.83	0.94	**6**	**0.40**	**1.00**	0.40	–	–	–	–
20	**6**	**0.30**	**1.00**	0.66	**6**	**0.30**	**1.00**	0.68	**6**	**0.30**	**1.00**	0.58	**6**	**0.30**	**1.00**	0.40	–	–	–	–
30	**6**	**0.20**	**1.00**	0.66	**6**	**0.20**	**1.00**	0.68	**6**	**0.20**	**1.00**	0.58	**6**	**0.20**	**1.00**	0.32	–	–	–	–
k=30																	–	–	–	–
5	4	0.80	0.67	**1.00**	4	0.80	0.67	0.83	4	0.80	0.67	**1.00**	4	0.80	0.67	0.77	–	–	–	–
10	**6**	**0.60**	**1.00**	0.84	**6**	**0.60**	**1.00**	0.78	**6**	**0.60**	**1.00**	0.84	5	0.50	0.83	0.71	–	–	–	–
15	**6**	**0.40**	**1.00**	0.84	**6**	**0.40**	**1.00**	0.78	**6**	**0.40**	**1.00**	0.84	**6**	**0.40**	**1.00**	0.44	–	–	–	–
20	**6**	**0.30**	**1.00**	0.84	**6**	**0.30**	**1.00**	0.78	**6**	**0.30**	**1.00**	0.84	**6**	**0.30**	**1.00**	0.38	–	–	–	–
30	**6**	**0.20**	**1.00**	0.84	**6**	**0.20**	**1.00**	0.78	**6**	**0.20**	**1.00**	0.84	**6**	**0.20**	**1.00**	0.31	–	–	–	–

Note: maximum values are marked bold.

Table 6
Shuttle, $\kappa=7,20,40,60$.

m	DB				LOF				RBDA				Our method				PBOD			
	n	p	r	rp	n	p	r	rp	n	p	r	rp	n	p	r	rp	n	p	r	rp
k=7																				
20	6	0.30	0.32	0.23	1	0.05	0.21	0.06	1	0.05	0.05	0.20	9	0.45	0.47	0.35	6	0.30	0.32	0.34
30	6	0.20	0.32	0.23	2	0.07	0.10	0.07	2	0.07	0.11	0.09	12	0.40	0.63	0.36	6	0.20	0.32	0.34
60	8	0.13	0.42	0.18	10	0.17	0.53	0.14	8	0.13	0.42	0.13	16	0.27	0.84	0.33	7	0.12	0.37	0.27
90	9	0.10	0.47	0.16	11	0.12	0.58	0.14	10	0.11	0.53	0.13	17	0.19	0.89	0.31	7	0.08	0.37	0.27
120	10	0.08	0.53	0.14	11	0.09	0.58	0.14	12	0.10	0.63	0.13	18	0.15	0.95	0.25	7	0.06	0.37	0.27
140	11	0.08	0.58	0.13	11	0.08	0.58	0.14	15	0.11	0.79	0.12	18	0.1	0.95	0.25	7	0.05	0.37	0.27
200	16	0.08	0.84	0.10	13	0.06	0.68	0.11	16	0.08	0.84	0.12	**19**	**0.09**	**1.00**	0.12	7	0.04	0.37	0.27
k=20																				
20	6	0.30	0.32	0.23	4	0.20	0.21	0.32	1	0.05	0.05	0.08	11	0.55	0.58	0.40	–	–	–	–
30	6	0.20	0.32	0.23	5	0.17	0.26	0.28	4	0.13	0.21	0.10	14	0.47	0.74	0.42	–	–	–	–
60	7	0.12	0.37	0.19	9	0.15	0.47	0.20	7	0.12	0.37	0.13	18	0.30	0.95	0.37	–	–	–	–
90	9	0.10	0.47	0.15	13	0.14	0.68	0.17	9	0.10	0.47	0.12	**19**	**0.21**	**1.00**	0.33	–	–	–	–
120	10	0.08	0.53	0.13	15	0.12	0.79	0.16	13	0.11	0.68	0.11	**19**	**0.16**	**1.00**	0.27	–	–	–	–
140	11	0.08	0.58	0.12	17	0.12	0.89	0.15	15	0.11	0.79	0.11	**19**	**0.14**	**1.00**	0.27	–	–	–	–
200	**19**	**0.10**	**1.00**	0.10	**19**	**0.09**	**1.00**	0.15	**19**	**0.10**	**1.00**	0.11	**19**	**0.09**	**1.00**	0.12	–	–	–	–
k=40																				
20	6	0.30	0.32	0.23	4	0.20	0.21	0.29	0				11	0.55	0.58	0.40	–	–	–	–
30	6	0.20	0.32	0.24	5	0.17	0.26	0.25	3	0.10	0.16	0.07	14	0.47	0.74	0.42	–	–	–	–
60	6	0.10	0.32	0.24	8	0.13	0.42	0.18	7	0.12	0.37	0.12	18	0.30	0.95	0.37	–	–	–	–
90	9	0.10	0.47	0.15	9	0.10	0.47	0.16	8	0.09	0.42	0.11	**19**	**0.21**	**1.00**	0.33	–	–	–	–
120	9	0.08	0.47	0.15	11	0.09	0.58	0.14	13	0.11	0.68	0.11	**19**	**0.16**	**1.00**	0.27	–	–	–	–
140	11	0.08	0.58	0.12	16	0.11	0.84	0.12	15	0.11	0.79	0.11	**19**	**0.14**	**1.00**	0.27	–	–	–	–
200	18	0.09	0.95	0.10	**19**	**0.09**	**1.00**	0.12	**19**	**0.10**	**1.00**	0.10	**19**	**0.09**	**1.00**	0.12	–	–	–	–
k=60																				
20	6	0.30	0.32	0.23	4	0.20	0.21	0.23	0				11	0.55	0.58	0.40	–	–	–	–
30	6	0.20	0.32	0.24	5	0.17	0.26	0.22	3	0.10	0.16	0.07	14	0.47	0.74	0.42	–	–	–	–
60	6	0.10	0.32	0.24	8	0.13	0.42	0.17	7	0.12	0.37	0.12	18	0.30	0.95	0.37	–	–	–	–
90	8	0.09	0.42	0.17	8	0.09	0.42	0.17	7	0.08	0.37	0.12	**19**	**0.21**	**1.00**	0.33	–	–	–	–
120	9	0.08	0.47	0.15	11	0.09	0.58	0.12	13	0.11	0.68	0.11	**19**	**0.16**	**1.00**	0.27	–	–	–	–
140	10	0.07	0.53	0.13	16	0.11	0.84	0.11	14	0.10	0.74	0.11	**19**	**0.14**	**1.00**	0.27	–	–	–	–
200	16	0.08	0.84	0.09	**19**	**0.09**	**1.00**	0.11	15	0.08	0.79	0.11	**19**	**0.09**	**1.00**	0.12	–	–	–	–

Table 7
Ionosphere, κ=5,10,20,30.

m	DB				RBDA				MST+LOF				Our method				PBOD			
	n	p	r	rp	n	p	r	rp	n	p	r	rp	n	p	r	rp	n	p	r	rp
k=5																				
5	**5**	**1.00**	**0.04**	**1.00**	**5**	**1.00**	**0.04**	**1.00**	4	0.80	0.03	**1.00**	**5**	**1.00**	**0.04**	**1.00**	4	0.80	0.03	**1.00**
10	**10**	**1.00**	**0.08**	**1.00**	**10**	**1.00**	**0.08**	**1.00**	9	0.90	0.07	**1.00**	**10**	**1.00**	**0.08**	**1.00**	7	0.70	0.06	0.76
30	**30**	**1.00**	**0.24**	**1.00**	**30**	**1.00**	**0.24**	**1.00**	29	0.97	0.23	**1.00**	**30**	**1.00**	**0.24**	**1.00**	20	0.67	0.16	0.73
60	**60**	**1.00**	**0.48**	**1.00**	**60**	**1.00**	**0.48**	**1.00**	59	0.98	0.47	**1.00**	59	0.98	0.47	**1.00**	32	0.53	0.25	0.72
90	88	0.98	0.70	**1.00**	88	0.98	0.70	0.99	89	0.99	0.71	0.99	83	0.92	0.66	**1.00**	34	0.38	0.27	0.69
120	107	0.90	0.85	0.98	98	0.82	0.78	0.98	106	0.88	0.84	0.94	94	0.78	0.75	0.97	58	0.48	0.46	0.51
130	109	0.84	0.87	0.98	100	0.77	0.79	0.97	109	0.84	0.86	0.92	96	0.74	0.76	0.96	68	0.52	0.54	0.51
140	110	0.79	0.87	0.97	103	0.74	0.82	0.96	113	0.81	0.90	0.91	98	0.70	0.78	0.95	70	0.50	0.56	0.51
k=10																				
5	**5**	**1.00**	**0.04**	**1.00**	**5**	**1.00**	**0.04**	**1.00**	4	0.80	0.03	**1.00**	**5**	**1.00**	**0.04**	**1.00**	–	–	–	–
10	**10**	**1.00**	**0.08**	**1.00**	**10**	**1.00**	**0.08**	**1.00**	9	0.90	0.07	**1.00**	**10**	**1.00**	**0.08**	**1.00**	–	–	–	–
30	**30**	**1.00**	**0.24**	**1.00**	**30**	**1.00**	**0.24**	**1.00**	29	0.97	0.23	**1.00**	**30**	**1.00**	**0.24**	**1.00**	–	–	–	–
60	**60**	**1.00**	**0.48**	**1.00**	**60**	**1.00**	**0.48**	**1.00**	59	0.98	0.47	**1.00**	**60**	**1.00**	**0.48**	**1.00**	–	–	–	–
90	88	0.98	0.70	**1.00**	88	0.98	0.70	0.99	89	0.99	0.71	0.99	89	0.99	0.71	**1.00**	–	–	–	–
120	107	0.90	0.85	0.98	97	0.81	0.77	0.98	106	0.88	0.84	0.95	96	0.80	0.76	0.99	–	–	–	–
130	109	0.84	0.87	0.98	100	0.77	0.79	0.97	111	0.85	0.88	0.93	99	0.76	0.79	0.97	–	–	–	–
140	112	0.80	0.89	0.96	101	0.72	0.80	0.96	114	0.81	0.90	0.93	100	0.71	0.79	0.97	–	–	–	–
k=20																				
5	**5**	**1.00**	**0.04**	**1.00**	**5**	**1.00**	**0.04**	**1.00**	4	0.80	0.03	**1.00**	**5**	**1.00**	**0.04**	**1.00**	–	–	–	–
10	**10**	**1.00**	**0.08**	**1.00**	**10**	**1.00**	**0.08**	**1.00**	9	0.90	0.07	**1.00**	**10**	**1.00**	**0.08**	**1.00**	–	–	–	–
30	**30**	**1.00**	**0.24**	**1.00**	**30**	**1.00**	**0.24**	**1.00**	29	0.97	0.23	**1.00**	**30**	**1.00**	**0.24**	**1.00**	–	–	–	–
60	**60**	**1.00**	**0.48**	**1.00**	**60**	**1.00**	**0.48**	**1.00**	59	0.98	0.47	**1.00**	**60**	**1.00**	**0.48**	**1.00**	–	–	–	–
90	85	0.94	0.67	**1.00**	84	0.93	0.67	0.99	89	0.99	0.71	**1.00**	**90**	**1.00**	**0.71**	**1.00**	–	–	–	–
120	103	0.86	0.82	0.97	98	0.82	0.78	0.96	106	0.88	0.84	0.94	103	0.86	0.82	**1.00**	–	–	–	–
130	107	0.82	0.85	0.96	102	0.78	0.81	0.95	110	0.85	0.87	0.90	105	0.81	0.83	0.99	–	–	–	–
140	111	0.79	0.88	0.95	105	0.75	0.83	0.93	111	0.79	0.88	0.89	106	0.76	0.84	0.98	–	–	–	–
k=30																				
5	**5**	**1.00**	**0.04**	**1.00**	**5**	**1.00**	**0.04**	**1.00**	4	0.80	0.03	**1.00**	**5**	**1.00**	**0.04**	**1.00**	–	–	–	–
10	**10**	**1.00**	**0.08**	**1.00**	**10**	**1.00**	**0.08**	**1.00**	9	0.90	0.07	**1.00**	**10**	**1.00**	**0.08**	**1.00**	–	–	–	–
30	**30**	**1.00**	**0.24**	**1.00**	**30**	**1.00**	**0.24**	**1.00**	29	0.97	0.23	**1.00**	**30**	**1.00**	**0.24**	**1.00**	–	–	–	–
60	**60**	**1.00**	**0.48**	**1.00**	**60**	**1.00**	**0.48**	**1.00**	59	0.98	0.47	**1.00**	**60**	**1.00**	**0.48**	**1.00**	–	–	–	–
90	83	0.92	0.66	0.99	86	0.96	0.68	0.99	89	0.99	0.71	**1.00**	**90**	**1.00**	**0.71**	**1.00**	–	–	–	–
120	98	0.82	0.78	0.96	98	0.82	0.78	0.97	106	0.88	0.84	0.93	107	0.89	0.85	**1.00**	–	–	–	–
130	103	0.79	0.82	0.94	102	0.78	0.81	0.95	111	0.85	0.88	0.90	108	0.83	0.86	0.99	–	–	–	–
140	106	0.76	0.84	0.93	105	0.75	0.83	0.94	112	0.80	0.89	0.89	109	0.78	0.86	0.98	–	–	–	–

Table 8
Optical digits, κ=7.

m	n	p	r	rp	n	p	r	rp	n	p	r	rp	n	p	r	rp	n	p	r	rp
	DB				DB-MAX				RBDA				Our method				MST+LOF			
Class '2'																				
5	0	0.00	0.00	0.00	2	0.40	0.40	0.50	0	0.00	0.00	0.00	1	0.20	0.20	0.33	0	0.00	0.00	0.00
10	1	0.10	0.20	0.13	3	0.30	0.60	0.40	1	0.10	0.20	0.20	3	0.30	0.60	0.43	0	0.00	0.00	0.00
15	2	0.13	0.40	0.16	4	0.27	0.80	0.40	1	0.07	0.20	0.20	4	0.27	0.80	0.42	0	0.00	0.00	0.00
20	3	0.15	0.60	0.18	4	0.20	0.80	0.40	1	0.05	0.20	0.20	**5**	**0.25**	**1.00**	0.38	0	0.00	0.00	0.00
30	4	0.13	0.80	0.16	**5**	**0.17**	**1.00**	0.33	4	0.13	0.80	0.13	**5**	**0.17**	**1.00**	0.38	0	0.00	0.00	0.00
40	4	0.10	0.80	0.16	**5**	**0.13**	**1.00**	0.33	4	0.10	0.80	0.13	**5**	**0.13**	**1.00**	0.38	0	0.00	0.00	0.00
50	4	0.08	0.80	0.16	**5**	**0.10**	**1.00**	0.33	4	0.08	0.80	0.13	**5**	**0.10**	**1.00**	0.38	1	0.02	0.20	0.20
	DB				DB-MAX				RBDA				Our method				LOF			
Class '0'																				
5	1	0.20	0.20	0.33	2	0.40	0.40	0.75	2	0.40	0.40	0.43	3	0.60	0.60	**1.00**	0	0.00	0.00	0.00
10	1	0.10	0.20	0.33	3	0.30	0.60	0.67	3	0.30	0.60	0.50	3	0.30	0.60	**1.00**	0	0.00	0.00	0.00
15	1	0.07	0.20	0.33	4	0.27	0.80	0.33	4	0.27	0.80	0.40	**5**	**0.33**	**1.00**	0.60	0	0.00	0.00	0.00
20	1	0.05	0.20	0.33	4	0.20	0.80	0.33	4	0.20	0.80	0.40	**5**	**0.25**	**1.00**	0.35	0	0.00	0.00	0.00
30	2	0.07	0.40	0.10	**5**	**0.17**	**1.00**	0.33	4	0.13	0.80	0.40	**5**	**0.17**	**1.00**	0.35	1	0.03	0.20	0.04
40	4	0.10	0.80	0.10	**5**	**0.13**	**1.00**	0.33	4	0.10	0.80	0.40	**5**	**0.13**	**1.00**	0.35	1	0.03	0.20	0.04
50	4	0.08	0.80	0.10	**5**	**0.10**	**1.00**	0.33	4	0.08	0.80	0.40	**5**	**0.10**	**1.00**	0.35	1	0.02	0.20	0.04

Table 9
Multiple features, $\kappa=7$.

m	n	p	r	rp	n	p	r	rp	n	p	r	rp	n	p	r	rp	n	p	r	rp
	DB				LOF				RBDA				Our method				COF			
Class '9'																				
5	0	0.00	0.00	0.00	0	0.00	0.00	0.00	0	0.00	0.00	0.00	4	0.80	0.80	**1.00**	0	0.00	0.00	0.00
10	1	0.01	0.20	0.11	0	0.00	0.00	0.00	0	0.00	0.00	0.00	**5**	**0.50**	**1.00**	0.75	1	0.10	0.20	0.11
15	1	0.07	0.20	0.11	0	0.00	0.00	0.00	0	0.00	0.00	0.00	**5**	**0.33**	**1.00**	0.75	1	0.07	0.20	0.11
20	1	0.05	0.20	0.11	1	0.05	0.20	0.05	2	0.10	0.40	0.08	**5**	**0.25**	**1.00**	0.75	1	0.05	0.20	0.11
30	2	0.07	0.40	0.08	2	0.07	0.40	0.07	2	0.07	0.40	0.08	**5**	**0.17**	**1.00**	0.75	1	0.03	0.20	0.11
40	2	0.05	0.40	0.08	2	0.05	0.40	0.07	3	0.08	0.60	0.08	**5**	**0.13**	**1.00**	0.28	1	0.03	0.20	0.11
50	2	0.04	0.40	0.08	2	0.04	0.40	0.07	3	0.06	0.60	0.08	**5**	**0.10**	**1.00**	0.28	1	0.02	0.20	0.11
	DB				DB-MAX				RBDA				Our method				COF			
Class '6'																				
5	0	0.00	0.00	0.00	1	0.20	0.20	0.33	0	0.00	0.00	0.00	1	0.20	0.20	0.33	0	0.00	0.00	0.00
10	2	0.20	0.40	0.21	3	0.30	0.60	0.30	0	0.00	0.00	0.00	3	0.30	0.60	0.43	0	0.00	0.00	0.00
15	3	0.20	0.60	0.25	4	0.27	0.80	0.29	0	0.00	0.00	0.00	4	0.27	0.80	0.40	0	0.00	0.00	0.00
20	4	0.20	0.80	0.24	**5**	**0.25**	**1.00**	0.29	0	0.00	0.00	0.00	4	0.20	0.80	0.40	0	0.00	0.00	0.00
30	**5**	**0.17**	**1.00**	0.22	**5**	**0.17**	**1.00**	0.29	0	0.00	0.00	0.00	**5**	**0.17**	**1.00**	0.33	1	0.03	0.20	0.05
40	**5**	**0.13**	**1.00**	0.22	**5**	**0.13**	**1.00**	0.29	1	0.03	0.20	0.03	**5**	**0.13**	**1.00**	0.33	1	0.03	0.20	0.05
50	**5**	**0.10**	**1.00**	0.22	**5**	**0.10**	**1.00**	0.29	1	0.02	0.20	0.03	**5**	**0.10**	**1.00**	0.33	1	0.02	0.20	0.05
	DB				DB-MAX				RBDA				Our method				LOF			
Class '4'																				
5	2	0.40	0.40	0.60	1	0.20	0.20	0.50	1	0.20	0.20	0.33	2	0.40	0.40	0.60	1	0.20	0.20	
10	3	0.30	0.60	0.55	3	0.30	0.60	0.46	1	0.10	0.20	0.33	2	0.20	0.40	0.60	1	0.10	0.20	
15	3	0.20	0.60	0.55	4	0.27	0.60	0.38	2	0.13	0.40	0.20	3	0.20	0.60	0.33	1	0.07	0.20	
20	**5**	**0.25**	**1.00**	0.33	4	0.20	1.00	0.38	3	0.15	0.60	0.19	**5**	**0.25**	**1.00**	0.31	1	0.05	0.20	
30	**5**	**0.17**	**1.00**	0.33	**5**	**0.17**	**1.00**	0.28	4	0.10	0.80	0.19	**5**	**0.17**	**1.00**	0.31	2	0.07	0.40	
40	**5**	**0.13**	**1.00**	0.33	**5**	**0.10**	**1.00**	0.28	**5**	**0.13**	**1.00**	0.15	**5**	**0.13**	**1.00**	0.31	3	0.08	0.60	
50	**5**	**0.10**	**1.00**	0.33	**5**	**0.13**	**1.00**	0.28	**5**	**0.10**	**1.00**	0.15	**5**	**0.10**	**1.00**	0.31	4	0.07	0.80	

Table 10
Shuttle, $k=7,20,40,60$.

m	1.0				1.5				2.0				2.5				3.0			
	n	p	r	rp	n	p	r	rp	n	p	r	rp	n	p	r	rp	n	p	r	rp
$k=7$																				
20	11	0.55	0.58	0.42	9	0.45	0.47	0.35	5	0.25	0.26	0.23	5	0.25	0.26	0.23	5	0.25	0.26	0.23
30	14	0.47	0.74	0.45	12	0.40	0.63	0.36	9	0.30	0.47	0.28	9	0.30	0.47	0.28	9	0.30	0.47	0.28
60	17	0.28	0.89	0.32	16	0.27	0.84	0.33	13	0.22	0.68	0.24	13	0.22	0.68	0.24	13	0.22	0.68	0.24
90	18	0.20	0.95	0.28	17	0.19	0.89	0.31	15	0.17	0.79	0.22	15	0.17	0.79	0.22	15	0.17	0.79	0.22
120	**19**	**0.16**	**1.00**	0.23	18	0.15	0.95	0.25	16	0.13	0.84	0.21	16	0.13	0.84	0.21	16	0.13	0.84	0.21
140	**19**	**0.14**	**1.00**	0.23	18	0.1	0.95	0.25	17	0.12	0.89	0.20	17	0.12	0.89	0.20	17	0.12	0.89	0.20
200	**19**	**0.14**	**1.00**	0.14	**19**	**0.09**	**1.00**	0.12	18	0.10	0.95	0.10	18	0.10	0.95	0.10	18	0.10	0.95	0.10
$k=20$																				
20	11	0.55	0.58	0.42	11	0.55	0.58	0.40	8	0.40	0.42	0.31	8	0.40	0.42	0.31	8	0.40	0.42	0.31
30	14	0.47	0.74	0.45	14	0.47	0.74	0.42	13	0.43	0.68	0.39	13	0.43	0.68	0.39	13	0.43	0.68	0.39
60	18	0.30	0.95	0.32	18	0.30	0.95	0.37	18	0.30	0.95	0.33	18	0.30	0.95	0.33	18	0.30	0.95	0.33
90	**19**	**0.21**	**1.00**	0.28	**19**	**0.21**	**1.00**	0.33	**19**	**0.21**	**1.00**	0.27	**19**	**0.21**	**1.00**	0.27	**19**	**0.21**	**1.00**	0.27
120	**19**	**0.16**	**1.00**	0.23	**19**	**0.16**	**1.00**	0.27	**19**	**0.16**	**1.00**	0.26	**19**	**0.16**	**1.00**	0.26	**19**	**0.16**	**1.00**	0.26
140	**19**	**0.14**	**1.00**	0.23	**19**	**0.14**	**1.00**	0.27	**19**	**0.14**	**1.00**	0.23	**19**	**0.14**	**1.00**	0.23	**19**	**0.14**	**1.00**	0.23
200	**19**	**0.14**	**1.00**	0.14	**19**	**0.09**	**1.00**	0.12	**19**	**0.10**	**1.00**	0.10	**19**	**0.10**	**1.00**	0.10	**19**	**0.10**	**1.00**	0.10
$k=40$																				
20	11	0.55	0.58	0.42	11	0.55	0.58	0.40	11	0.55	0.58	0.51	11	0.55	0.58	0.51	11	0.55	0.58	0.51
30	14	0.47	0.74	0.45	14	0.47	0.74	0.42	14	0.47	0.74	0.44	14	0.47	0.74	0.44	14	0.47	0.74	0.44
60	18	0.30	0.95	0.32	18	0.30	0.95	0.37	18	0.30	0.95	0.33	18	0.30	0.95	0.33	18	0.30	0.95	0.33
90	**19**	**0.21**	**1.00**	0.28	**19**	**0.21**	**1.00**	0.33	**19**	**0.21**	**1.00**	0.27	**19**	**0.21**	**1.00**	0.27	**19**	**0.21**	**1.00**	0.27
120	**19**	**0.16**	**1.00**	0.23	**19**	**0.16**	**1.00**	0.27	**19**	**0.16**	**1.00**	0.26	**19**	**0.16**	**1.00**	0.26	**19**	**0.16**	**1.00**	0.26
140	**19**	**0.14**	**1.00**	0.14	**19**	**0.14**	**1.00**	0.27	**19**	**0.14**	**1.00**	0.23	**19**	**0.14**	**1.00**	0.23	**19**	**0.14**	**1.00**	0.23
200	**19**	**0.14**	**1.00**	0.14	**19**	**0.09**	**1.00**	0.12	**19**	**0.10**	**1.00**	0.10	**19**	**0.10**	**1.00**	0.10	**19**	**0.10**	**1.00**	0.10
$k=60$																				
20	11	0.55	0.58	0.42	11	0.55	0.58	0.40	11	0.55	0.58	0.51	11	0.55	0.58	0.51	11	0.55	0.58	0.51
30	14	0.47	0.74	0.45	14	0.47	0.74	0.42	14	0.47	0.74	0.44	14	0.47	0.74	0.44	14	0.47	0.74	0.44
60	18	0.30	0.95	0.32	18	0.30	0.95	0.37	18	0.30	0.95	0.33	18	0.30	0.95	0.33	18	0.30	0.95	0.33
90	**19**	**0.21**	**1.00**	0.28	**19**	**0.21**	**1.00**	0.33	**19**	**0.21**	**1.00**	0.27	**19**	**0.21**	**1.00**	0.27	**19**	**0.21**	**1.00**	0.27
120	**19**	**0.16**	**1.00**	0.23	**19**	**0.16**	**1.00**	0.27	**19**	**0.16**	**1.00**	0.26	**19**	**0.16**	**1.00**	0.26	**19**	**0.16**	**1.00**	0.26
140	**19**	**0.14**	**1.00**	0.14	**19**	**0.14**	**1.00**	0.27	**19**	**0.14**	**1.00**	0.23	**19**	**0.14**	**1.00**	0.23	**19**	**0.14**	**1.00**	0.23
200	**19**	**0.14**	**1.00**	0.14	**19**	**0.09**	**1.00**	0.12	**19**	**0.10**	**1.00**	0.10	**19**	**0.10**	**1.00**	0.10	**19**	**0.10**	**1.00**	0.10

Table 11
Ionosphere, $k = 5,10,20,30$.

m	1.0				1.5				2.0				2.5				3.0			
	n	p	r	rp	n	p	r	rp	n	p	r	rp	n	p	r	rp	n	p	r	rp
$k=5$																				
5	**5**	**1.00**	**0.04**	**1.00**	**5**	**1.00**	**0.04**	**1.00**	**5**	**1.00**	**0.04**	**1.00**	**5**	**1.00**	**0.04**	**1.00**	**5**	**1.00**	**0.04**	**1.00**
10	**10**	**1.00**	**0.08**	**1.00**	**10**	**1.00**	**0.08**	**1.00**	**10**	**1.00**	**0.08**	**1.00**	**10**	**1.00**	**0.08**	**1.00**	**10**	**1.00**	**0.08**	**1.00**
30	**30**	**1.00**	**0.24**	**1.00**	**30**	**1.00**	**0.24**	**1.00**	27	0.90	0.21	0.93	25	0.83	0.20	0.90	25	0.83	0.20	0.90
60	**60**	**1.00**	**0.48**	**1.00**	59	0.98	0.47	**1.00**	47	0.78	0.37	0.85	44	0.73	0.35	0.80	44	0.73	0.35	0.80
90	**90**	**1.00**	**0.71**	**1.00**	83	0.92	0.66	**1.00**	63	0.70	0.50	0.78	59	0.66	0.47	0.73	59	0.66	0.47	0.73
120	109	0.91	0.87	0.95	94	0.78	0.75	0.97	72	0.60	0.57	0.74	67	0.56	0.53	0.70	67	0.56	0.53	0.70
130	113	0.87	0.90	0.93	96	0.74	0.76	0.96	74	0.57	0.59	0.73	69	0.53	0.55	0.69	69	0.53	0.55	0.69
140	116	0.83	0.92	0.92	98	0.70	0.78	0.95	79	0.56	0.63	0.71	72	0.51	0.57	0.67	72	0.51	0.57	0.67
$k=10$																				
5	**5**	**1.00**	**0.04**	**1.00**	**5**	**1.00**	**0.04**	**1.00**	**5**	**1.00**	**0.04**	**1.00**	**5**	**1.00**	**0.04**	**1.00**	**5**	**1.00**	**0.04**	**1.00**
10	**10**	**1.00**	**0.08**	**1.00**	**10**	**1.00**	**0.08**	**1.00**	**10**	**1.00**	**0.08**	**1.00**	**10**	**1.00**	**0.08**	**1.00**	**10**	**1.00**	**0.08**	**1.00**
30	**30**	**1.00**	**0.24**	**1.00**	**30**	**1.00**	**0.24**	**1.00**	**30**	**1.00**	**0.24**	**1.00**	**30**	**1.00**	**0.24**	**1.00**	**30**	**1.00**	**0.24**	**1.00**
60	**60**	**1.00**	**0.48**	**1.00**	**60**	**1.00**	**0.48**	**1.00**	**60**	**1.00**	**0.48**	**1.00**	**60**	**1.00**	**0.48**	**1.00**	**60**	**1.00**	**0.48**	**1.00**
90	**90**	**1.00**	**0.71**	**1.00**	89	0.99	0.71	**1.00**	76	0.84	0.60	0.90	73	0.81	0.58	0.87	73	0.81	0.58	0.87
120	109	0.91	0.87	0.95	96	0.80	0.76	0.99	83	0.70	0.66	0.82	79	0.66	0.63	0.79	79	0.66	0.63	0.79
130	114	0.88	0.90	0.93	99	0.76	0.79	0.97	86	0.66	0.68	0.80	82	0.63	0.65	0.77	82	0.63	0.65	0.77
140	116	0.83	0.92	0.92	100	0.71	0.79	0.97	89	0.64	0.71	0.76	85	0.61	0.67	0.74	85	0.61	0.67	0.74
$k=20$																				
5	**5**	**1.00**	**0.04**	**1.00**	**5**	**1.00**	**0.04**	**1.00**	**5**	**1.00**	**0.04**	**1.00**	**5**	**1.00**	**0.04**	**1.00**	**5**	**1.00**	**0.04**	**1.00**
10	**10**	**1.00**	**0.08**	**1.00**	**10**	**1.00**	**0.08**	**1.00**	**10**	**1.00**	**0.08**	**1.00**	**10**	**1.00**	**0.08**	**1.00**	**10**	**1.00**	**0.08**	**1.00**
30	**30**	**1.00**	**0.24**	**1.00**	**30**	**1.00**	**0.24**	**1.00**	**30**	**1.00**	**0.24**	**1.00**	**30**	**1.00**	**0.24**	**1.00**	**30**	**1.00**	**0.24**	**1.00**
60	**60**	**1.00**	**0.48**	**1.00**	**60**	**1.00**	**0.48**	**1.00**	**60**	**1.00**	**0.48**	**1.00**	**60**	**1.00**	**0.48**	**1.00**	**60**	**1.00**	**0.48**	**1.00**
90	**90**	**1.00**	**0.71**	**1.00**	**90**	**1.00**	**0.71**	**1.00**	88	0.98	0.70	0.96	86	0.96	0.68	0.97	86	0.96	0.68	0.97
120	117	0.97	0.93	0.97	103	0.86	0.82	1.00	96	0.80	0.76	0.89	93	0.78	0.74	0.86	93	0.78	0.74	0.86
130	121	0.93	0.96	0.97	105	0.81	0.83	0.99	97	0.75	0.77	0.86	94	0.72	0.75	0.83	94	0.72	0.75	0.83
140	123	0.88	0.98	0.96	106	0.76	0.84	0.98	99	0.71	0.79	0.81	96	0.69	0.76	0.79	96	0.69	0.76	0.79
$k=30$																				
5	**5**	**1.00**	**0.04**	**1.00**	**5**	**1.00**	**0.04**	**1.00**	**5**	**1.00**	**0.04**	**1.00**	**5**	**1.00**	**0.04**	**1.00**	**5**	**1.00**	**0.04**	**1.00**
10	**10**	**1.00**	**0.08**	**1.00**	**10**	**1.00**	**0.08**	**1.00**	**10**	**1.00**	**0.08**	**1.00**	**10**	**1.00**	**0.08**	**1.00**	**10**	**1.00**	**0.08**	**1.00**
30	**30**	**1.00**	**0.24**	**1.00**	**30**	**1.00**	**0.24**	**1.00**	**30**	**1.00**	**0.24**	**1.00**	**30**	**1.00**	**0.24**	**1.00**	**30**	**1.00**	**0.24**	**1.00**
60	**60**	**1.00**	**0.48**	**1.00**	**60**	**1.00**	**0.48**	**1.00**	**60**	**1.00**	**0.48**	**1.00**	**60**	**1.00**	**0.48**	**1.00**	**60**	**1.00**	**0.48**	**1.00**
90	**90**	**1.00**	**0.71**	**1.00**	**90**	**1.00**	**0.71**	**1.00**	**90**	**1.00**	**0.73**	**1.00**	**90**	**1.00**	**0.73**	**1.00**	**90**	**1.00**	**0.73**	**1.00**
120	**120**	**1.00**	**0.71**	**1.00**	107	0.89	0.85	**1.00**	98	0.82	0.78	0.91	96	0.80	0.76	0.87	96	0.80	0.76	0.87
130	123	0.95	0.98	0.99	108	0.83	0.86	0.99	98	0.75	0.78	0.87	96	0.74	0.76	0.84	96	0.74	0.76	0.84
140	124	0.89	0.98	0.97	109	0.78	0.86	0.98	100	0.71	0.79	0.82	97	0.70	0.77	0.80	97	0.70	0.77	0.80

4.2.3. داده یونوسفر

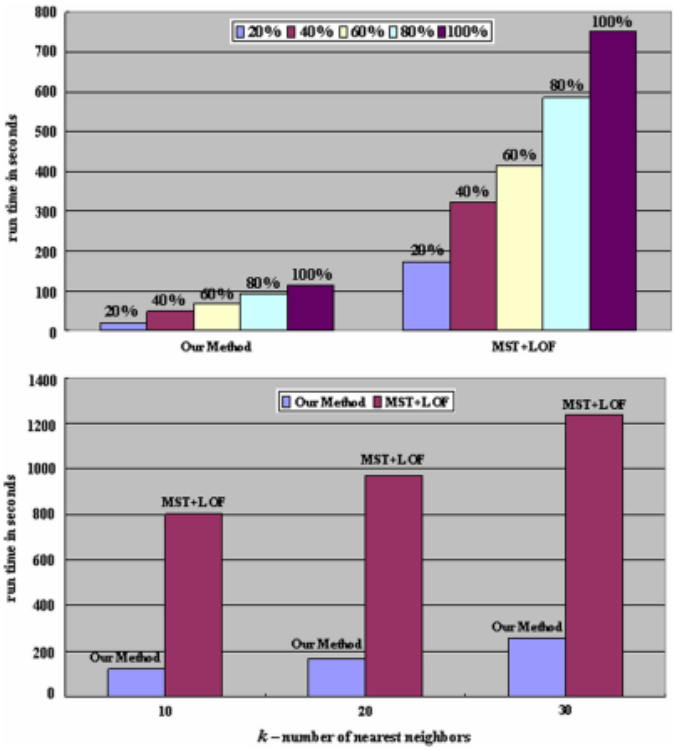
مجموعه داده ی یوتوسفر دانشگاه جان هاپکینز شامل 351 نمونه با 34 ویژگی است که 225 مورد آن به عنوان نمونه های خوب برچسب خورده اند در حالیکه 126 تای باقی مانده به عنوان نمونه های بد برچسب خورده اند. همه ی ویژگی ها در محدوده ی صفر تا یک نرمالیزه شده اند و هیچ نمونه ی تکثیر شده ای در مجموعه داده وجود ندارد. به جای گزینش تصادفی نمونه از کلاس بد در جهت شکل دهی کلاس های نادر، ما از همه ی 351 نمونه استفاده می کنیم و عملکرد همه ی روش ها را برای چهار k مختلف و هشت m مختلف مقایسه می کنیم. نتایج تجربیِ بدست آمده بوسیله ی PBOD، روش ما و سه مورد از بهترین روش های باقی مانده در جدول 7 ارائه می شوند.

به وضوح می توان از جدول دید که بهخ طور کلی، روش ما بهترین عملکرد را دارد و روش DB عملکرد بعدی را دارد. RBDA نسبت به MST+LOF اندکی بهتر عمل می کند. PBOD بدترین عملکرد را دارد. همه ی روش ها برای این مجموعه داده ی آزمایش، در مقایسه با آزمایشات قبلی حساسیت کمتری نسبت به k دارند.

ما برای مجموعه های داده با ابعاد بالا، دو مجموعه داده (ویژگی های چندگانه و ارقام نوری) از UCI برمیداریم.

4.2.4. ارقام نوری

مجموعه داده ی ارقام نوری متشکل از داده ی بیرون کشیده شده از اعداد دست نویس (0-9) است و شامل 5620 نمونه با 645 ویژگی می شود. برای بدست آوردن مجموعه داد، طرح های بیتی نرمالیزه شده ی ارقام دست نویس از شکل پیش چاپ شده بیرون کشیده می شوند. در ادامه، طرح های بیتی 32*32 به بلوک های 4*4 ای تقسیم می شوند که با هم همپوشانی ندارند و تعداد پیکسل ها در هر بلوک محاسبه می شوند و یک ماتریس ورودی 8*8 تولید می کنند که در آن هر مولفه یک عدد صحیح در محدوده ی صفر تا 16 است. از آنجایی که کلاس نادر کوچک وجود ندارد تا به عنوان بخش مجزا لحاظ شود، به صورت تصادفی پنج نقطه ی داده را از یکی از کلاس ها برداشتیم تا آن را نادر کنیم (به عنوان بخش مجزا). در جدول 8، بهترین نتایج تجربیِ روشمان را برای دو کلاس (2 و صفر) و چهار مورد از بهترین روش های باقی مانده برای k=7و هفت مقدار m نشان دادیم.



شکل13: عملکرد زمان راه اندازیِ الگوریتم ما برای داده ی IPUMAS

از جدول به وضوح دیده می شود که بهخ طور کلی، روش ما بهتر کار می کند و روش DB-MAX عملکرد بعدی را دارد. RBDA اندکی بهتر از روش DB عمل می کند.

4.2.5. ویژگی های چندگانه

مجموعه های داده ی ویژگی های چندگانه متشکل از داده ی شماره های دست نویس (0-9) است اما از یک مجموعه ی نقشه های ابزار هلندی بیرون کشیده شده و شامل 2000 نمونه با 649 ویژگی می شود. این ویژگی ها شامل شش مجموعه ی ویژگی (76 ضریب فوریه، 216 همبستگی مشخصات، 64 ضریب عشق-کارهونن، 240 میانگین پیکسل در 23 پنجره، 47 لحظه ی زرنیک و 6 ویژگی مورفولوژیکی) می شوند. از آنجایی که همه ی این کلاس ها، هر کدام شامل 200 نقطه ی داده می شود و کلاس های نادر وجود ندارد تا به عنوان بخش مجزا لحاظ شوند، به طور تصادفی 5 نقطه ی داده را از یکی از کلاس ها انتخاب می کنیم تا آن را نادر کنیم (به عنوان بخش های مجزا). جدول 9 ، بهترین یافته ها از نتایج تجربی روش ما برای سه کلاس (9 و 6 و 4) و چهار مورد از بهترین روش های باقی مانده را برای k=7 و هفت مقدار m را خلاصه می کند.

به وضوح از جدول می توان دید که به طور کلی روش ما به صورت ایده آل کار می کند چون که دیگران بسیار بدتر عمل می کنند. برای دو مورد دیگر، روش ما و روش DB-MAX به طور یکسان عمل می کنند. DB برای کلاس های 6 و 4 به خوبی عمل می کند. RBDA برای کلاس 4 به خوبی عمل می کند. بنابراین، به طور کلی، روش ما بهترین عملکرد را دارد.

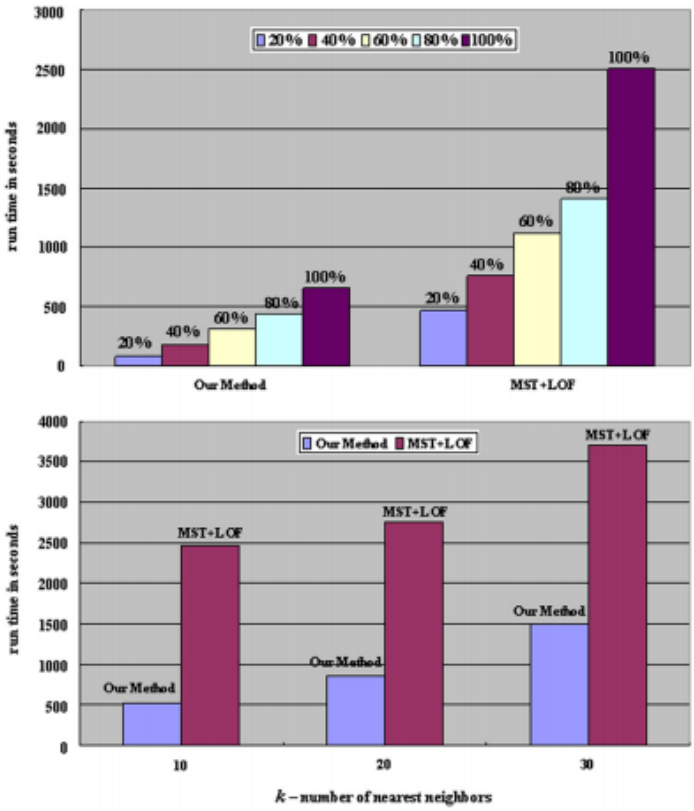
4.3. عملکرد الگوریتم ما با SOM-TH

در نتایج بالا، آستانه را (SOM-TH) به طور ثابت روی 1.5 نگاه می داریم. در این زیربخش، آزمایشات را برای مطالعه ی شناسایی کارآمدیِ روش پیشنهاد شده بوسیله ی SOM-TH متغیر در محدوده ی (1 و 1.5 و 2 و 2.5 و 3) انجام دادیم و نتایج برای دو مجموعه داده ی شاتل و یونوسفر در جداول 10 و 11 نشان داده می شود. از جداول

می توانیم ببینیم که روش های ما در زمانی که SOM-TH برای گرفتن مقدار 1 تنظیم می شود، می تواند نتایج بهتری بدست آورد.

4.4. عملکرد زمان راه اندازی برای الگوریتم ما روی مجموعه داده هایی با ابعاد بالا

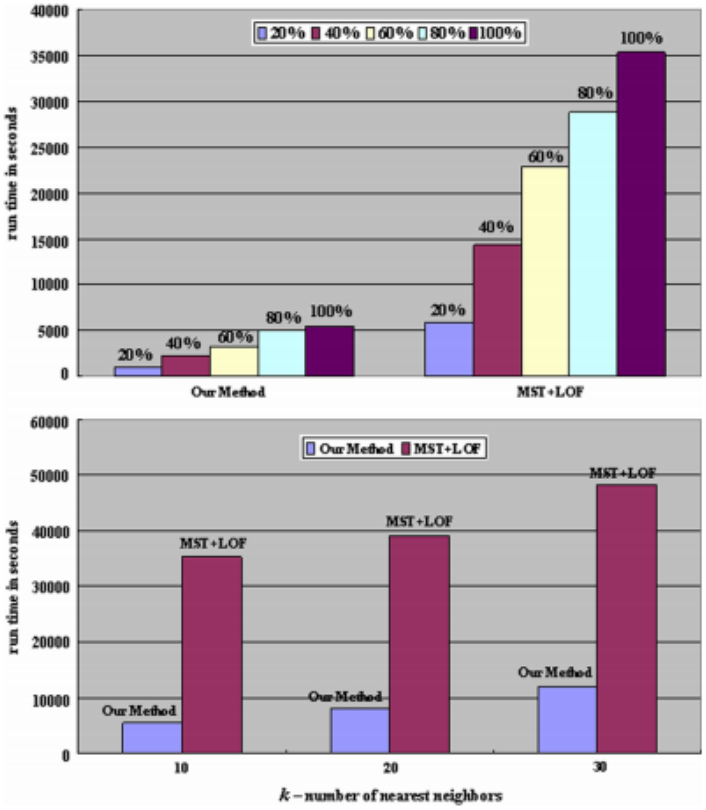
در این زیربخش، مطالعه ی خود را روی رفتار زمان راه اندازیِ روش پیشنهاد شده با پارامترهای متغیر k و تحت بارهای کاری مختلف متمرکز می کنیم. خصوصا می خواهیم تا تاثیر اندازه ی مجموعه داده روی عملکرد الگوریتممان را نشان بدهیم (مقیاس پذیری الگوریتم ما) و زمان راه اندازی الگوریتممان را با زمان راه اندازی الگوریتم MST+LOF روی سه مجموعه داده با ابعاد بالای مختلف مقایسه کنیم. این سه مجموعه داده می توانند از UCI دانلود شوند.این مجموعه های داده به اختصار در جدول 1 خلاصه می شوند. همه ی ویژگی های دسته بندی شده در آنها (اگر وجود داشته باشند) به مقادیر صحیح تبدیل شده اند.



شکل 14: عملکرد زمان راه اندازیِ الگوریتم ما برای داده ی نوع پوشش

یک مشکل بزرگ در زمان ارزیابی روش های شناسایی بخش های مجزا برای داده با ابعاد بالا این است که مجموعه های داده ی دنیای واقعیِ بسیار کمی در جایی که دقیقا مشخص است کدام اشیاء واقعا به علت تعلق به یک مکانیزم نادر و مختلف، متفاوت رفتار می کنند، وجود دارد. معمولا از پیش، بخش مجزا بودن کدام شیء و چقدر از اشیاء داده نشده است. اگر چه آنجا چندین مورد مطالعه در مورد شناسایی بخش مجزا وجود دارد اما این سوال که آیا یک شیء ، یک بخش مجزا است یا نه، معمولا به نگر ش بستگی دارد. مشکلی دیگر این است که لیست بخش های مجزای ممکن معمولا ناقص است. این موضوع، آن را برای ارزیابی اینکه آیا الگوریتم همه ی بخش های مجزا در پایگاه داده را به خوبی رتبه بندی می کند یا نه، دشوار می کند. از آنجایی که از زیربخش های قبلی نشان داده شده است که روش شناسایی بخش مجزای ما قابل قیاس با دیگر طرح های شناسایی بخش مجزای مدرن است (اگر کارآمدتر از آنها نباشد)، روی مقایسه کردن رویکردمان با الگوریتم MST+LOF در مورد موضوع زمان راه اندازی تمرکز می کنیم. حساسیت DHCA به پارامتر ورودی k (تعداد مراکز قسمت ها در هر مرحله از DHCA) در 21 مورد مطالعه قرار گرفته است. برای k های بزرگ، فواصل بیشتر به مراکز قسممت ها نیاز به محاسبه شدن دارند و زمان راه اندازی به همراه k افزایش می یابد. در این مجموعه از آزمایشات، ورودی k به DHCA روی 5 تنظیم شده است.

از آنجایی که بخش های مجزا تنها برای بخش های بسیار کوچک در یک مجموعه داده لحاظ می شود، در این مجموعه آزمایشات، نتایج زمان راه اندازی روش ما و روش MST+LOF در جهت کاویدن سی بخش مجزای برتر برای سه مجموعه داده ی واقعی تحت بارهای کاری مختلف برای k=10 و با k متغیر (تعداد نزدیک ترین همسایه ها از ده تا سی) برای داده ی IPUMS در شکل 13 و برای داده ی نوع پوشش در شکل 14 و برای داده ی آمار امریکا درشکل 15 ارائه می شوند. برای این مقایسه، آستانه ی شناسایی بخش مجزای سراسری ما $SOM_{MST}=3$ است، و آستانه ی شناسایی بخش مجزای محلی ما روی دو تنظیم می شود.



شکل 15: عملکرد زمان راه اندازیِ الگوریتم ما برای داده ی آماری ایالات متحده

در هر شکل دو نمودار میله ای برای هر مجموعه داده وجود دارد. قسمت بالاییِ هر شکل زمان راه اندازی عملکردِ الگوریتم ما و روش MST+LOF برای پنج اندازه ی داده ی مختلف (در پنج رنگ مختلف) را ارائه می کند. این مقادیر زمان راه اندازی بوسیله ی تغییر دادن اندازه های داده بین 20٪ و 100٪ از کل هر مجموعه داده بدست آمده اند. نتایج روش ما در سمت چپ نشان داده شده اند در حالیکه نتایج الگوریتم MST+LOF در سمت راست نشان داده می شوند. قسمت پایینی هر شکل زمان راه اندازی عملکردِ الگوریتم ما (به رنگ آبی) و روش MST+LOF (به رنگ بنفش) را برای سه k مختلف (k=10,20,30) نشان می دهدزمان اجرای الگوریتم ما به طور واضح با اندازه ی داده و شکل k افزایش می یابد و در حدود پنج برابر سریعتر از زمان اجرایی MST+LOF است. از انجایی که این مجموعه های داده ویژگی های بسیار مختلفی دارند و از ابعاد مختلفی هستند، به طور کلی، می توان دید که الگوریتم ما بهتر از MST+LOF عمل می کند.

5. نتیجه گیری

در این مقاله، یک روش شناسایی بخش مجزای کارآمد بر پایه ی kNN و الگو گرفته از MST ارائه کرده ایم که می تواند هر دو بخش مجزای سراسری و محلی را شناسایی کند. علاوه بر قابل قیاس بودن با روش های شناسایی بخش مجزای فاصله بنیان مرسوم، رویکرد ما در شناسایی بخش های مجزا نیز بهتر عمل می کند. لازم به ذکر است که این بخش های مجزا از الگوهای اصلی در مجموعه داده ی داده شده منحرف می شوند. کاندیداهای بخش های مجزابر مبنای امتیازات بخش مجزای الگو گرفته از MST که به هر نقطه ی داده تخصیص داده می شوند، رتبه بندی می شوند. برای نشان دادن سودمندیِ مولفه های بخش مجزای ارائه شده ی ما، یک مقایسه ی دقیق از عملکرد آن با تعدادی از روش های شناسایی بخش مجزای مدرن انجام داده ایم. از طریق یک ارزیابی کامل، روش ما توانایی اش را برای رتبه بندی بهترین کاندیداها برای بخش مجزا بودن با دقت و یادگیری بالا، نشان می دهد. علاوه بر رویکرد پایه، پیشنهاد می کنیم تا از DHCA به عنوان یک تشدید مناسب برای مجموعه های داده ی بزرگ با ابعاد بالا استفاده شود. نشان داده شده است که تعداد کمی از راه اندازی متوالیِ DHCA میتواند شناسایی بخش های مجزای برتر را ساده سازی کند. توجیه تئوری و اعتبار سنجی تجربی ، هر دو کارآمدی روش پیشنهاد شده را نشان می دهند.

همچنین مطالعه ی ما نشان می دهد که نباید یک روش را به صورت برتر در همه ی جنبه ها از دیگر روش ها ببینیم بلکه باید از آن به عنوان یک متمم (به جای یک جایگزین) برای دیگر روش ها در کاربردهایی با ملزومات مختلف استفاده کنیم. این موضوع ذاتی است چون ، در واقعیت، معمولا شناسایی همه ی بخش های مجزایی که با بینش های کاربر تناسب دارند دشوار است. بنابراین، ترکیب کردن روش شناسایی بخش مجزای پیشنهادی ما به عنوان یک مولفه در چارچوب شناسایی بخش مجزای کنونی، احتمالا معنادار به نظر برسد.

References

[1] D.M. Hawkins, Identification of outliers, Monographs on Applied Probability and Statistics, Chapman and Hall, London, 1980.

[2] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, S. Stolfo. A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data. in: Daniel Barbará and Sushil Jajodia (Eds.), Data Mining for Security Applications, 2002, pp.77-101.

[3] T. Lane, C.E. Brodley, Temporal sequence learning and data reduction for anomaly detection, ACM Trans. Inf. Syst. Secur. 2 (3) (1999) 295–331.

[4] R.J. Bolton, J.H. David, Unsupervised profiling methods for fraud detection, Stat. Sci. 17 (3) (2002) 235–255.

[5] W. Wong, A. Moore, G. Cooper, and M. Wagner, Rule-based anomaly pattern detection for detecting disease outbreaks, in: Proceedings of the 18th National Conference on Artificial Intelligence, 2002, pp. 217–223.

[6] B. Sheng, Q. Li, W. Mao, W. Jin, Outlier detection in sensor networks, in: Proceedings of ACM International Symposium on Mobile Ad Hoc Networking and Computing, 2007, pp. 219–228.

[7] V.J. Hodge, J. Austin, A survey of outlier detection methodologies, Artif. Intell. Rev. 22 (2004) 85–126.

[8] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: a survey, ACM Comput. Surv. 41 (July) (2009). (Article 15).

[9] X. Wang, X.L. Wang, D.M. Wilkes, A spanning tree-inspired clustering based outlier detection technique, in: Proceedings of the 12th Industry Conference on Data Mining, Berlin, Germany, 2012, pp. 209–223.

[10] E.M. Knorr, R.T. Ng, Algorithms for mining distance-based outliers in large datasets, in: Proceedings of the 24th VLDB Conference, New York, USA,1998, pp. 392–403.

[11] S. Ramaswamy, R. Rastogi, K. Shim, Efficient algorithms for mining outliers from large data sets, in: Proceedings of the ACM SIGMOD Conference, 2000, pp. 427–438.

[12] F. Angiulli, C. Pizzuti, Fast outlier detection in high dimensional spaces, Proceedings of the Sixth European Conference on the Principles of Data Mining and Knowledge Discovery (2002) 15–26.

[13] M.M. Breuning, H.P. Kriegel, R.T. Ng, J. Sander, LOF: identifying density-based local outliers, in: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, 2000, pp. 93–104.

[14] J. Tang, Z. Chen, A.W.C. Fu, D.W. Cheung, Enhancing effectiveness of outlier detections for low density patterns, in: Proceedings of the 6th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), vol. 2336, Taipei, Taiwan, 2002, pp. 535–548.

[15] P.B. Gibbons, S. Papadimitriou, H. Kitagawa, C. Faloutsos, LOCI: fast outlier detection using the local correlation integral, in: Proceedings of the IEEE 19th International Conference on Data Engineering, Bangalore, India, 2003, pp. 315–328.

[16] P. Sun, S. Chawla, On local spatial outliers, in: Proceedings of the 4th International Conference on Data Mining (ICDM), Brighton, UK, 2004, pp. 209–216.

[17] W. Jin, A.K.H. Tung, J. Han, W. Wang, Ranking outliers using symmetric neighborhood relationship, in: Proceedings of the 10th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), vol. 3918, Singapore, 2006, pp. 577–579.

[18] K. Zhang, M. Hutter, H. Jin., A new local distance-based outlier detection approach for scattered real-world data, Adv. Knowl. Discov. Data Min. 5476 (2009) 813–822.

[19] H. Huang, K. Mehrotra, C.K. Mohan, Rank-based outlier detection, J. Stat. Comput. Simul. 83 (3) (2013) 1–14.

[20] C.T. Zahn, Graph-theoretical methods for detecting and describing gestalt clusters, IEEE Trans. Comput. C-20 (1971) 68–86.

[21] X. Wang, X.L. Wang, D.M. Wilkes., A divide-and-conquer approach for minimum spanning tree-based clustering, IEEE TKDE 21 (7) (2009) 945–958.

[22] C. Zhong, D. Miao, R. Wang., A graph-theoretical clustering method based on two rounds of minimum spanning trees, Pattern Recognit. 43 (3) (2010) 752–766.
[23] T. Luo, C. Zhong., A neighborhood density estimation clustering algorithm based on minimum spanning tree, LNAI 6401 (2010) 557–565.
[24] T. Luo, C. Zhong, H. Li, X. Sun, A multi-prototype clustering algorithm based on minimum spanning tree, in: Proceedings of 2010 7th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2010), 2010, pp. 1602–1607.
[25] C. Zhong, D. Miao, P. Franti, Minimum spanning tree based splitand-merge: a hierarchical clustering method, Inf. Sci. 181 (2011) 3397–3410.
[26] F.J. Rohlf, Generalization of the gap test for the detection of multivariate outliers, Biometrics 31 (1975) 93–101.
[27] M.F. Jiang, S.S. Tseng, C.M. Su., Two-phase clustering process for outliers detection, Pattern Recognit. Lett. 22 (2001) 691–700.
[28] J. Lin, D. Ye, C. Chen, M. Gao, Minimum spanning tree based spatial outlier mining and its applications, Lecture Notes Computer Science, vol. 5009, Springer-Verlag, Berlin/Heidelberg, 2008, 508–515.
[29] K. Beyer, J. Goldstein, R. Ramakrishnan, U. Shaft, When is nearest neighbors meaningful?, in: Proceedings of ICDT, 1999, pp. 217–235.
[30] C. Aggarwal, P. Yu, An effective and efficient algorithm for highdimensional outlier detection, Int. J Very Large Data Bases 14 (2) (2005) 211–221.
[31] S. Parthasarathy, C.C. Aggarwal, On the use of conceptual reconstruction for mining massively incomplete data sets, IEEE TKDE 15 (6) (2003) 1512–1531.
[32] X. Wang, X.L. Wang, D.M. Wilkes, Modifying iDistance for a fast CHAMELEON with application to patch based image segmentation, in: Proceedings of the 9th IASTED International Conference on Signal Processing, Pattern Recognition and Applications (SPPRA 2012), Crete, Greece, 2012, pp. 107–114.
[33] UCI: The UCI KDD Archive, University of California, Irvine, CA. ⟨http://kdd.ics.uci.edu/⟩.
[34] C. Aggarwal, P. Yu, Outlier detection for high-dimensional data, in: Proceedings of SIGMOD'01, Santa Barbara, CA, USA, 2001, pp. 37–46.
[35] X. Meng, Z. Chen, On user-oriented measurements of effectiveness of web information retrieval systems, in: H.R. Arabnia, O. Droegehorn (Eds.), Proceedings of the International Conference on Internet Computing, vol. 1, Las Vegas, Nevada, USA, 2004, pp. 527–533.