تاریخ دریافت مقاله: ۹۶/۰۷/۲۲
تاریخ پذیرش مقاله: ۹۷/۰۳/۲۷

خوشه‌بندی داده‌های جریان داده دارای برچسب

زهرا بیات
دانشجوی کارشناسی ارشد دانشکده ریاضی، آمار و علوم کامپیوتر، پردیس علوم، دانشگاه تهران
پست الکترونیکی: bayat.zahra@ut.ac.ir

هدیه ساجدی*
استادیار دانشکده ریاضی، آمار و علوم کامپیوتر، پردیس علوم، دانشگاه تهران
پست الکترونیکی: hhsajedi@ut.ac.ir

چکیده

امروزه مسئلهٔ پیداکردن الگوریتم‌های خوشه‌بندی نظارت‌شده برای داده‌های پویا و جریان داده اهمیت زیادی دارد. محققان سعی می‌کنند برای حل کردن این مسئله الگوریتم‌های جدید ارائه نمایند و یا الگوریتم‌های موجود را بهبود دهند. در میان این الگوریتم‌ها، روش SAIC[1] برای خوشه‌بندی داده‌های پویا با خوشه‌های با اندازه و شکل دلخواه ارائه شده‌است. در این روش، تعداد خوشه‌ها به طور خودکار توسط الگوریتم مشخص می‌شود اما این الگوریتم قادر به تشخیص صحیح خوشه‌های مسائل تک دسته‌ای نیست. این امر موجب می‌شود که بعضاً در خوشه‌بندی جریان داده اختلال ایجاد شود. در این مقاله ضمن توضیح علت ایجاد این مشکل، الگوریتم ISAIC[2] برای بهبود الگوریتم SAIC پیشنهاد می‌شود. همچنین عملکرد الگوریتم ISAIC با الگوریتم SAIC روی چند مجموعه داده مورد مقایسه قرار گرفته و نتایج ارایه شده است. میزان بهبود دقت دسته‌بندی بر روی مجموعه داده‌های مورد آزمایش حداقل صفر و حداکثر حدود ۶۵٪ است.

واژه‌های کلیدی: توده جریان داده[3]، جریان داده[4]، خوشه‌بندی[5]، خوشه‌بندی نظارت شده[6].

۱-مقدمه

واژه جریان داده اشاره بر توالی سریع اطلاعاتی که بالقوّه حجیم و پیوسته هستند دارد[۱۲]. کاربردهای گوناگونی وجود دارند که این‌گونه داده‌ها را تولید می‌کنند که ازجمله آن‌ها می‌توان به سیستم‌های نظارت بر شبکه، سیستم‌های تحلیل فروش، سیستم‌های ارتباطی و جریان کلیک کاربران اشاره کرد [۵].

تفاوت جریان داده با داده ایستا را می‌توان در موارد زیر برشمرد[۶]:

- داده‌ها در جریان داده برخلاف داده‌های ایستا به صورت برخط و به مرور دریافت می‌شوند.
- هیچ کنترلی بر روی ترتیب ورود عناصر جریان داده وجود ندارد.
- در حالت کلی اندازه یک جریان داده نامحدود است.

* نویسندهٔ مسئول

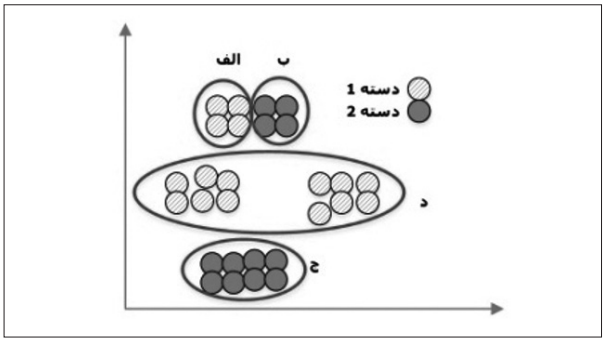
1- Supervised Adaptive Incremental Clustering
2- Improved Supervised Adaptive Incremental Clustering
3- Data stream of chunks
4- Data stream
5- Clustering
6- Supervised clustering

با توجه به خصوصیات جریان داده روش‌های خوشه‌بندی باید به صورت افزایشی باشند زیرا تمامی داده‌ها به یکباره در دسترس نیستند[۷].

فرآیند خوشه‌بندی نظارت شده سعی دارد که یک مجموعه داده را به چندین خوشه تقسیم نماید به‌طوری که داده‌های قرار گرفته در یک خوشه دارای یک برچسب باشند و تعداد خوشه‌ها کمینه باشد. در حال حاضر روش‌های متعددی برای خوشه‌بندی داده‌ها وجود دارد که بر اساس نوع داده‌ها، شکل خوشه‌ها، فاصله داده‌ها و غیره عمل خوشه‌بندی را انجام میدهند [۱۳,۱٤].

در میان الگوریتم‌های خوشه‌بندی جریان داده روش SAIC، یک الگوریتم تقریبی است که در سال ۲۰۱۷ توسط Zheng و دیگران ارائه شده است. الگوریتم SAIC به خوشه‌بندی توده‌های جریان داده می‌پردازد. در این الگوریتم فرض می‌شود که در هر لحظه یک توده از داده‌ها وارد می‌شود. تفاوت توده‌های جریان داده با جریان داده این است که وقتی با توده‌های جریان داده کار می‌کنیم درواقع با ترکیبی از جریان داده و داده‌های ایستا روبرو هستیم و در واقع با هر توده می‌توان مشابه داده‌های ایستا برخورد کرد. بخصوص وقتی با داده‌های حجیم روبرو هستیم گاهی برای پردازش این داده‌ها آن‌ها را به توده‌های جریان داده تبدیل می‌کنیم. الگوریتم SAIC مبتنی بر یک معیار فاصله است و هر یک از فاصله‌های همیلتونی، اقلیدسی و یا فاصله ماهالانوبیس قابل استفاده هستند. این الگوریتم تصادفی بر مبنای مقایسه فاصله بین خوشه‌ها است و نه تنها فاصلهٔ یک داده از خوشه‌های با برچسب مشابه در تعیین موقعیت آن داده موثر است بلکه فاصلهٔ داده با خوشه‌های با برچسب مخالف نیز تاثیرگذار است.

عدم کارایی این الگوریتم زمانی است که تمام داده‌های یک توده از جریان داده یک برچسب داشته باشند و در واقع به یک دسته تعلق داشته باشند و مجموعه خوشه‌هایی که تا آن زمان در این جریان داده ایجاد



شکل ۱: نمونه‌ای از خوشه‌بندی نظارت شده.

شده‌اند در فاصله زیادی از این توده‌های جریان داده باشند. در این حالت بدون توجه به شکل داده‌های این توده، تنها یک خوشه در مرکز تودهٔ جدید توسط الگوریتم SAIC معرفی می‌شود.

در این مقاله یک نسخه بهبود یافته از الگوریتم SAIC برای حل مسئلهٔ خوشه‌بندی ارائه می‌شود که نواقص الگوریتم SAIC را برطرف می‌نماید. الگوریتم SAIC یک الگوریتم تصادفی است که بر مبنای مقایسه خوشه‌ها عمل می‌کند و این امر بخصوص وقتی با توده جریان دادهٔ تک دسته‌ای روبرو می‌شویم ممکن است مشکل ایجاد کند. لازم به ذکر است پیچیدگی محاسباتی و زمان اجرای الگوریتم بهبود یافته ISAIC تفاوت قابل ملاحظه‌ای نسبت به الگوریتم SAIC ندارد.

ساختار مقاله در ادامه شرح داده می‌شود. در بخش ۲ کارهای مرتبط مطرح می‌شود. در بخش ۳ الگوریتم بهبود یافته ISAIC معرفی می‌شود. بخش ٤ آزمایش‌ها را شرح می‌دهد و بخش ٥ شامل جمع‌بندی مقاله است.

۲- کارهای مرتبط

در این بخش ابتدا تعدادی از الگوریتم‌های خوشه‌بندی مطرح برای مجموعه داده‌های ایستا به صورت خلاصه شرح داده می‌شوند و سپس به خوشه‌بندی جریان داده پرداخته می‌شود. الگوریتم‌های خوشه‌بندی نظارت شده، همان‌طور که شکل ۱ نشان می‌دهد، یک مجموعه داده را به چندین خوشه طوری تقسیم می نمایند که دادههای عضو یک خوشه دارای برچسب یکسان باشند.

– الگوریتم‌های چگالی محور

در این الگوریتم‌ها فرض می‌شود که خوشه‌ها مناطقی با چگالی بیشتر هستند که توسط مناطق با چگالی کمتر از هم جدا شده‌اند. یکی از مطرح‌ترین الگوریتم‌ها در این زمینه الگوریتم DBSCAN[۷] است[۲]. روش این الگوریتم به این صورت است که هر داده متعلق به یک خوشه، در دسترس چگالی برای سایر داده‌های همان خوشه است، ولی در دسترس چگالی سایر داده‌های خوشه‌های دیگر نیست (چگالی داده در همسایگی به مرکز داده و شعاع همسایگی دلخواه ε در نظر گرفته می‌شود). مزیت این روش این است که تعداد خوشه‌ها به صورت خودکار مشخص می‌شود. از نقاط قوت این الگوریتم، تشخیص نوفه و داده‌های پرت است. مشکل این روش پیچیدگی زمانی بالای آن است.

– الگوریتم k–میانگین

این روش در عین سادگی یک روش بسیار کاربردی و پایه چند روش دیگر مثل خوشه‌بندی فازی می‌باشد. در این الگوریتم ابتدا به تعداد دلخواه نقطه به‌عنوان مرکز خوشه در نظر گرفته می‌شود. سپس با بررسی هر داده، آن داده به نزدیک‌ترین مرکز خوشه نسبت داده می‌شود. پس از اتمام این کار با گرفتن میانگین در هر خوشه، مراکز خوشه‌ها و به دنبال آن خوشه‌ها اصلاح می‌شود. از جمله مشکلات این روش این است که بهینگی آن وابسته به انتخاب اولیه مراکز است. مشکل دیگر الگوریتم k-میانگین این است که تعداد خوشه‌ها به‌عنوان ورودی توسط کاربر باید تنظیم شود[۲].

– خوشه‌بندی سلسله مراتبی

خوشه‌بندی سلسله مراتبی خود شامل دو نوع روش خوشه‌بندی است:

در روش‌های خوشه‌بندی پایین به بالا[۸] ابتدا هر داده به‌عنوان یک خوشه در نظر گرفته می‌شود. سپس در ادامه با به‌کارگیری یک الگوریتم هر بار خوشه‌های دارای ویژگی‌های نزدیک به یکدیگر با هم ادغام شده و این کار ادامه می‌یابد تا چند خوشهٔ مجزا حاصل شود. مشکل این روش‌ها حساس بودن به نوفه و پیچیدگی فضایی بالا است.

در روش‌های خوشه‌بندی بالا به پایین[۹] ابتدا تمام داده‌ها به‌عنوان یک خوشه در نظر گرفته شده و با به‌کارگیری یک الگوریتم تکرار شونده هر بار این خوشه با در نظر گرفتن یک معیار شباهت داده‌ها، به خوشه‌های مجزا تقسیم می‌شوند. این کار ادامه می‌یابد تا یک یا چند خوشه یک عضوی ایجاد شود[۲] .

– الگوریتم‌های خوشه‌بندی جریان داده

الگوریتم‌های خوشه‌بندی جریان داده توسط [٤] به دو دستهٔ یک فازی و دو فازی تقسیم می‌شوند. الگوریتم‌های خوشه‌بندی یک فازی، خوشه‌بندی جریان داده را یک نسخه پیوسته از خوشه‌بندی داده‌های ایستا در نظر می‌گیرند. محدودیتی که این روش‌ها دارند این است که وزن‌های یکسانی را برای داده‌های قدیمی و جدید درنظر می‌گیرند. الگوریتم STREAM[۱۰] و الگوریتم STREAM-E[۱۱] [۱۰] نمونه‌هایی از الگوریتم‌های یک فازی هستند. در ادامه الگوریتم STREAM به‌طور خلاصه شرح داده می‌شود. در الگوریتم STREAM خوشه‌ها بر اساس تمام داده‌هایی که تا زمان جاری دریافت شده‌اند، محاسبه می‌شوند. به عبارت دیگر در این الگوریتم با استفاده از روش تقسیم و حل[۱۲] جریان‌های داده به جزءهایی تقسیم و خوشه‌های موجود در آن بر اساس الگوریتم k–میانگین محاسبه می‌شوند [۸].

الگوریتم‌های دوفازی از فاز برخط و فاز برون خط تشکیل می‌شوند. در فاز برخط که به صورت پیوسته در حال انجام است، اطلاعات آماری از داده‌ها گردآوری می‌شود. در واقع در این مرحله نوعی پیش‌پردازش بر روی داده‌ها انجام می‌شود. در فاز برون‌خط با توجه به این اطلاعات پیش‌پردازش شده، نتیجه نهایی خوشه‌بندی به‌دست

7- Density-based spatial clustering of applications with noise
8- Bottom-Up
9- Top-Down
10- Stream clustering
11- Evolution-based technique for stream clustering
12- Divide and conquer

می‌آید[۹]. الگوریتم CluStream[۱۳]، الگوریتم HPStream[۱۴]، الگوریتم DenStream[۱۵] و الگوریتم D-Stream[۱۶] نمونه‌هایی از الگوریتم‌های دو فازی هستند.

– الگوریتم SAIC

الگوریتم SAIC، الگوریتمی یک فازی است که برای خوشه‌بندی توده‌های جریان داده به‌کار می‌رود. الگوریتم SAIC داده‌ها را در هر توده به طورجداگانه خوشه‌بندی می‌کند ولی در زمان خوشه‌بندی هر توده، خوشه‌های ایجادشده در توده‌های قبلی را نادیده نمی‌گیرد، گاهی این خوشه‌ها بروز می‌شوند و گاهی خوشه جدید ایجاد می‌شود. اطلاعات هر خوشه علاوه بر برچسب آن خوشه، شامل تعداد اعضای آن خوشه نیز هست. الگوریتم SAIC خوشه‌بندی را بر اساس مقایسه برچسب داده‌ها و فاصلهٔ بین خوشه‌ها انجام می‌دهد.

تعداد اعضای خوشه‌ها دو کاربرد دارد. اولین کاربرد آن، حذف نقاط دورافتاده و نوفه است. اگر در پایان خوشه‌بندی یک توده از جریان داده، خوشه ای تعداد اعضای کمی داشته باشد (از حد آستانه کمتر باشد) آن خوشه حذف می‌شود.

دومین کاربرد آن، بروزرسانی مرکز خوشه‌ها است. در زمان بروزرسانی یک خوشه، وقتی داده‌ای وارد یک خوشه می‌شود، مرکز خوشه تغییر می‌کند و از طرفی در این الگوریتم تنها مرکز خوشه نگهداری می‌شود.

در الگوریتم SAIC داده‌ها یکی یکی بررسی می‌شوند و در نهایت یا یک خوشه جدید ایجاد می‌شود یا مرکز یک خوشه بروزرسانی می‌شود. اگر برچسب داده جدید باشد یا اگر کمترین فاصلهٔ این داده با خوشه‌های با برچسب مشابه بیشتر از کمترین فاصلهٔ این داده با خوشه‌های با برچسب مخالف خود باشد، خوشه جدید ایجاد می‌شود.

دلیل این‌که این الگوریتم قادر به تشخیص خوشه‌ها با اشکال خاص می‌باشد به نوع بررسی آن برمی‌گردد که هم فاصله با خوشه‌های با برچسب مشابه و هم فاصله با خوشه‌های با برچسب مخالف را درنظر می‌گیرد.

به علت آن‌که داده‌ها تک تک وارد و بررسی می‌شوند و مبنای این الگوریتم مقایسه است، ترتیب نقاط ورودی در نتیجهٔ خوشه‌بندی تاثیرگذار است. به منظور کاهش این تاثیر، الگوریتم داده‌های توده را به هم ریخته و کار خوشه‌بندی را آنقدر تکرار می‌کند که در دو مرحله متوالی تعداد خوشه‌ها تغییر نکند[۱] .

۳- روش پیشنهادی

مشکل الگوریتم SAIC زمانی است که در یک توده ازجریان داده، داده‌های با برچسب یکسان به هم نزدیک و از داده‌های با برچسب مخالف دور باشند و مجموعه خوشه‌هایی که تا آن زمان در این جریان داده ایجاد شده‌اند در فاصله زیادی از این داده‌ها قرار گرفته باشند. در واقع این الگوریتم بر مبنای مقایسه فاصله هر داده ورودی با خوشه‌های موجود کار می‌کند. خوشه‌بندی داده‌های درون یک توده مشابه روش k-میانگین است با این تفاوت که به تعیین تعداد خوشه‌های اولیه نیاز ندارد؛ چون داده‌ها دارای برچسب می‌باشند الگوریتم بر اساس مقایسه داده جاری با خوشه‌های قبلی، خوشه جدید را در صورت لزوم تولید می‌کند. در روش SAIC فرض شده است که آنقدر تنوع برچسب (و در نتیجه دسته) در هر توده داده موجود است که بتوانیم خوشه‌بندی را بدون نیاز به فرض خوشه‌های اولیه ایجاد کنیم. اما ممکن است در عمل یک توده داده شامل داده‌هایی با تنوع کافی برچسب نباشد یا پراکندگی داده‌ها به گونه‌ای باشد که تنوع برچسب تاثیری در خوشه‌بندی توده داده نداشته باشد. در این شرایط تعداد خوشه‌های حاصل در هر توده برابر با تعداد برچسب‌های موجود در آن توده خواهد بود و خوشه‌بندی قادر به تشخیص شکل داده نیست.

در این مقاله جهت حل این مشکل در الگوریتم پیشنهادی ISAIC برای خوشه‌بندی هر توده، تعدادی خوشه اولیه

13- a method of clustering data streams, based on the concept of microclusters
14- High dimensional projected data stream clustering
15- Density based stream clustering
16- clustering stream data using density

درنظـر می‌گیریم. به این منظور پارامتـر k در نظر گرفته می‌شـود که تعداد خوشـه‌های اولیه را مشخص می‌کند. الگوریتم پیشـنهادی، هر توده را چندین بار خوشـه‌بندی می‌کند تـا این‌که از بی‌اثـر بودن ترتیـب داده‌ها اطمینان حاصل شود. بنابراین با انتخاب k دادهٔ اول به‌عنوان مراکز خوشه‌های اولیه، با تنوع خوبی از خوشه‌های اولیه روبرو می‌شـویم. از طرفی ایـن الگوریتم زمان اجـرا را افزایش نمی‌دهـد. در الگوریتم SAIC تا وقتی که در دو دور متوالی خوشه‌بندی، تغییر ایجاد نشـود، الگوریتم ادامه می‌یابد و خوشه‌بندی مجدد انجام می‌شود. در هر دور این الگوریتم تعداد خوشه‌ها افزایش می‌یابد.

در الگوریتم ISAIC با در نظر گرفتن یک تعداد خوشـه اولیـه، تعداد دورهـای اجرای الگوریتـم کاهش می‌یابد و الگوریتم سریعتر همگرا می‌شود.

انتخاب مقدار متغیر k به عوامل ذیل وابسته است:

• تعداد برچسب‌های مشاهده شـده در یک توده: به ازای هر برچسـب حداقل یک خوشه تشـکیل می‌شود. هر چه تعداد برچسـب‌های درون یک توده بیشـتر باشد، حداقل تعداد خوشه‌هایی که آن توده می‌تواند داشته باشد افزایش می‌یابد. بنابراین اگر تنوع برچسـب کمتر باشـد الگوریتم SAIC تعداد خوشه‌های کمتری را شناسایی می‌کند و ممکن است در تشخیص شکل خوشه و در نهایت خوشه‌بندی با مشکل مواجه شود.

• شـماره توده: چون بـا ورود هر توده تعدادی خوشـه تولید می‌شود که در روند خوشه‌بندی توده بعدی استفاده می‌شـود، انتظـار داریم با افزایش شـماره تـوده، تعداد خوشه‌ها روندی افزایش داشته باشد و در نتیجه به مقدار کمتری برای متغیر k نیاز باشد.

• پراکندگـی داده‌های درون یک توده: با افزایش پراکندگی داده‌های درون یک توده نیاز به k با مقدار بیشـتر اسـت. به‌عنوان مثال برای خوشـه‌بندی توده‌ای به شکل دایره به شعاع ۱۰ در مقایسه با توده‌ای به شکل دایره و شعاع ۱ به تعداد خوشه‌های بیشتری نیاز است تا بتوان شکل توده را به درستی تشخیص داد.

• تعـداد داده‌های درون هر توده: هر چـه تعداد داده‌های درون یک توده کمتر باشد به k کوچک‌تری نیاز است.

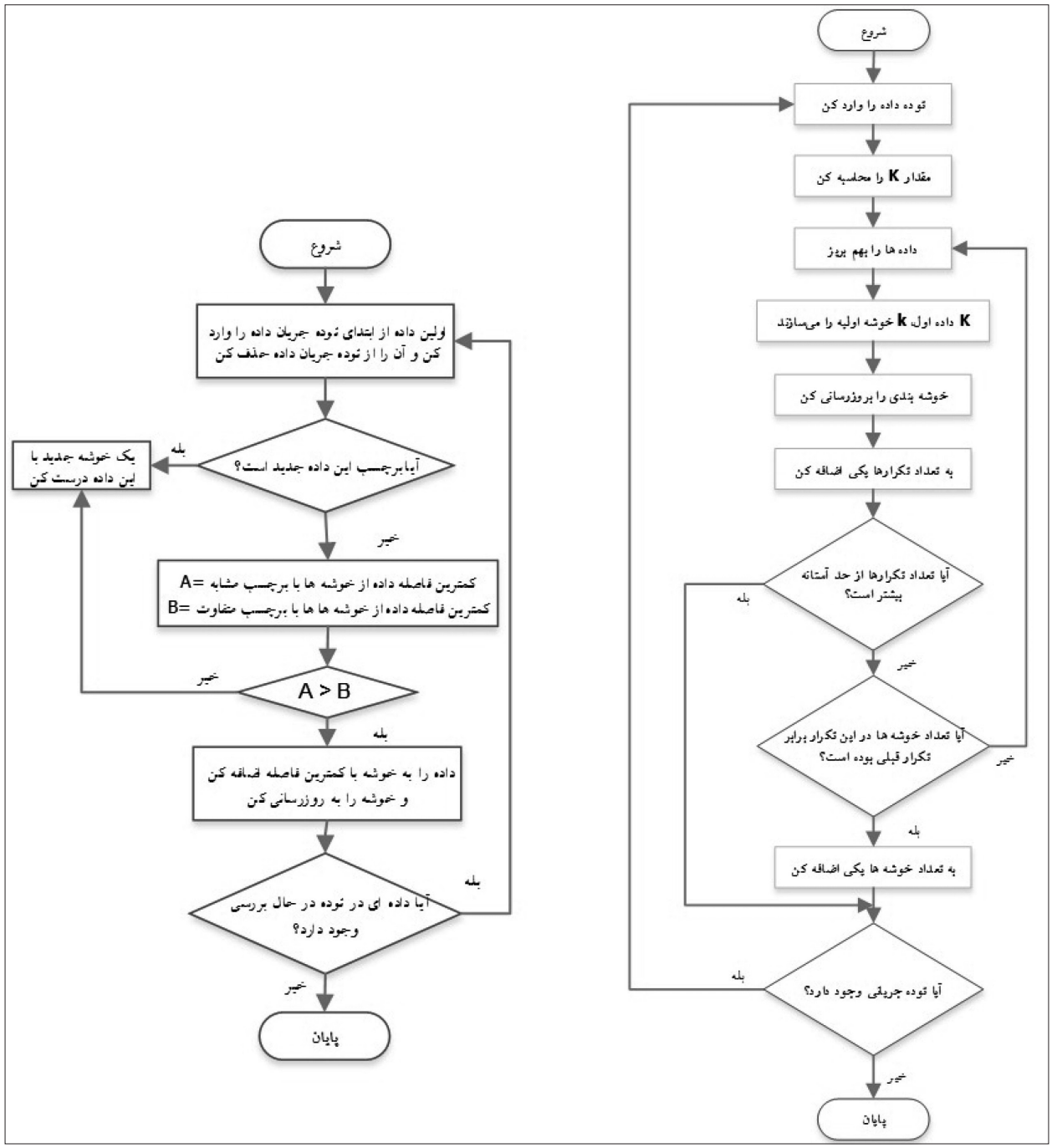
همه عوامل مذکور در تعیین مقدار k موثر هسـتند ولی برای تعییـن k تنها به آخرین عامل توجـه کرده‌ایم؛ زیرا عوامل اول و دوم معیارهای قطعی برای تعیین k نیسـتند و محاسـبه سـومین عامل متحمل هزینه زیادی در زمان اجرای الگوریتم است.

هر چه مقدار k بیشتر باشد باعث افزایش دقت می‌شود ولی از سـوی دیگر با افزایش k امکان خوشه‌بندی بیش از اندازه لازم افزایش می‌یابد. به گونه ای که در بدترین حالت هر یک از داده‌ها معرف یک خوشه می‌شوند.

در الگوریتـم ISAIC مقـدار k طبق رابطـه (۱) در نظر گرفته شده اسـت که در آن تعداد داده‌های هرتوده است. متغیر C با توجه به پراکندگی داده مقداردهی می‌شود. اگر C عددی بزرگ باشـد، باعث کاهش تعداد خوشه‌های اولیه خواهد شـد و ممکن اسـت خوشـه‌بندی به درست انجام نشـود و الگوریتم پیشنهادی مشـابه الگوریتم SAIC عمل می‌کند. اگر C مقداری کوچک داشـته باشـد باعث افزایش تعداد خوشه‌های اولیه می‌شود و اگر C برابر با یک باشد، برای هر داده یک خوشه در نظر گرفته می‌شود.

$$k = NS/C \qquad (1)$$

شـکل ۲ روند کلی الگوریتم ISAIC را نشـان می‌دهد. همان‌طور که در شـکل ۲(الف) ملاحظه می‌شود الگوریتم ISAIC توده‌های جریـان داده را به ترتیب دریافت می‌کند. بـه دلیل این‌که در گام خوشـه‌بندی داده‌هـا ترتیب ورود داده‌ها مهم اسـت، این الگوریتم داده‌هـای یک توده را به هم می‌ریزد سـپس خوشـه‌بندی می‌کند و تعداد خوشه‌ها را می‌شـمارد. درصورتـی کـه تعداد خوشـه‌ها برابر با تعداد خوشه‌های مرحله قبل باشد، عمل خوشه‌بندی توده جـاری خاتمه می‌یابد و پـس از آن الگوریتم منتظر ورود توده جدید می‌شـود. درصورتی که تعداد خوشه‌ها برابر با تعداد خوشـه‌های مرحله قبل نباشد الگوریتم بار دیگر



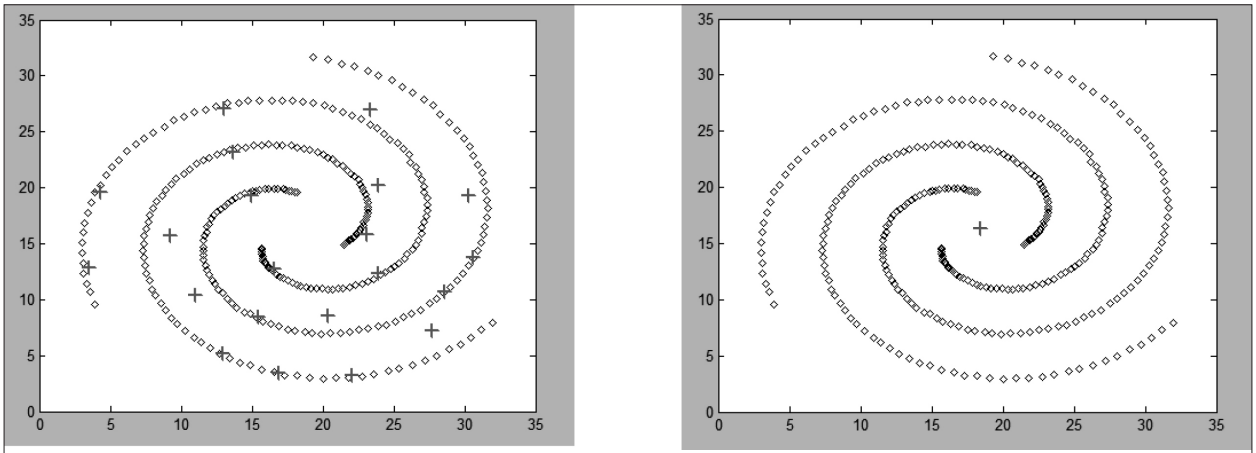
(الف) (ب)

شکل ۲: الف-روند کلی الگوریتم خوشه‌بندی پیشنهادی ISAIC. ب- خوشه‌بندی هر توده جریان داده

داده‌ها را به هم می‌ریزد و خوشه‌بندی را تکرار می‌کند. این عملیات تکرار می‌شود تا جایی که یا تعداد تکرارها برابر حد آستانه شـــود یا تعداد خوشه‌ها در دو تکرار متوالی تغییر نکند. قبل از ورود به گام خوشه‌بندی در الگوریتم ISAIC در صورتی که تعداد خوشه‌های موجود از یک حد آستانه‌ای کمتر باشد k خوشه اولیه در نظر گرفته می‌شود، تا در گام خوشه‌بندی شـــکل داده‌ها به درستی تشخیص داده شود. در شکل ۲(ب) روال خوشه‌بندی نشان داده شده است.

۴- آزمایش‌ها

بـــرای ارزیابـــی روش ISAIC و مقایســـه بـــا روش SAIC مجموعهٔ داده‌های متفاوتی به‌کار گرفته شده است.



شکل ۳: نتیجه خوشه‌بندی برای مجموعه داده Spiral۱ الف- روش SAIC ب- روش ISAIC.

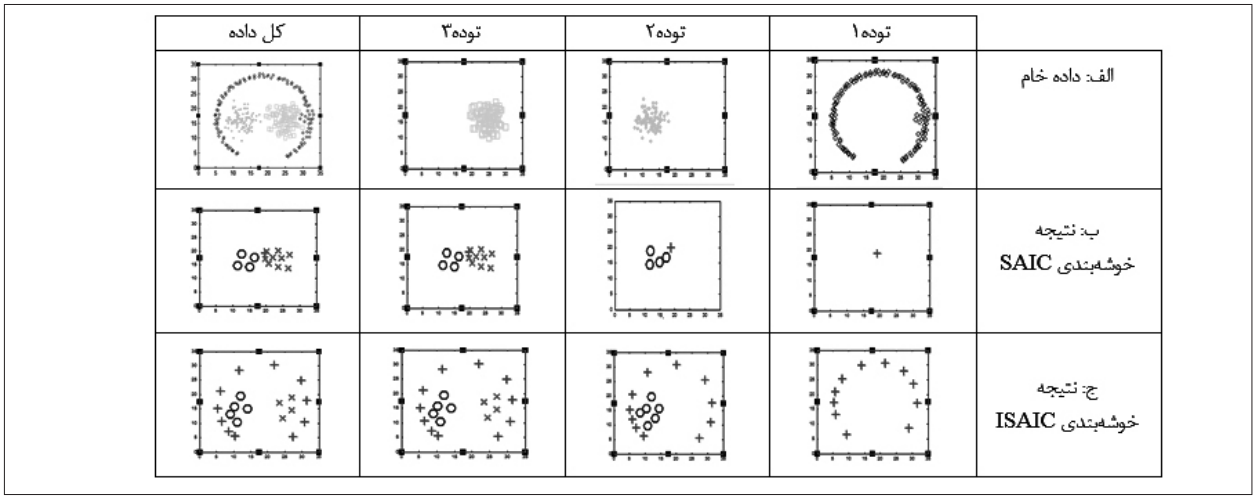
برخی از ایـــن مجموعه داده‌ها بـــا روش SAIC به خوبی خوشـــه‌بندی می‌شـــوند و برخی با این روش به درســـتی خوشه‌بندی نمی‌شوند. از سوی دیگر برخی از این مجموعه داده‌ها ایســـتا هستند و برخی دیگر جریان داده هستند. در مجموع شش مجموعه داده برای ارزیابی به‌کار گرفته شده است که دو مورد از آن‌ها ایستا و بقیه جریان داده هستند. به‌عنوان اولین مثال مجموعه دادهٔ ایستای Spiral۱ را مورد ارزیابـــی قرار می‌دهیم کـــه از روی مجموعهٔ دادهٔ Spiral [۱۳]، با تغییر برچســـب تمام داده‌ها به یک به‌دســـت آمده است. متغیر C طبق آزمایش‌های متعدد برابر با ۳۰ در نظر گرفته شده است.

نتایج آزمایش‌های در شـــکل ۳ نشان می‌دهد که برای مجموعـــه داده Spiral1 تک دســـته‌ای روش SAIC تنها یک خوشه در مرکز شکل می‌یابد (شـــکل ۳-الف) در حالی‌که روش پیشنهادی شـــکل را تشخیص می‌دهد (شکل ۳-ب). اگرچه این امر در مسئلهٔ تک‌هدفه مشکلی ایجاد نمی‌کند ولی اگر یک توده جریان داده از یک دسته باشد و از خوشه‌های شـــکل گرفته تا آن لحظه (در توده‌های قبل)، به اندازه کافی دور باشـــد ممکن اســـت در نهایت منجر بـــه خطای قابل ملاحظه‌ای گردد.

به‌عنـــوان دومین مثال، دو الگوریتـــم SAIC و ISAIC را روی جریـــان داده pathbased با درنظرگرفتن ســـه توده مطابق شکل ٤ آزمایش کرده و زمان اجرا و نحوه عملکرد این دو الگوریتم را روی این جریان داده بررســـی می‌کنیم. جریان داده مورد آزمایش دارای ســـه برچســـب است که تـــوده ۱ دارای برچســـب ۱ و توده‌هـــای ۲ و ۳ به ترتیب دارای برچسب ۲ و ۳ هستند. زمان اجرای الگوریتم SAIC روی این مجموعه داده ۰,۲۰۲۷ ثانیه اســـت. با به‌کارگیری الگوریتـــم ISAIC روی همین مجموعـــه داده زمان ۰,۲٦۰۸ ثانیه است.

در شـــکل ٤، داده‌های ســـیاه رنگ، برچسب ۱ دارند و داده‌های ســـبز و آبی رنگ به ترتیب برچســـب‌های ۲و ۳ دارند. ازطرفی خوشه‌های قرمز دارای برچسب ۱ هستند و خوشه‌های آبی و بنفش رنگ به ترتیب دارای برچسب‌های ۲ و ۳ هســـتند. عمـــده اختلاف روش‌هـــای SAIC و ISAIC در ســـتون دوم متناظر با توده اول شـــکل ٤ قابل مشاهده اســـت. هر دو الگوریتم در این مرحله داده‌ها را به درستی خوشـــه‌بندی کرده‌اند ولی روش ISAIC شـــکل توده را نیز تشخیص داده است. با ورود توده جدید، نتیجهٔ خوشه‌بندی روش SAIC درست نمی‌باشد زیرا انتظار داریم خوشه‌های قرمز رنگ (که معرف داده‌های با برچســـب ۱ هســـتند) به داده‌های ســـیاه رنگ (که دارای برچسب ۱ هستند) نسبت به خوشه‌های دیگر نزدیک‌تر باشند. همان‌طورکه در شکل ٤-ج نشـــان داده شده است روش ISAIC جریان داده را به درستی خوشه‌بندی کرده است.

ترتیب توده‌ها در جریان خوشه‌بندی با روش SAIC مهم



شکل ۴: (الف) تودهٔ جریان دادهٔ pathbased، نتیجه خوشه‌بندی الگوریتم‌های (ب) SAIC و (ج) ISAIC روی جریان داده pathbased.

است ولی این امر هیچ خللی در روش ISAIC ایجاد نمی‌کند. در واقع در تمام مسائلی که روش SAIC عملکرد خوبی دارد، روش ISAIC نیز عملکرد خوبی دارد. به این منظور این دو روش را روی جریان داده Ring-Ball2 [۱] یک مرتبه با همان ترتیب ذکر شده توسط Zheng و دیگران [۱] و مرتبه بعد این ترتیب را تغییر می‌دهیم و مورد آزمایش قرار می‌دهیم

در شکل ۵–الف داده‌های اولیه جریان داده نشان داده شده‌اند. توده‌های نشان داده شده در این قسمت به ترتیب (۱) وارد می‌شوند بعد از پردازش به وسیله الگوریتم تنها خوشه‌های ایجاد شده توسط الگوریتم باقی می‌ماند و این توده داده از بین می‌رود. درنهایت همان‌طور که مقایسه شکل‌های ۵(ب) و ۵(ج) نشان می‌دهد بعد از ورود تودهٔ اول، الگوریتم SAIC یک خوشه با برچسب ۱ در مرکز شکل ایجاد می‌کند که با توجه به این‌که این خوشه نمایندهٔ تعداد زیادی داده است با ورود توده‌های بعدی نیز حذف نمی‌شود(در سطر توده اول و ستون (ب) در شکل ۵). این درحالی است که هیچ داده‌ای با برچسب ۱ در حوالی مرکز شکل هیچگاه وارد نشده است و در این نواحی فقط داده با برچسب ۲ وارد شده است. علاوه بر این خوشه با برچسب نادرست که در الگوریتم SAIC رخ می‌دهد، چگالی خوشه‌های با برچسب ۲ نیز در حوالی این خوشهٔ نادرست زیاد است؛ در شکل ٦ این مسئله مشهود است. درحالی‌که این مشکلات همان‌طور که در شکل ۵(ج) نشان داده شده است در الگوریتم ISAIC رخ نمی‌دهد.

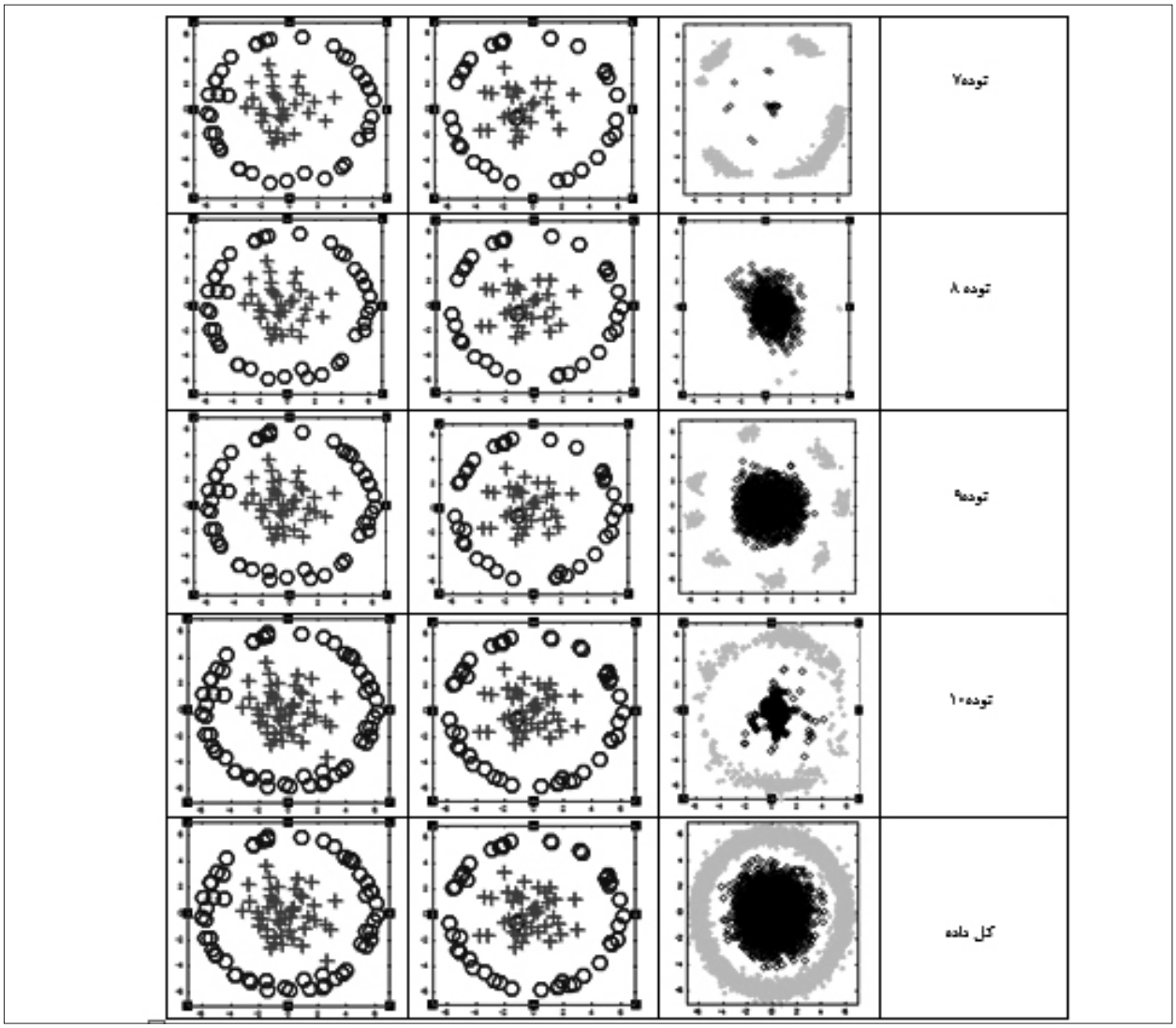
شکل ٦ نمای کلی دو روش را روی جریان دادهٔ Ring-Ball2 با ترتیب (۱) نشان می‌دهد مجموعهٔ داده‌ها و خوشه‌ها همزمان در این شکل نشان داده شده است. در شکل ٦–الف انباشتگی کلاس ۲ در نقطه‌ای نزدیک به مرکز کاملا مشهود است در این نقطه خوشه‌ای با برچسب ۱ به نادرستی قرار داده شده است. در حالی که وقتی این توده‌های جریان داده را به‌عنوان ورودی الگوریتم ISAIC قرار دادیم همان‌طور که در شکل ٦–ب مشهود است دیگر این انباشتگی نابجا رویت نشد.

با جابجایی توده اول و هفتم روی جریان داده Ring-Ball2 ترتیب (۲) را تعریف می‌کنیم. همان‌طور که گفته شد ترتیب ورود توده‌ها در الگوریتم SAIC بسیار تاثیرگذار است. این دفعه همان‌طور که در شکل ۷ می‌بینید با تغییر ترتیب توده‌ها هر دو الگوریتم نتیجهٔ خوبی می‌دهند، بنابراین این امر نشان می‌دهد که الگوریتم SAIC تحت تاثیر ترتیب ورود توده‌هاست درحالی‌که به دلیل این‌که جریان داده برخط است هیچ کنترلی بر ترتیب ورود این توده‌ها نداریم.

همان‌طور که در شکل ۷ مشاهده می‌شود روی جریان دادهٔ Ring-Ball2 با ترتیب(۲) هر دو الگوریتم خوب عمل کرده و هیچ انباشتگی یا خوشه با برچسب نادرست دیده نمی‌شود.

حال این آزمایش را روی دو مجموعه داده دیگر که

(الف) داده اولیه
(ب) نتیجه
خوشه‌بندی SAIC
(ج) نتیجه
خوشه‌بندی ISAIC
توده ۱
توده۲
توده۳
توده۴
توده۵
توده۶



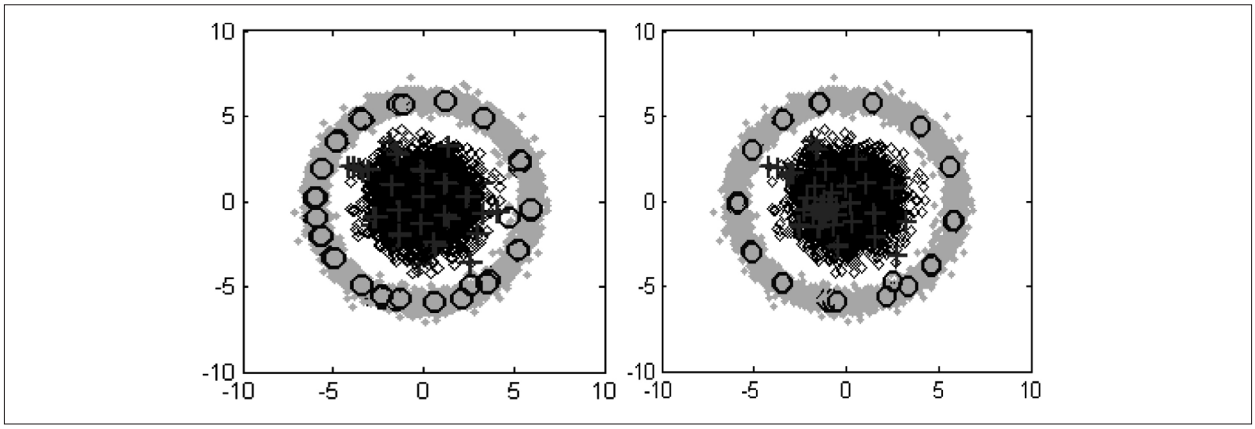
شکل ۵: توده‌های جریان دادهٔ Ring-Ball2 با ترتیب (۱)

یکی ایستا و دیگری جریان داده است امتحان می‌کنیم. برای این مجموعه داده‌ها روش SAIC خوب عمل می‌کند و نشان می‌دهیم روش ISAIC نیز خوب عمل می‌کند. مجموعه داده ایستای مورد بررسی Ringball1 [۱۱] است.

نتیجه آزمایش روی مجموعه داده ایستا Ringball1 با الگوریتم ISAIC، ٦,٣٩٤٣ ثانیه طول کشیده است و الگوریتم SAIC، ٦,٥٤٥١ ثانیه به طول انجامیده است. در شکل ۸ نتیجهٔ آزمایش نشان داده شده‌است.

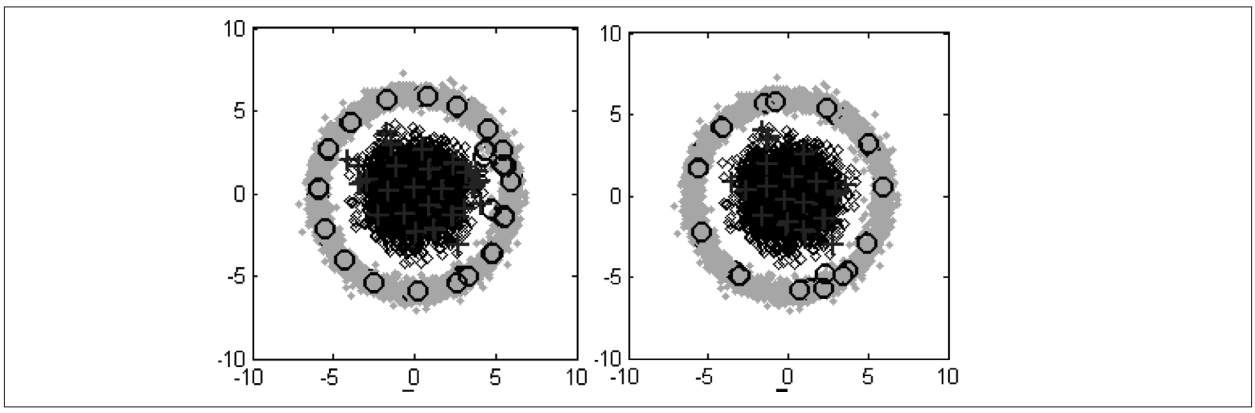
همان‌طور که در شکل ۸ مشاهده می‌شود هر دو الگوریتم نه تنها توانسته‌اند خوشه‌هایی با برچسب صحیح را ایجاد کنند بلکه شکل دادهٔ ایستای Ringball1 را نیز به درستی نشان داده‌اند. زمان اجرای دو الگوریتم نزدیک به هم بوده است. در این الگوریتم برای بی‌تاثیر کردن ترتیب نقاط توده، داده‌های توده را به هم می‌ریزیم و این کار را آنقدر تکرار می‌کنیم که تعداد خوشه‌ها در ۲ تکرار متوالی برابر یا حداقل به اندازهٔ حد آستانه تکرار صورت گرفته باشد. این امر سبب می‌شود اولا در هر بار اجرا زمان صرف شده کمی تغییر کند دوم این که زمان اجرای دو الگوریتم نزدیک به هم باشند. آنچه که مشهود است زمان اجرای هر دو الگوریتم کسری از دقیقه است که نشان از کارا بودن الگوریتم دارد.

نتیجهٔ تکرار این آزمایش روی جریان دادهٔ Spril در



(ب) (الف)

شکل ۶: الف-اجرای الگوریتم SAIC روی Ring-Ball2 با ترتیب(۱)- ب- اجرای الگوریتم ISAIC روی Ring-Ball2 با ترتیب(۱)



(ب) (الف)

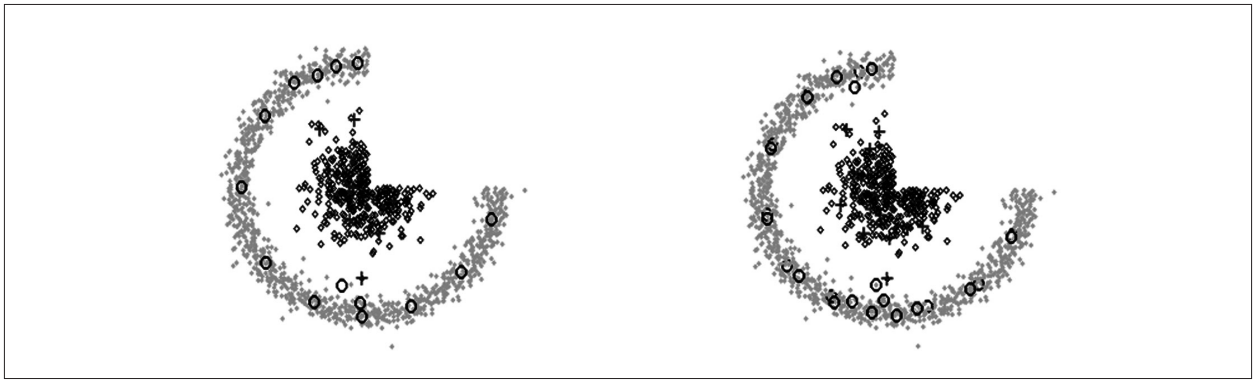
شکل ۷: الف-اجرای الگوریتم SAIC روی جریان دادهٔ Ring-Ball2 با ترتیب(۲) - ب- اجرای الگوریتم ISAIC روی جریان دادهٔ Ring-Ball2 با ترتیب(۲).

شکل ۹ نشـــان داده شده است. در قسمت ۹(الف) داده‌های اولیه نشان داده شـــده‌اند که به ترتیب نمایانگر توده اول، توده دوم، توده ســـوم و نهایتاً کل داده‌ها هست. در ۹(ب) نتیجهٔ خوشـــه‌بندی الگوریتم SAIC بعد از ورود هر یک از توده‌ها اســـت. در شـــکل ۹(ج) نتیجهٔ خوشه‌بندی الگوریتم ISAIC بعد از ورود توده‌ها به ترتیب ذکر شده هست.

همان‌طور که در شکل ۹ مشهود است هر دو الگوریتم توانســـته‌اند به خوبی این تودهٔ جریان داده را خوشه‌بندی کنند. نتیجهٔ خوشه‌بندی توســـط این دو الگوریتم در شکل تکمیلی ۱۰ نشـــان داده می‌شود. در شـــکل-الف داده‌های برچسب دار و خوشه‌های معرفی شده توسط دو الگوریتم SAIC و ISAC همزمان نشان داده شده‌اند که کمک می‌کنند بتوان نتیجهٔ خوشـــه‌بندی را بهتر ملاحظه کرد. زمان اجرا و نحوه عملکرد ایـــن دو الگوریتم را باهم روی این جریان داده بررسی می‌کنیم. زمان اجرای الگوریتم SAIC روی این مجموعه داده ۰,۲۹۹۵ ثانیه اســـت. با به‌کارگیری الگوریتم ISAIC روی همین مجموعه داده زمان ۰,۳۰۷۹ ثانیه است.

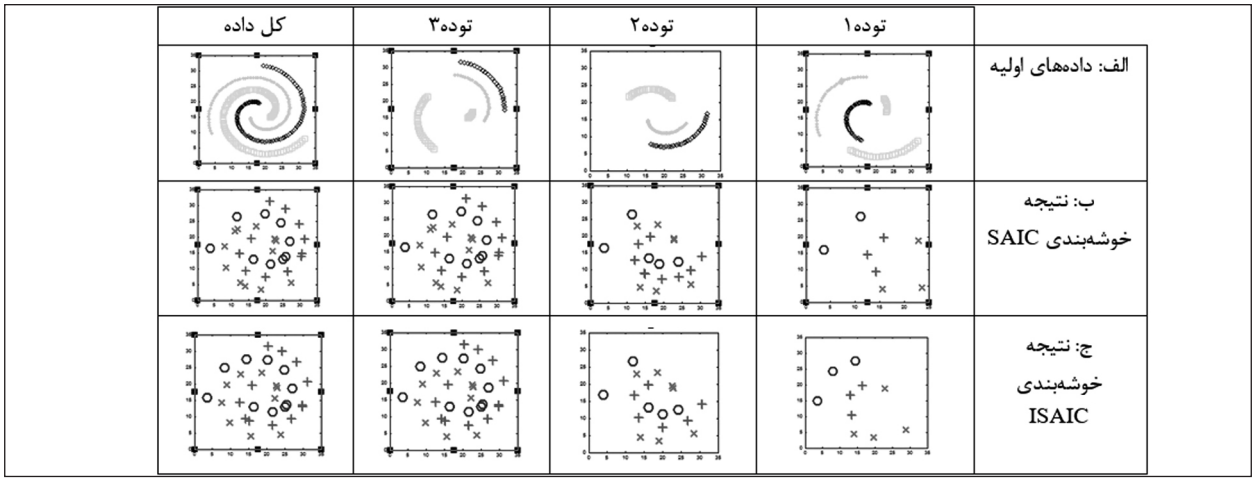
شکل ۱۰ نشـــان می‌دهد هر دو الگوریتم SAIC و ISAIC خوشـــه‌ها برچسب درســـتی دارند و شـــکل را تشخیص می‌دهند.

در آزمایشی دیگر، یک جریان داده جدید با نام circles تعریف می‌کنیم. این جریان داده شـــامل دو دایره به مرکز (۲و۲) و به شعاع‌های ۰,۸ و ۲ ، یک دایره به مرکز (۲و۹) و به شعاع ۱ و دو دایره به مرکز (۲و۱۶) و به شعاع‌های ۰,۵ و ۲ است. توده‌هایی که منجر به شکل‌گیری این جریان داده شده‌اند در شـــکل ۱۱ و در ستون دوم آن به ترتیب نشان داده شـــده‌اند. نکته قابل توجهی کـــه در این توده‌ها وجود دارد تنوع برچســـب می‌باشد و برخلاف مثال‌های قبلی که



(ب) (الف)

شکل ۸: اجرای الگوریتم‌های (الف) SAIC و (ب) ISAIC روی مجموعه داده Ringball1.

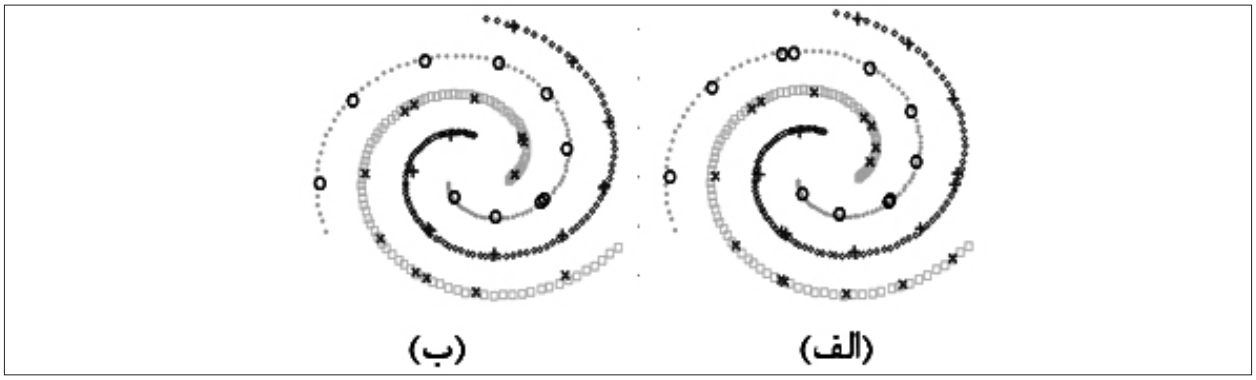


شکل ۹: توده‌های جریان داده Spiral به ترتیب از راست به چپ

مشکل خوشـــه‌بندی تنها در توده اول روی می‌داد در این مثال حتی در توده ٤ هم با این مشـــکل مواجه هستیم. در ستون دوم شکل ۱۱ این جریان داده نشان داده شده است که شـــامل ٥ توده و در کل شامل ۳ دسته است. در ستون سوم شکل ۱۱ نتیجه خوشه‌بندی این جریان داده به ترتیب با ورود این توده‌ها نشان داده شده است. آخرین سطر از این شکل در ابتدا تمام داده‌های اولیه حاصل از این جریان داده را در ســـتون دوم خود نشان میدهد. سپس در ستون سوم خوشـــه‌بندی این جریان داده با ورود هر یک از این توده‌ها نشان داده شده است.

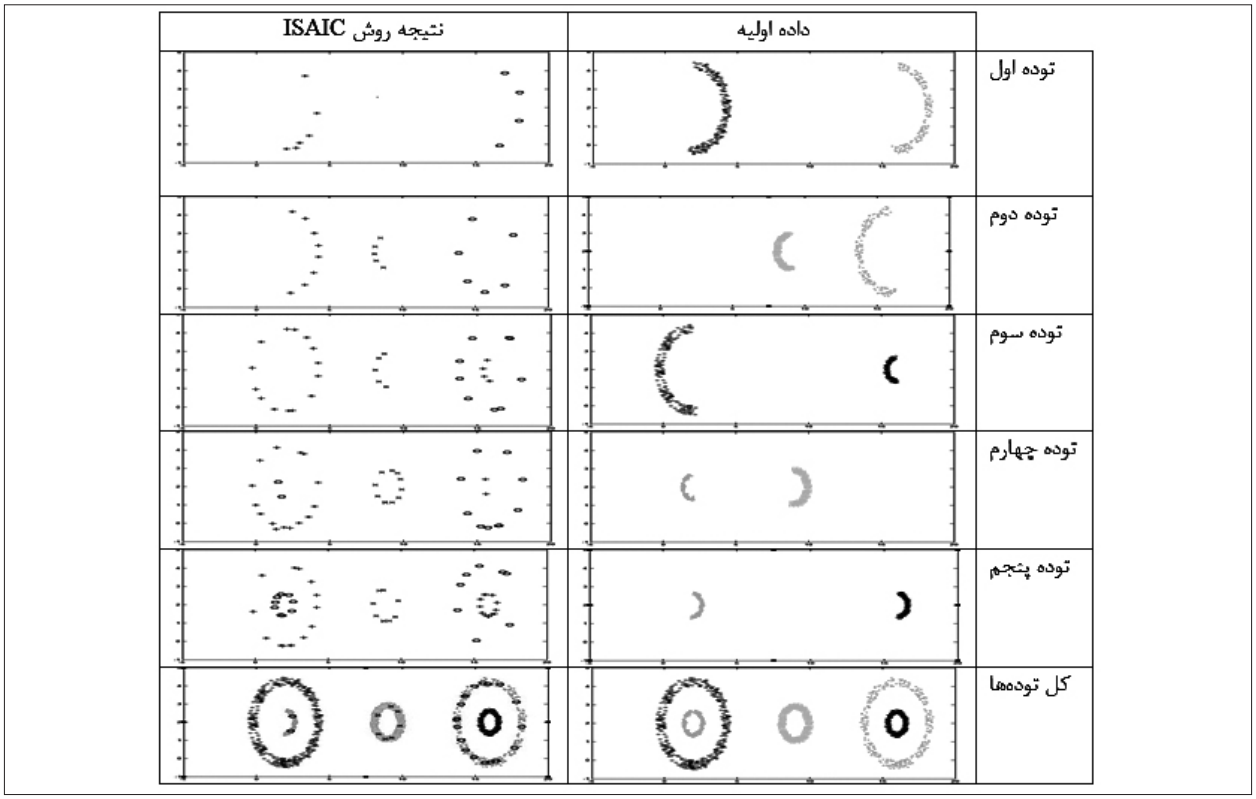
در شکل ۱۲ داده‌های اولیه و نتیجه خوشه‌بندی همزمان نشـــان داده شـــده‌اند. شـــکل ۱۲(الف) نتیجه خوشه‌بندی حاصـــل از الگوریتم SAIC را نشـــان می‌دهد. همان‌طور که در این شـــکل مشهود است. الگوریتم نتوانسته خوشه‌های مناســـب برای داده‌های اولیه سبز و مشکی رنگ با شعاع بیشـــتر را به‌درستی پیدا کند. بنابراین اشتباها به ازای این مقادیر خوشـــه‌هایی در مرکز داده‌های ســـبز و سیاه رنگ با شعاع کوچک‌تر تشخیص داده در حالی‌که این خوشه‌ها به جای نزدیکی به داده‌های با شـــعاع بزرگتر به داده‌های با شـــعاع کوچک‌تر نزدیک‌تر هســـتند. شکل ۱۲(ب) اجرای الگوریتم ISAIC را نشـــان می‌دهد. این الگوریتم علاوه بر این‌که خوشـــه‌بندی صحیحی انجام داده است شکل را هم به درستی تشخیص داده است. نکته قابل ملاحظه‌ای که در این روش‌ها وجود دارد زمان اجرای آن‌هاســـت که بسیار مناسب است و در تمام این الگوریتم‌ها مشابه هست.

جدول ۱ دقت دســـته‌بندی روی چند مجموعه داده را با به‌کارگیری الگوریتم‌های SAIC و ISAIC نشان می‌دهد. نتایج نشـــان می‌دهد که کارایی الگوریتم ISAIC مشابه یا بالاتر



(الف) (ب)

شکل ۱۰: اجرای الگوریتم‌های (الف) SAIC و (ب) ISAIC روی مجموعه داده Spiral.

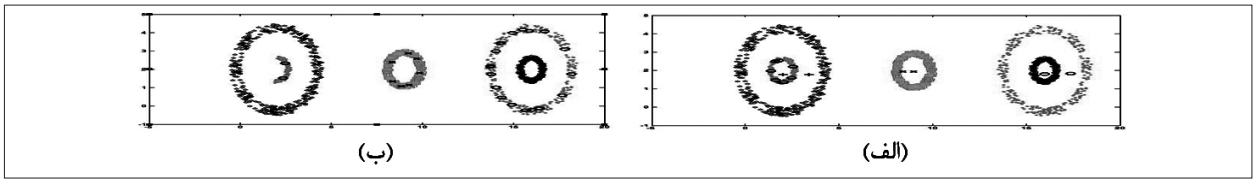


شکل ۱۱: توده‌های اولیه جریان داده circles در ستون دوم و حاصل خوشه‌بندی ISAIC در ستون سوم نشان داده شده است.

از الگوریتم SAIC است. الگوریتم ISAIC روز مجموعه داده‌های Spiral، Ring-Ball، Pathbased و Circles نتایج بهتری به‌دست آورده است. میزان بهبود دقت دسته‌بندی بر روی مجموعه داده‌های مورد آزمایش حداقل صفر و حداکثر حدود ٦٥٪ است.

۵- جمع بندی

در این مقاله ابتدا اهمیت الگوریتم‌های جریان داده و چالش‌های پیش روی این مسائل را ذکر کرده سپس به طور خلاصه الگوریتم SAIC را که به‌تازگی معرفی شده است بیان نمودیم. بعد از ذکر محاسن این روش، مشکل پیش روی این الگوریتم بیان شده و راهکاری برای حل این مشکل ارائه شد. در ادامه با ذکر چند نمونه مسئلهٔ جریان داده، برتری الگوریتم بهبود یافته نسبت به الگوریتم اولیه نشان داده شد. این الگوریتم عملکرد مناسبی دارد و نسبت به الگوریتم اولیه افزایش قابل توجهی در زمان اجرا



(ب) (الف)

شکل ۱۲: الف خوشه‌بندی بر اساس SAIC ب: خوشه‌بندی بر اساس ISAIC

ندارد. در کارهای آینده عملکرد روش پیشـــنهادی بر روی مجموعه داده‌های جریـــان داده واقعی مورد ارزیابی قرار خواهد گرفت.

جدول ۱: دقت دسته‌بندی روی چند مجموعه داده با به‌کارگیری الگوریتم‌های SAIC و ISAIC.

مجموعه داده	دقت دسته‌بندی(٪) الگوریتم ISAIC	دقت دسته‌بندی(٪) الگوریتم SAIC
Spiral1	100	100
Pathbased	97.055	61.391
$Ring\text{-}Ball_2$ (ترتیب ۱)	97.055	83.020
$Ring\text{-}Ball_2$ (ترتیب ۲)	97.701	97.360
Ring-Ball1	89.309	89.226
Spiral	81.010	79.010
Circles	98.289	23.090

منابع

[1] Laiwen Zheng, Hong Huo, Yiyou Guo, Tao Fang, Supervised Adaptive Incremental Clustering for data stream of chunks, Neurocomputing 219 (2017) 502–517.

[2] Kevin. Machine learning a probabilistic perspective. MIT Press, 2012. 875. ISBN 0262018020. Murphy

[3] Maryam Mousavi, Azuraliza Abu Bakar, and Mohammadmahdi Vakilian, Data Stream Clustering Algorithms: A Review, Int. J. Advance Soft Compu. Appl, Vol. 7, No. 3, November 2015 ISSN 2074-8523.

[4] Gaber, M. M 2012. Advances in data stream mining. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. 2(1) 79-85.

[۵] فاتحـی، کاوان؛ هادی مقدر و محمدرضا زارع میرک آباد، ۱۳۹۳، مروری بر الگوریتم‌هـا و چالش‌های موجود در خوشـه‌بندی جریان داده، دومین کنفرانس بین‌المللی دسـتاوردهای نوین در علوم مهندسـی و پایه، به‌صورت الکترونیکی، مرکز علمی کاوشگر علم،

[6]Wan, L., Ng, W. K., Dang, X. H., Yu, P. S., Zhang, K 2009. Density-based clustering of data streams at multiple resolutions. ACM Transactions on Knowledge Discovery from Data, 3(3) 1-28.

[7] Ziembiński, R. Z 2013. DeltaDens–Incremental Algorithm for On–Line Density–Based Clustering. New Trends in Databases and Information Systems.163-172.

[8] O›callaghan, L., Mishra, N., Meyerson, A., Guha, S., Motwani, R 2002. Streaming-data algorithms for high-quality clustering. Proceedings of the 18th IEEE International Conference on Data Engineering.685- 694.

[9] Gama, J 2012. A survey on learning from data streams: current and future trends. Progress in Artificial Intelligence.45-55.

[10] Udommanetanakit, K., Rakthanmanon, T., Waiyamai, K 2007. E-stream: Evolution-based technique for stream clustering. Advanced Data Mining and Applications.605-615.

[11] C.-D. Wang, J.-H. Lai, D. Huang, W.-S. Zheng, Svstream: a support vector-based algorithm for clustering data streams, IEEE Trans. Knowl. Data Eng. 25 (6) (2013) 1410–1424.

[12] Ericsson,"5G for the networked society beyond 2020," Mobile World Congress 2013, February 2013.

[۱۳] اسـماعیلی، مهـدی.(۱۳۹۲). مفاهیـم و تکنیک‌هـای داده کاوی. کاشـان: دانشگاه آزاد اسلامی.

[14] Eick, C.F., Zeidat, N., and Zhenghong, Z., "Supervised Clustering – Algorithms and Benefits", In proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligenc (ICTAI04), Boca Raton, Florida, November (2004) 774-776.