خوشه بندی سلولی جریان داده با تعدد ابعاد

تکتم دهقانی۱؛ محمود نقیب زاده۲

چکیده

در این مقاله روشی مقیاس پذیر از نظر تعدد ابعاد و اندازه مجموعه داده برای خوشه بندی روی خط جریان های داده ارائه شده است که در مقایسه با روش های پیشین علاوه بر مقیاس پذیر ی نسبت به تغییرات در اطلاعات تطابق پذیرتر و در شناسایی و تعیین خوشه ها دقیق تر و سریع تر است. در این روش در ابتدا فضای چند بعدی داده ها به سلول هایی با اندازه یکسان تقسیم می شود. در هر سلول توزیع آماری داده های اخیر که در محدوده ی آن سلول قرار دارند، ذخیره می شوند و بر اساس این اطلاعات، بدون نیاز به ذخیره سازی داده ها، خوشه بندی انجام می شود. سلول های پرتراکم به سلول های کوچکتر شکسته می شوند و این روند تا رسیدن به سلول پایه ادامه می یابد و سلول های خلوت برای کاهش حافظه ادغام می شوند. در این روش ساختاری کارا برای مدیریت سلول های در تمام ابعاد ارائه شده است، که دسترسی تصادفی و سریع به سلول ها را ممکن می سازد. ابتدا خوشه بندی یک بعدی انجام می شود، سپس خوشه ها با توجه به ارتباط بین توزیع داده ها در ابعاد مختلف، در یک روند پایین به بالا، با یکدیگر ترکیب و خوشه های نهایی تولید می شوند. با ذخیره سازی مرزهای دقیق خوشه ها در ابعاد مختلف، خوشه بندی دقیق تر انجام می شود و با اصلاح تعریف همسایگی زمان جستجو برای شناسایی همسایه های یک سلول که یکی از مشکلات اصلی خوشه بندی چندبعدی است، نیز کاهش می یابد. به منظور کاهش اثر داده های قدیمی در خوشه بندی، به اطلاعات وزنی اختصاص داده شده است و با گذشت زمان وزن آنها کاهش می یابد. در این روش خوشه بندی روی خط و تغییرات جریان در خوشه ها در نظر گرفته شده است.

کلمات کلیدی

خوشه بندی، جریان داده، شبکه ی سلولی، مقیاس پذیری، خوشه بندی با ابعاد بالا

Grid-based Clustering of High-Dimensional Data Streams

Toktam Dehghani, Mahmoud Naghibzadeh

ABSTRACT

Most data streams clustering methods lack the capability to deal with data streams of high dimensionality or large data set, in order to solve the problem, we proposed a new grid-based clustering method. In this method, the multi-dimensional data space of data stream is partitioned into equal-sized grid cells; each cell monitors the distribution statistics of data elements within its range. A dense cell is partitioned into smaller cells and such partitioning is continued until the grid cells become a unit cell. Conversely, a set of consecutive sparse cells can be merged into a single grid cell. Cells of each dimension are stored in a structure similar to B+tree. In a bottom up algorithm, in each dimension, the dense unit cells which are connected in the current dimension and, belong to the same clusters in all subspaces, are combined to generate a cluster. In this method, in order to improve the accuracy of identified clusters, the definition of neighbors of a cell is refined and a DNF expression is used for displaying boundaries of clusters. Furthermore, as time goes by, the old distribution statistics of each cell is diminished by predefined decay rate, so the results become update. The proposed method is capable of on-line clustering and adaptive to sudden changes in the stream.

KEYWORDS

Clustering, Data Streams, Grid-Based Clustering, High-Dimension Data, Scalability

---

۱. دانشجوی کارشناسی ارشد، گروه کامپیوتر، دانشگاه آزاد اسلامی واحد مشهد، t.dehghani@email.com

۲.استاد، گروه کامپیوتر، دانشکده مهندسی، دانشگاه فردوسی مشهد، naghibzadeh@ferdowsi.um.ac.ir

۱. مقدمه

در سال های اخیر، با پیشرفت تکنولوژی سخت افزاری امکان جمع آوری متوالی داده ها فراهم شده و سبب ظهور کاربردهای مانند پردازش و داده کاوی جریان کلیک های مشتری، مجموعه تراکنش های روی خط زنجیره های فروش، جریان جستجوهای کاربران، ثبت وقایع وب سایت ها، جریان پاکت های روترها، داده های سنسورها، داده های چند رسانه ای، داده های بازار سهام و پیش بینی روند مالی، داده های ترافیک شبکه و نظارت بر شبکه و... شده است. با توجه به اینکه در این کاربردها بیشتر از آنکه پردازش مجموعه داده ای بزرگ و ذخیره شده در حافظه نیاز باشد، داده کاوی جریان های سریع و گذرا لازم است، از این رو روش های مختلفی برای داده کاوی جریان داده ها مطرح شده اند.

جریان داده[1] یک توالی نامحدود و حجیم از عناصر داده ای است که متوالیاً با سرعت زیاد تولید می شوند. [۱] به دلیل ویژگی های ذکر شده امکان ذخیره سازی تمام داده های جریان وجود ندارد، در نتیجه جریان های داده باید بصورت روی خط پردازش شوند. در روش های پردازش جریان های داده لازم است به سه نکته توجه شود: [۲]

۱. هر داده ورودی باید حداکثر یکبار آنالیز شود.

۲. با وجود تولید ادامه دار داده ها ، باید حافظه محدودی برای پردازش جریان های داده در نظر گرفته شود.

۳. داده های جدید باید با حداکثر سرعت پردازش شوند و آنالیزهای آنها برای بروزرسانی نتایج استفاده شود، در نتیجه خروجی دقیق و بروز در هر لحظه ارائه گردد.

خوشه بندی داده ها یکی از شاخه های اصلی داده کاوی است. در این فرآیند داده ها به گروه های معنا دار به نام خوشه تقسیم می شوند. خوشه بندی به نحوی صورت می گیرد که بیشترین شباهت بین داده های درون یک خوشه و کمترین شباهت بین داده های خوشه های مختلف وجود داشته باشد. [۳]

یکی از روش های خوشه بندی ارائه شده برای جریان داده ها CS tree می باشد. این روش از نوع خوشه بندی توری است و فضای داده های جریان را به مجموعه ای از سلول ها تقسیم می کند. با متراکم شدن یک سلول، آن سلول به سلول های کوچکتر شکسته می شود و این روند تا رسیدن به سلول های واحد ادامه می یابد. سپس در هر بعد سلول های همسایه خوشه های یک بعدی می سازند و خوشه های ابعاد مختلف ترکیب و خوشه های چند بعدی ساخته می شوند. روش CS tree خوشه های با اشکال مختلف را در فضای چند بعدی به طور تقریبی شناسایی می کند و نسبت به روش های خوشه بندی دیگر سریع تر است و نویز و نقاط پرت را بهتر مدیریت می کند. در این روش مرز خوشه های یک بعدی به دقت ذخیره می شود اما در زمان ترکیب خوشه ها مرز خوشه های چند بعدی دقیق نمی باشد و با افزایش ابعاد و تغییر شکل خوشه ها با ورود داده های جدید به سرعت دقت خوشه بندی کاهش می یابد.

در روش پیشنهادی در این مقاله در ابتدا ساختاری مشابه B^+tree برای ذخیره اطلاعات سلول ها در هر بعد ارائه شده است که امکان دسترسی سریع و تصادفی به سلول ها را فراهم می کند. همچنین با اصلاح مفهوم همسایگی و ارائه ساختاری مشابه DNF برای ذخیره سازی خوشه ها، مرز یک خوشه در تمام ابعاد را با دقت بیشتری حفظ می شود. به منظور اصلاح خوشه بندی چند بعدی، روش جدیدی برای ترکیب خوشه های یک بعدی و ایجاد خوشه های چند بعدی ارائه شده است. در این روش، ترکیب بر اساس ارتباط توزیع داده بین بعدهای مختلف خوشه بندی صورت می گیرد و در نتیجه دقت خوشه ها با افزایش ابعاد کاهش نمی یابد و همچنین با ورود داده های جدید و تغییر شکل خوشه ها می توان مرزهای دقیق خوشه های جدید را محاسبه کرد.

در این مقاله، ابتدا روش های خوشه‌بندی جریان داده را بررسی می شود، سپس در بخش ۳ خوشه بندی توری جریان داده و در بخش ۴ الگوریتم CS tree مورد بحث و بررسی قرار می‌گیرد، در بخش ۵ به ارزیابی روش CS tree و اشکالات آن پرداخته می شود. در بخش ۶ الگوریتم پیشنهادی برای بهبود روش قبلی بیان می شود و تاثیر این روش در ارائه الگوریتمی مقیاس پذیر برای خوشه بندی جریان داده ای با ابعاد بالا مطرح می شود. در بخش ۷ نتیجه گیری نهایی ارائه شده است .

۲. مروری بر روش های خوشه بندی جریان داده

الگوریتم های مختلفی برای خوشه بندی ارائه شده اند که عموماً اندازه مجموعه داده را ثابت فرض کرده و داده ها را چندین بار پیمایش می کنند . در نتیجه این الگوریتم ها مناسب مجموعه داده های بزرگ و جریان داده نمی باشند. تعداد معدودی الگوریتم برای خوشه بندی جریان داده ها ارائه شده اند و مهمترین مساله در آنها چگونگی استفاده از خوشه های شناسایی شده مربوط به داده ها قبلی در یافتن خوشه های بروز و مربوط به داده اخیر، با توجه به محدودیت در حافظه و سرعت ورود داده ها است.[۳] در ادامه به بررسی اجمالی روش های ارائه شده برای خوشه بندی جریان داده ها می پردازیم .

در روش ارائه شده در[۴]، از تکنیک K میانگین برای خوشه بندی جریان داده ها استفاده شده است. این الگوریتم از خوشه بندی یک نمونه با اندازه مشخص ۲K خوشه شروع می کند، سپس با پر شدن حافظه، در لایه دوم خوشه های بدست آمده از نمونه ها به ۲k خوشه دسته بندی می شوند. این فرایند در چندین لایه تکرار می شود و در آخر ۲k خوشه به k خوشه تبدیل می شوند. از اشکالات این روش می توان به افزایش فاکتور تقریبی با افزایش لایه ها و نادیده گرفتن تاثیر گذشت زمان در ساختار خوشه ها ی ذخیره شده اشاره کرد.

در روش ارائه شده در [۵]، دو الگوریتم STREAM و LOCALSEARCH برای خوشه بندی جریان داده تعریف شده اند. الگوریتم STREAM اندازه نمونه را مشخص می کند و در LOCALSEARCH، k داده از نمونه ها به عنوان مراکز محلی انتخاب می شوند. اگر تعداد مراکز بیشتر از حافظه از پیش تعیین شده شود آنگاه LOCALSEARCH بر روی تمام داده هایی که تاکنون تولید شده اند، مجدداً اعمال و K مرکز جدید ایجاد می شود. در این روش اگر تعداد خوشه ها از ابتدا مشخص نباشد ، الگوریتم تا رسیدن به خوشه های با کیفیت مناسب ادامه می یابد که سبب افزایش هزینه محاسباتی می شود.

در روش مطرح شده در[۶] ، یک چارچوب برای خوشه بندی جریان داده با نام Clustream ارائه شده است. دراین روش فرایند خوشه بندی به دو کامپوننت تقسیم می شود. کامپوننت روی خط آماره های مربوط به داده ها را ذخیره می کند و از روش k میانگین برای پیدا کردن خوشه های اولیه که به آنها میکروخوشه گفته می شود، استفاده می کند. اطلاعات هرخوشه در بردار ویژگی آن خوشه ذخیره می شود و با ورود داده های جدید بردار ویژگی خوشه ها بروز می شود. بردارهای ویژگی در زمان های مشخصی در حافظه ذخیره می شوند. کامپوننت برون خطی با اعمال الگوریتم K میانگین بر روی میکرو خوشه ها در بازه مشخص شده توسط کاربر ماکرو خوشه ها را ایجاد می کند. همچنین برای خوشه بندی جریان داده هایی با ابعاد زیاد HPStream ارائه شده است که ابعاد موثر در خوشه بندی را شناسایی می کند و با روشی مشابه Clustream خوشه بندی را بر روی ابعاد موثر انجام می دهد.

از دیگر از روشهای ارائه شده برای خوشه بندی جریان داده ها می توان به cell tree و CS tree اشاره کرد، که برخلاف روشهای پیشین که خوشه بندی تقسیمی[۲] و سلسله مراتبی استفاده می گردد، از خوشه بندی توری استفاده می کنند. روش CS tree اصلاح شده روش cell tree می باشد و نسبت به روش قبلی به ازای ابعاد بیش تری مقیاس پذیر است. در ادامه در ابتدا به بررسی خوشه بندی توری می پردازیم و سپس روند الگوریتم خوشه بندی CS tree را مرور می کنیم.

۳. خوشه بندی توری جریان داده[۳]

در خوشه بندی توری فضای داده به مجموعه محدود سلول تقسیم و یک شبکه سلولی ایجاد می شود. هر سلول حاوی اطلاعات آماری داده های متناظر با آن سلول است. [۷] در این روش ها خوشه بندی با ادغام سلول های متراکم همسایه انجام می شود و در نتیجه زمان پردازش برای خوشه بندی مستقل از تعداد داده و متناسب با تعداد سلول های شبکه توری است که این تعداد بسیار کمتر از تعداد کل داده هاست. در روش های خوشه بندی توری کیفیت خوشه ها و میزان حافظه مصرفی به اندازه سلول ها وابسته است. هرچه سلول ها کوچکتر باشند کیفیت خوشه بندی افزایش و میزان حافظه و محاسبات نیز افزایش می یابد.

در نتیجه پیشنهاد می شود از سلول هایی با اندازه متغیر و منطبق با تراکم داده ها استفاده شود. به عبارت دیگر، در نواحی پر تراکم اندازه سلول ها کوچکتر و در نواحی کم تراکم اندازه سلول ها بزرگتر در نظر گرفته شوند تا با تمرکز بر روی فضاهای متراکم که احتمال خوشه شدن آنها زیاد است کیفیت خوشه بندی افزایش یابد و همچنین با کاهش تعداد سلول ها در نواحی کم تراکم زمان محاسبات و میزان حافظه لازم کاهش یابد.

از مزایای روش خوشه بندی توری می توان به سرعت بالا پردازش، عدم وابستگی به ترتیب ورود داده ها، امکان تولید خوشه هایی با اشکال متفاوت ، شناسایی نویزها و عدم نیاز به پیش فرض درباره تعداد خوشه ها اشاره کرد. [۸]

با توجه به ماهیت جریان داده، امکان ذخیره تمام داده ها وجود ندارد و نیاز به پردازش سریع داده ها است. همچنین با گذشت زمان خوشه هایی با اشکال مختلف ایجاد می شوند که تعداد این خوشه می تواند متغیر باشد و علاوه بر این امکان بروز نویزهای تصادفی در جریان وجود دارد، در نتیجه روش خوشه بندی توری روشی مناسب برای خوشه بندی جریان داده بنظر می رسد .

در روش های خوشه بندی توری برای جریان داده ای با ابعاد بالا، برای ذخیره تمام سلول های ایجاد شده در فضای داده نیاز به حافظه زیادی می باشد و با توجه به رشد نمایی تعداد سلول ها با افزایش ابعاد و خلوت بودن بسیاری از آنها، لازم است روشی مناسب برای ذخیره اطلاعات سلول ها ارائه شود. در روش CS tree ساختاری مناسب برای ذخیره سلول ها و خوشه بندی آنها ارائه شده است که در ادامه به بررسی آن می پردازیم.

۴. مروری بر روش خوشه بندی توری CS tree[۴]

در روش CS tree [۹] جریان داده D به صورت یک فضای d بعدی $N=N_1\times\ldots\times N_d$ تعریف شده است. ابعاد به مجموعه از بازه ها با اندازه یکسان شکسته می شود و در نتیجه فضای داده جریان D به مجموعه از سلول های d بعدی تقسیم می شود. به ازای هر داده ورودی اطلاعات

آماری مربوط به توزیع داده ها در بازه های سلول بروز می شود. اطلاعات آماری هر سلول شامل میانگین داده ها، انحراف معیار و تعداد داده های قرار گرفته در بازه ها می باشد. برای کاهش تاثیر داده های قدیمی و افزایش اثر داده های اخیر، به داده ها بر اساس زمان تولید وزنی اختصاص داده می شود که در صورتی عدم ارجاع به آن بازه، در مدت زمان مشخص، وزن آن بازه بر اساس نرخ زوال[5] کاهش می یابد. تمامی اطلاعات آماری سلول ها با در نظرگرفتن نرخ زوال بروز می شوند. به منظور مدیریت اطلاعات سلول ها ، در هر بعد اطلاعات آماری در یک لیست پیوندی مرتب بر اساس ترتیب بازه ها ذخیره می شود.

در روش CS tree برای خوشه بندی داده ها ، با ورود هر داده سه گام زیر تکرار می شود:

۱. ذخیره سازی و بروزرسانی اطلاعات آماری سلول متناظر
۲. خوشه بندی یک بعدی و ایجاد خوشه های چند بعدی
۳. تخمین مرزهای خوشه های d بعدی

درگام اول اطلاعات آماری سلول مرتبط با آن داده بروز می شود. به عبارت دیگر، اطلاعات آماری بازه های متناظر با داده در هر یک از ابعاد در لیست های پیوندی متناسب با نرخ زوال مجددا محاسبه می شود. در این مرحله اگر سلولی متراکم شود و تراکم آن از حد مشخصی بیشتر باشد، آن سلول به h سلول کوچکتر شکسته می شود و اطلاعات آماری هر بازه جدید بر اساس اطلاعات ذخیره شده در سلول قبلی، مقداردهی اولیه می شود. سلول کوچک جدید در لیست های پیوندی جایگزین سلول والد می شوند. با ورود داده ها ، به تدریج در یک ناحیه متراکم ، سلول های کوچک و متراکم نیز مجددا شکسته می شوند و این روند تا رسیدن به سلول واحد ادامه می یابد. سلول واحد ، کوچکترین سلول در فضای داده تعریف شده است و ابعاد آن معادل مقدار مشخص λ است. همچنین به منظور کاهش حافظه مصرفی و حذف سلول های اضافه، که در گذشته متراکم بودند و به مرور زمان میزان ارجاع به آنها کاهش یافته است، سلول های خلوت در صورتی که h-۱ سلول همسایه ی آنها نیز خلوت باشند، با هم ادغام می شوند و یک سلول بزرگتر می سازند .

درگام دوم ، در هر بعد ، سلول های واحد که در اثر افرازهای متوالی سلول های اولیه بدست آمده اند، اگر متراکم باشند و تراکم داده در آنها از حد مشخصی بیشتر باشد، می توانند با سلول های متراکم همسایه خود تشکیل خوشه های یک بعدی دهند. به ازای هر بعد، خوشه ها در یک جدول ذخیره می شوند. در این الگوریتم خوشه بندی به صورت محلی انجام می شود و هر خوشه یک ناحیه متراکم از فضا است و خوشه ها با نواحی خلوت جدا می شوند. این خوشه ها با گذشت زمان و تغییر در تراکم سلول هایشان ممکن است با هم ادغام و یا یک خوشه به خوشه های کوچکتر شکسته شود.

در ادامه، ابعاد بر اساس ترتیبی از پیش تعیین شده پیمایش و خوشه ها یک بعدی ترکیب و خوشه های چند بعدی ایجاد می شوند. در این مرحله یک درخت با نام CS tree ساخته می شود که هر سطح آن با یک بعد از جریان داده ها متناظر است. با ورود هر داده بر اساس ترتیب ابعاد، خوشه های یک بعدی که داده در بازه آنها قرار گرفته است، بررسی می شوند. اگر به ازای آن داده در لیست پیوندی بعد اول خوشه ای ایجاد شده باشد، آن خوشه به درخت اضافه می شود، اگر در بعد بعدی نیز خوشه ای وجود داشته باشد نودی متناظر با آن خوشه و به عنوان فرزند نود قبلی به درخت اضافه می شود و این روند تا رسیدن به بعد d ام ادامه می یابد. اگر خوشه ای وجود داشته باشد و نود مربوط به آن قبلاً به درخت اضافه شده باشد، فقط اطلاعات آماری آن بروز می شود. از آنجا که پیمایش ابعاد بر اساس ترتیب مشخصی صورت می گیرد، در این روش تا زمانیکه یک سلول در ابعاد زیرین متراکم و خوشه نشده باشند، خوشه شدن آن داده در ابعاد بعدی بررسی نمی شود. در این الگوریتم، در زمان بررسی سطح به سطح CS tree نودهایی که مدتی به آنها ارجاعی صورت نگرفته است به همراه فرزندانشان از درخت حذف می شوند زیرا احتمال اینکه این خوشه ها که در گذشته متراکم بودند اما اخیرا به ازای داده های جدید بروز نشده اند، مجددا در آینده نزدیک متراکم شوند بسیار کم است.

درگام سوم، به ازای خوشه های d بعدی (d تعداد کل ابعاد داده جریان است) یک خلاصه ی آماری درباره توزیع داده ها ی خوشه ذخیره می شود. این گام به منظور مشخص کردن مرز دقیق خوشه های نهایی انجام می شود. در آخر، خوشه های d بعدی با پیمایش درخت بدست می آیند.

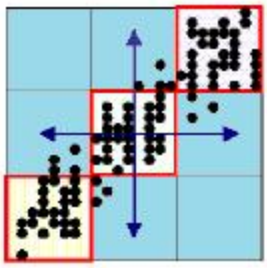
الگوریتم CS tree در مقایسه با دیگر روش های خوشه بندی برای جریان های داده، نویزهای تصادفی و نقاط پرت را بهتر مدیریت می کند، نیازی به فرض اولیه درباره تعداد خوشه ها ندارد، تعداد خوشه ها در طول جریان می تواند تغییر کند، همچنین تا حدودی می تواند خوشه هایی با اشکال متفاوت ایجاد و این اشکال در طول جریان می توانند تغییر کنند. با این وجود اشکالاتی در این الگوریتم وجود دارد که در ادامه به بررسی آنها می پردازیم .

۵. بررسی نقاط ضعف الگوریتم CS tree

همان طور که اشاره شد در روش CS tree به ازای هر بعد لیست پیوندی برای ذخیره سازی اطلاعات آماری سلول ها در نظرگرفته شده است. به ازای هر داده ورودی این لیست در تمام ابعاد پیمایش و بازه های مربوط با داده شناسایی و اطلاعات آماری ذخیره شده در آنها بروز می

شود. در این روش با این فرض که در بعد N، اندازه سلول واحد λ باشد و تمام سلول های اولیه پس از افرازهای متعدد تبدیل به سلول های واحد شده باشند، در این صورت حداکثر تعداد سلول های ذخیره شده در لیست پیوندی برابر $\frac{range(N)}{\lambda}$ خواهد بود. در نتیجه به ازای هر داده ورودی در هر بعد به طور متوسط $\frac{range(N)}{2\lambda}$ خانه از لیست پیمایش می شود که سبب کاهش سرعت پردازش جریان داده ها می شود.

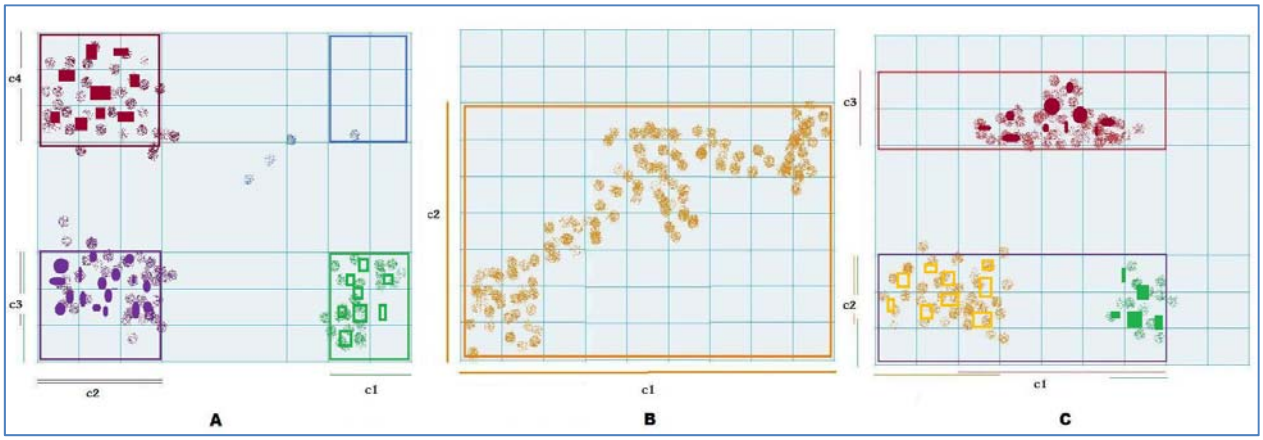
همچنین در اکثر روش های خوشه بندی توری همسایگی به این صورت تعریف شده است که دو خانه همسایه اند اگر وجه مشترک داشته باشند و یا یک سلول واسط که با هر دو همسایه است بین آنها قرار گرفته باشد، اما این تعریف از همسایگی سبب بروز خطا در خوشه بندی می شود. به عنوان مثال، زمانیکه توزیع داده ها موازی محورها نباشند احتمال تقسیم خوشه های با معنی به زیر خوشه های بی معنی وجود دارد.[۱۰] در شکل ۱ در فضای دو بعدی یک خوشه وجود دارد اما بر اساس تعریف ارائه شده از همسایگی، خروجی سه خوشه است.



شکل۱) تقسیم خوشه های با معنی به زیر خوشه های بی معنی

علاوه بر این در بسیاری از کاربرد های لازم است خوشه بندی بر روی جریان داده ای با ابعاد زیاد صورت گیرد، اما خوشه بندی چند بعدی با افزایش تعداد ابعاد پیچیده تر و زمانبرتر می شود. در روش CS tree به منظور کاهش پیچیدگی خوشه بندی در ابتدا خوشه های یک بعدی از مجموعه ای از سلول های متراکم همسایه ایجاد می شوند و سپس به ازای هر داده بر اساس ترتیب از پیش تعیین شده ابعاد بررسی و خوشه هایی که داده در هر یک از ابعاد به آنها تعلق دارد، با یکدیگر ترکیب و یک خوشه چندبعدی می سازند. به عبارت دیگر، دو داده که در بعد اول در یک خوشه و در بعد دوم در یک خوشه قرار دارند، در یک خوشه دو بعدی قرار می گیرند. با وجود اینکه این تعریف درست به نظر می رسد اما این روش خوشه بندی سبب کاهش دقت در خوشه ها می شود و در بعضی از حالات خوشه های ایجاد شده با خوشه های واقعی از نظر تعداد، شکل و مرزها متفاوت می باشند. این خطا در خوشه بندی با افزایش تعداد ابعاد، تشدید می شود. علت اصلی این مشکل ، خوشه بندی مجزای هر بعد و ترکیب خوشه های یک بعدی، بدون درنظرگرفتن ارتباط توزیع داده ها ی ابعاد مختلف است.

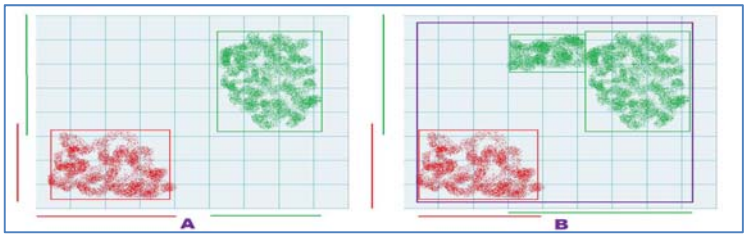
در شکل ۲ قسمت A، به ازای داده های ورودی در بعد x دو خوشه C1 و C2 و در بعد y دو خوشه C3 و C4 از تصویر نقاط روی محور ها ایجاد شده اند. در الگوریتم CS tree به ازای هر داده ورودی در هر بعد خوشه هایی که در آنها داده قرار دارد شناسایی و با ترکیب خوشه ها، خوشه چند بعدی ساخته می شود. همان طور که در شکل مشخص است در فضای دو بعدی سه خوشه وجود دارد، اما الگوریتم در اثر وجود نویز یا نقاط پرت چهار خوشه را شناسایی می کند. در شکل ۲ قسمت B، در بعد x و y دو خوشه C1 و C2 ایجاد شده اند که با ترکیب آنها خوشه ی متفاوت با شکل خوشه اصلی ایجاد می شود. در شکل ۲ قسمت C، فضای داده دو بعدی شامل سه خوشه مجزا است، با تصویر کردن نقاط بر روی محورها، در بعد x یک خوشه C1 و در بعد y دو خوشه C2 و C3 ایجاد شده اند، در نتیجه الگوریتم CS tree به اشتباه دو خوشه را با هم ادغام می کند.



شکل۲) خطاهای روش CS tree در ترکیب خوشه های یک بعدی و ایجاد خوشه های چند بعدی- قسمت A خطا در تعداد خوشه ها، قسمت B خطا در شکل خوشه ها، قسمت C خطا در مرز خوشه ها

همچنین در الگوریتم CS tree در هر بعد شکست یا ادغام خوشه ها به صورت مستقل انجام می شود و سپس تغییرات بر روی خوشه های چند بعدی ذخیره شده در درخت اعمال می شود. اما این روش ادغام سبب روی هم افتادگی خوشه ها و ایجاد خوشه هایی با مرزهای نادقیق می شود.

در شکل ۳ در قسمت A دو خوشه در فضای داده ها نشان داده شده است. با افزوده شدن داده های یکی از خوشه ها، تصویر خوشه ها در بعد x با هم همسایه می شوند و خوشه های در بعد x ادغام می شوند. در نتیجه با یکی شدن خوشه ها در هر دو بعد دو خوشه در فضای دو بعدی در یک خوشه قرار می گیرند.

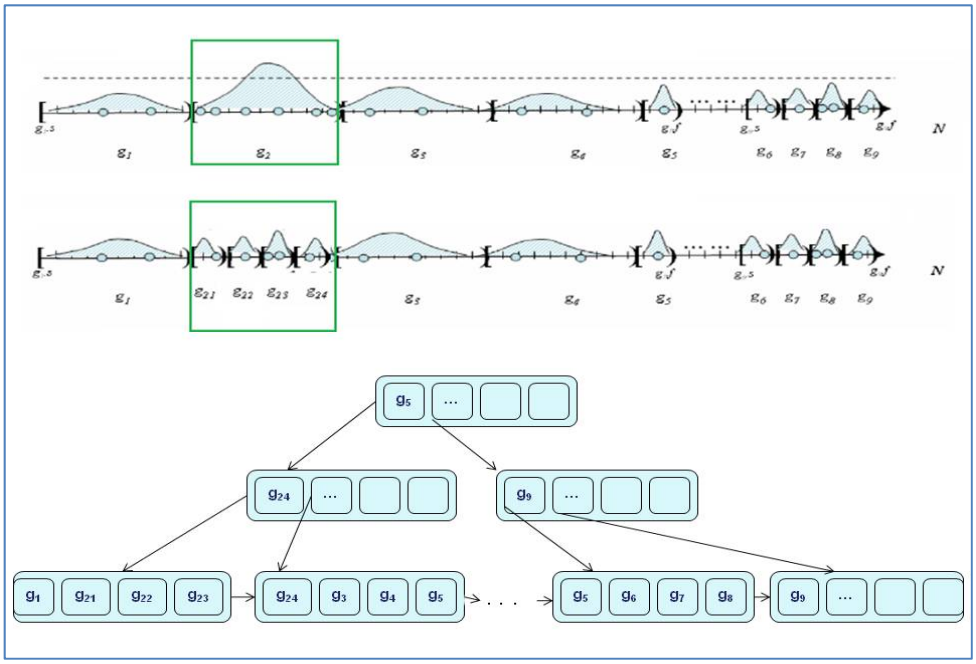


شکل ۳) روی هم افتادگی خوشه ها در بروز رسانی به روش CS tree

۶. الگوریتم پیشنهادی

در این قسمت به بررسی الگوریتم ارائه شده برای بهبود روش CS tree می پردازیم. در این الگوریتم به ازای هر داده جدید گام های زیر تکرار می شوند:

در ابتدا، در صورت لزوم مولفه های داده نرمال سازی می شوند و ابعادی که در خوشه بندی موثر نیستند، حذف می شوند. سپس به ازای هر مولفه داده، سلول متناظر با آن داده شناسایی و اطلاعات آماری سلول بروز می شود. هر سلول متراکم به h سلول کوچکتر شکشته می شود و اطلاعات آماری سلول های جدید بر اساس آماره های سلول اولیه محاسبه می شوند. در این الگوریتم ، به ازای هر بعد از ساختاری مشابه B^+tree برای ذخیره اطلاعات آماری سلول ها استفاده می شود. در این ساختار، در ابتدا سلول های اولیه به درخت اضافه می شوند سپس با شکست هر سلول اطلاعات آماری سلول های جدید بر اساس اطلاعات سلول قبلی محاسبه و سلول های جدید به درخت اضافه می شوند و سلول اولیه در سطوح مختلف درخت با سلول جدیدی که انتهای بازه سلول اولیه را در بردارد، جایگزین می شود. در شکل ۴ سلول g_2 متراکم شده و در نتیجه به چهار سلول کوچکتر $\{g_{24},g_{23},g_{22},g_{21}\}$ تقسیم می شود. در شکل نحوه ی ذخیره سازی سلول ها در درخت پس از تقسیم نشان داده شده است.



شکل۴) ذخیره سازی اطلاعات سلول ها با ساختار B^+tree

قضیه ۱: اگر در بعد N ، اندازه سلول واحد λ باشد و در هر افراز یک سلول به h سلول کوچکتر شکسته شود، آنگاه حداکثر تعداد افزار برای تبدیل یک سلول اولیه به سلول واحد $\log_h \frac{\text{range(N)}}{\lambda}$ می باشد.[۱۱]

قضیه ۲: اگر در B^+tree تعداد داده ها n و تعداد فرزندان هر نود حداکثر m باشد، آنگاه عمق درخت و پیچیدگی محاسباتی عملیات حذف، اضافه و پیمایش درخت نیز برابر $O(\log_m n)$ می باشد.[۱۲]

با توجه به دو قضیه فوق و با فرض اینکه تعداد کل سلول های ذخیره شده در B^+tree برابر $\frac{range(N)}{\lambda}$ است و هر نود در درخت B^+tree حداکثر h فرزند داشته باشد، آنگاه عمق درخت حاصل$O(\log_h range(N)/\lambda)$ خواهد بود. در نتیجه متوسط تعداد پیمایش لازم برای دسترسی به یک سلول در ساختار پیشنهادی تقریبا معادل حداکثر تعداد شکست لازم برای تبدیل سلول اولیه به سلول واحد است. زمان دسترسی به سلول ها در روش پیشنهادی کمتر از روش CS tree می باشد و در نتیجه در الگوریتم پیشنهادی سرعت پردازش روی خط داده ها افزایش یافته است.

در این الگوریتم به منظور خوشه بندی دقیق تر اصلاحات زیر اعمال شده است:

- بازتعریف مفهوم همسایگی و رفع مشکل تقسیم بی معنی خوشه ها
- اصلاح روش ترکیب خوشه ها و ایجاد خوشه های دقیق تر
- اصلاح روند بروز رسانی خوشه ها
- کاهش میزان جستجوها برای شناسایی سلول های همسایه متراکم
- اصلاح ساختار نمایش خوشه ها

۱.۶ بازتعریف مفهوم همسایگی و رفع مشکل تقسیم بی معنی خوشه ها

در الگوریتم پیشنهادی برای رفع مشکل تقسیم بی معنی خوشه ها با معنی، مفهوم همسایگی اصلاح شده است. در این روش دو خانه همسایه اند اگر وجه مشترک داشته باشند و یا در نقطه ای مشترک باشند. این تعریف از همسایگی سبب می شود که تعداد سلول های همسایه یک سلول $2d + 2^d$ شود [۱۳]و در نتیجه با افزایش تعداد ابعاد ، تعداد همسایه ها افزایش نمایی خواهد داشت و خوشه بندی زمانبر می شود. در الگوریتم پیشنهادی این مشکل با محدود کردن فضای جستجو و اصلاح روند ترکیب خوشه ها که در ادامه به بررسی آن می پردازیم، برطرف می شود.

۲.۶ اصلاح روش ترکیب خوشه ها و ایجاد خوشه های دقیق

روش جدید ترکیب خوشه ها مطرح شده در این مقاله بر پایه قضیه زیر است:

قضیه : اگر یک مجموعه نقطه S در یک خوشه k بعدی وجود داشته باشند آنگاه نقاط s جزئی از یک خوشه در هر یک از k-1 بعد زیرین درفضا هستند.[۱۱]

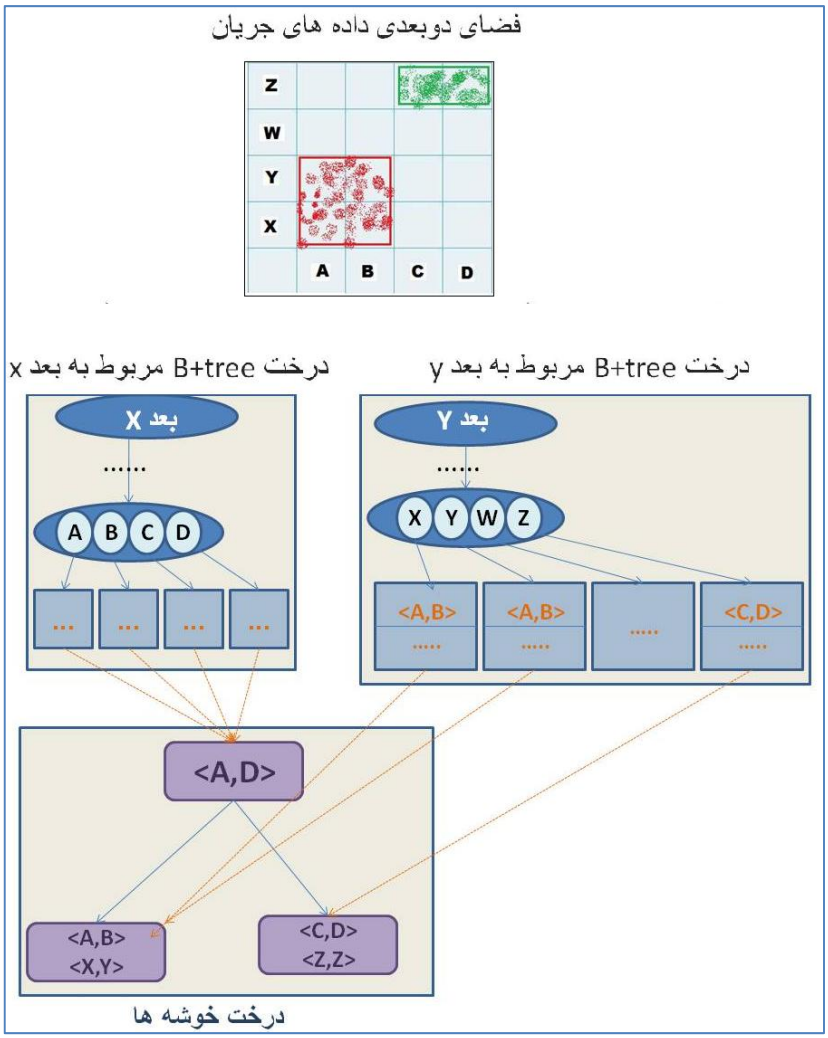
اثبات : اگر یک خوشه K بعدی شامل نقاط S، را مجموعه ای از سلول های متراکم k بعدی در نظر گرفته شود، آنگاه هر یک از این سلو ل ها شامل زیر مجموعه از نقاط S است که آنها را متراکم می کند. اگر این سلول ها بر روی ابعاد مختلف تصویر شوند، در تصاویر این سلول ها در هر یک از ابعاد نیز تصویر نقاط S قرار دارند و در نتیجه بازه مربوط به تصاویر این سلول ها در هر بعد متراکم خواهد بود. از طرفی از آنجا که این سلول ها در خوشه K بعدی همسایه اند، پس تصاویر آنها روی ابعاد نیز همسایه است. بر این اساس، تصاویر سلول های متراکم و همسایه در هر یک از k بعد زیرین نیز در یک خوشه قرار دارند.

بر اساس این قضیه تنها نقاطی که در K-1 بعد زیرین در یک خوشه قرار دارند، می توانند در K بعد نیز در یک خوشه باشند. در روش پیشنهادی به ازای هر داده در اولین بعد بازه متناظر با مولفه اول داده شناسایی و اطلاعات بازه بروز می شود. اگر این بازه متراکم شود، با بازه های همسایه اش که داده های آنها در بعد اول با داده های بازه مورد بررسی در یک خوشه بوده اند، ادغام و تشکیل یک خوشه دو بعدی می دهند.این روند به ازای ابعاد بعدی تکرار می شود. به عبارت دیگر، در بعد K ام خوشه جدید با خوشه های بازه های همسایه مقایسه می شود و در صورتیکه که در k-1 بعد قبلی نیز همسایه باشند ، با هم ادغام و خوشه جدید را تشکیل می دهند. با این روش بدون نیاز به ذخیره اطلاعات تک تک سلول ها چندبعدی، تنها با ذخیره سازی اطلاعات سلول های متراکم که در بعدهای زیرین نیز متراکم هستند می توان خوشه هایی دقیق ایجاد کرد .

در روش ترکیب ارائه شده لازم است برای مقایسه همسایگی در k-1 بعد قبلی، اطلاعات درباره خوشه های ابعاد زیرین یک داده در سلول متناظر با بعد kام موجود باشد، به این منظور در هر سلول واحد مجموعه ای از گروه ها تعریف شده است که هر گروه مرز خوشه ی مربوط به ابعاد قبلی داده هایی را که در بعد جاری متعلق به سلول می باشند ذخیره می کند. مجموعه گروه ها بر اساس بازه های ابعاد قبلی مرتب شده اند.

برای بروز رسانی خوشه ها، در این روش به ازای هر گروه در هر سلول واحد متراکم یک اشارگر به نود متناظر با خوشه آن گروه در درخت وجود دارد

در این روش نیز مشابه CS tree به منظور ذخیره سازی خوشه های موجود در جریان داده در هر لحظه، یک ساختار درختی ارائه شده است. در این ساختار درختی تعداد سطوح برابر تعداد ابعاد جریان است و هر نود سطح k متناظر با یک خوشه k بعدی در جریان می باشد. در این درخت بر خلاف CS tree در هر نود در سطح k مرز دقیق خوشه k بعدی در تمام ابعاد ذخیره می شود. در نتیجه، مشکلات روش CS tree که در زمان ایجاد خوشه های چند بعدی و بروز رسانی خوشه ها رخ می دهند، برطرف می شود. در شکل ۵ ساختار ارائه شده برای ذخیره خوشه ها که امکان خوشه بندی دقیق و بروز آنها را فراهم می کند، نشان داده شده است.



شکل ۵) ساختار ذخیره سازی خوشه ها در فضای چند بعدی

۳.۶ اصلاح روند بروز رسانی خوشه ها

در زمان ادغام دو خوشه یا شکست یک خوشه ابتدا مرزهای خوشه جدید در ساختار B+tree اصلاح می شود، سپس از طریق اشارگر، تغییرات آن خوشه بر روی نود متناظر با آن خوشه، در درخت اعمال می شود. در ادامه تغییرات به خوشه های ابعاد بعدی در درخت ارسال می شود تا مرزهای آنها نیز بروز شود. در این روش به علت ذخیره سازی مرز دقیق خوشه ها در تمام ابعاد، بروز رسانی به درستی انجام می شود و مشکلات روش CS tree را ندارد.

همچنین با پیمایش درخت خوشه هایی که به آنها اخیراً ارجاعی نشده است، حذف می شوند. عمل حذف سبب دقیق تر شدن خوشه ها و مصرف کمتر حافظه برای ذخیره آنها می شود. حذف خوشه در سه حالت زیر انجام می شود:

۱. زمانیکه به خوشه ای در هر یک از ابعاد ارجاع شود، اگر از آخرین ارجاع به آن خوشه بیش ازحد مشخصی گذشته باشد، آن خوشه در آن بعد و فرزندانش در ابعاد بعدی حذف می شود.

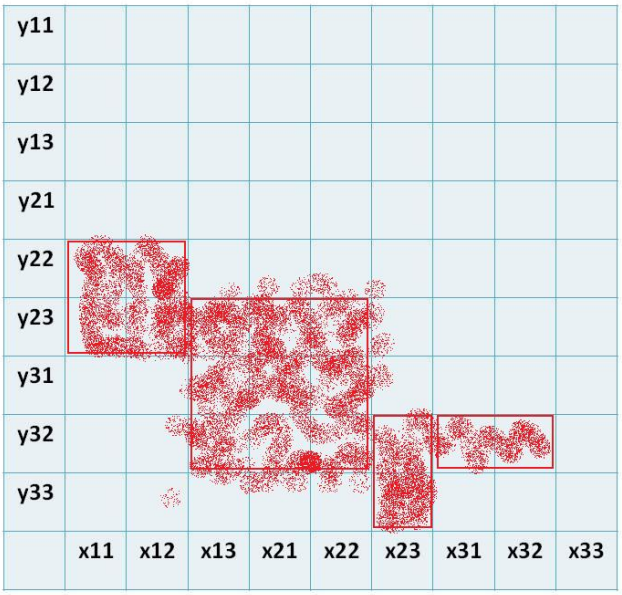
۲. با توجه به ذخیره یک خوشه به صورت DNF به ازای هر یک قسمت از خوشه آخرین ارجاع به آن قسمت مشخص است و می توان قسمت های از یک خوشه که از آخرین ارجاع به آنها بیش ازحد مشخصی گذشته باشد، حذف می شوند.

۳. در بازه های زمانی مشخصی درخت پیمایش می شود و کلیه خوشه های قدیمی پاکسازی می شوند. حالت دوم زمانی انجام می شود که حافظه مصرفی برای درخت بیش از حد مشخص شود.

۴.۶ اصلاح ساختار نمایش خوشه ها

در الگوریتم پیشنهادی ساختاری مشابه DNF [۶] برای نمایش خوشه ها استفاده می شود[۱۳] که علاوه بر ذخیره دقیق مرزهای خوشه ها ، بدون پس پردازش خاصی می توان نتایج خوشه بندی را در اختیار کاربران قرار می گیرد. در این ساختار، یک خوشه با حداقل تعداد مکعب مستطیل ممکن نمایش داده می شود و به ازای هر مکعب دو گوشه ی بالا سمت راست و پایین سمت چپ ذخیره می شوند. در شکل ۶ نحوه نمایش یک خوشه با این ساختار نشان داده شده است. مرزهای خوشه ی شکل ۶ با عبارت زیر مشخص می شوند.

$$<(x11, y23) \vee (x12,y22)> \wedge <(x13, y32) \vee (x22,y22)> \wedge <(x23, y33) \vee (x23,y32)> \wedge <(x13, y32) \vee (x33,y32)>$$



شکل ۶) نحوه نمایش یک خوشه

۷. نتیجه گیری

در بسیاری از کاربرد ها لازم است جریان داده ای با ابعاد زیاد خوشه بندی شود. یکی از الگوریتم های ارائه شده برای خوشه بندی جریان داده ها CS tree می باشد. این روش از نوع خوشه بندی توری است که بر اساس توزیع داده ها در سلول ها در هر بعد خوشه ها ایجاد می کند و با ترکیب آنها خوشه های چند بعدی ساخته می شوند. این روش در حافظه محدود و با سرعت متناسب با نرخ ورود داده ها به خوشه بندی روی خط داده ها می پردازد. با وجود ویژگی های کارامد روش CS tree، در این الگوریتم اشکالاتی وجود دارد که سبب می شود به ازای تعداد ابعاد زیاد مقیاس پذیر نباشد. در الکوریتم ارائه شده در این مقاله ساختار ذخیره سازی داده ها در هر بعد اصلاح شده و سرعت دسترسی به داده ها افزایش یافته است. همچنین با بازتعریف مفهوم همسایگی مشکل تقسیم بی معنی خوشه ها رفع و از میزان جستجوها برای شناسایی سلول های همسایه متراکم کاسته شده است. روشی جدید برای ترکیب خوشه ها و ایجاد خوشه های دقیق تر ارائه و روند بروز رسانی خوشه ها اصلاح شده است. علاوه بر این با اصلاح ساختار نمایش خوشه ها، مرزهای خوشه ها دقیق تر ذخیره شده اند و نیازی به پس پردازشی برای ارائه نتایج خوشه بندی به کاربران نمی باشد.

۸. مراجع

[1] Park, N.H.; Lee, W.S.; "***A statistical grid-based clustering over data streams***", ACM SIGMOD, Record 33(1), P.P. 32–37, 2004.

[2] Park, N.H.; Lee, W.S.; "***Cell trees: an Adaptive Synopsis structure for clustering multi-dimensional on-line data streams***", Data & Knowledge Engineering, 63(2), P.P.528–549, 2007.

[3] Gaber, M. M.; Zasavsky, A.; Krishnaswamy, S.; " ***Mining Data Streams: A Review",*** SIGMOD conference 34(2) , 2005.

[4] Guha, S.; Meyerson, A.; Mishra, N.; Motwani, R.; O'Callaghan, L.; ***"Clustering data streams: Theory and practice"***, IEEE Trans. Knowledge Data Engineering, 15 (3), P.P. 626 515–528, 2003.

[5] O'Callaghan, L.; Mishra, N.; Meyerson, A.; Guha, S.; Motwani, R.; "***Streaming-data Algorithms for High-quality Clustering",***IEEE International Conference on Data Engineering, 2002.

[6] Aggarwal C. C.; "***Data Streams Models and Algorithms***", Springer Publishers, 2007.

[7] Lin, N. P.; Chang, C.; Chueh, H.; Chen, H.; Hao, W; ***" A deflected Grid-based Algorithm for Clustering Analysis***", W. Trans. on Comp.VOL. 7, Issue 4, P.P.125-132, 2008.

[8] Goil, S.; Nagesh, H.; Choudhary, A.; "***MAFIA: Efficient and scalable subspace clustering for very large data sets".*** Technical Report CPDC-TR-9906-010, Center for Parallel and Distributed Computing, Department of Electrical & Computer Engineering, Northwestern University, 1999.

[9] Park, N.H.; Lee, W.S.; "***Efficiently tracing clusters over high-dimensional on-line data streams***", Data & Knowledge Engineering, 2009.

[10] Hsu, C. M.; Chen, M. S.; "***Subspace Clustering of High Dimensional Spatial Data with Noises***", PAKDD 2004, LNAI 3056, P.P. 31-40, 2004.

[11] Agrawal, R.; Gehrke, J.; Gunopulos, D.; Raghavan, P.; "***Automatic Subspace Clustering of High Dimensional Data***". Data Mining Knowledge Discovery, Vol. 11, 1, P.P. 5-33, 2005.

[12] Mehta, D. P; Sahni, S; "***Handbook of Data Structures and Applications***", Chapman & Hall/CRC, chapter 15, 2004.

[13] Hinneburg, A.; Keim, D. A.; "***Optimal Grid-Clustering: Towards Breaking the Curse of Dimensionality in High-Dimensional Clustering***", In Proceedings of the 25th international Conference on Very Large Data Bases.1999.

زیر نویس ها

[1] Data Streams
[2] Partitioning Clustering
[3] Grid-based Clustering
[4] Clustering Statistics Tree
[5] Decay Rate
[6] Disjunctive Normal Form