دانشکده مهندسی کامپیوتر و فناوری اطلاعات

رشته مهندسی کامپیوتر گرایش هوش مصنوعی و رباتیک

پایان‌نامه کارشناسی ارشد

بهبود الگوریتم‌های خوشه‌بندی برای کلان‌داده با استفاده از محاسبات خوشه‌ای

نگارنده: زهره فریدون مقدم

استاد راهنما

دکتر هدی مشایخی

استاد مشاور

دکتر وحید ابوالقاسمی

بهمن ۱۳۹۵

شماره:

باسمه تعالی

تاریخ:

مدیریت تحصیلات

ویرایش:

تکمیلی

دانشکده :

گروه :

پایان نامه کارشناسی ارشد خانم زهره فریدون مقدم به شماره دانشجویی: ۹۳۱۳۱۰۴

تحت عنوان:

بهبود الگوریتم‌های خوشه‌بندی برای کلان‌داده با استفاده از محاسبات خوشه‌ای

در تاریخ ..... توسط کمیته تخصصی زیر جهت اخذ مدرک کارشناسی ارشد مورد ارزیابی و با درجه ..... مورد پذیرش قرار گرفت.

اساتید راهنما	امضاء	اساتید مشاور	امضاء
نام و نام خانوادگی : دکتر هدی مشایخی		نام و نام خانوادگی : دکتر وحید ابوالقاسمی	

اساتید داور	امضاء	نماینده تحصیلات تکمیلی	امضاء
نام و نام خانوادگی :		نام و نام خانوادگی :	
نام و نام خانوادگی :			
نام و نام خانوادگی :			
نام و نام خانوادگی :			

تقدیم به

مقدسترین واژه ها در لغت نامه دلم، مادر مهربانم که زندگی‌ام را مدیون مهر و عطوفت او می‌دانم.

و تقدیم به

روح پاک پدرم که پشتکار و سخت کوشی را از اوآموختم.

تشکر و قدردانی

از استاد گرامیم **سرکار خانم دکتر هدی مشایخی** بسیار سپاسگزارم چرا که بدون راهنمایی‌های ایشان تأمین این پایان‌نامه بسیار مشکل می‌نمود.

از جناب آقای **دکتر وحید ابوالقاسمی**سپاسگزارم به دلیل یاری‌ها و راهنمایی‌های بی‌چشم‌داشت ایشان که بسیاری از سختی‌ها را برایم آسان‌تر نمودند.

از جناب آقای **دکتر هادی صدوقی** که در راستای انجام این پایان‌نامه کمک شایانی به بنده فرموند، نهایت قدردانی و تشکر را دارم.

تعهد نامه

اینجانب **زهره فریدون مقدم** دانشجوی دوره کارشناسی ارشد رشته هوش مصنوعی دانشکده کامپیوتر و فناوری اطلاعات دانشگاه صنعتی شاهرود، نویسنده پایان‌نامه بهبود الگوریتم‌های خوشه‌بندی برای کلان داده با استفاده از محاسبات خوشه‌ای، تحت راهنمائی خانم دکتر هدی مشایخی متعهد می‌شوم.

- تحقیقات در این پایان‌نامه توسط اینجانب انجام‌شده است و از صحت و اصالت برخوردار است.
- در استفاده از نتایج پژوهش‌های محققان دیگر به مرجع مورداستفاده استناد شده است.
- مطالب مندرج در پایان‌نامه تاکنون توسط خود یا فرد دیگری برای دریافت هیچ نوع مدرک یا امتیازی در هیچ جا ارائه نشده است.
- کلیه حقوق معنوی این اثر متعلق به دانشگاه صنعتی شاهرود است و مقالات مستخرج با نام «دانشگاه صنعتی شاهرود» یا «**Shahrood University of Technology**» به چاپ خواهد رسید.
- حقوق معنوی تمام افرادی که در به دست آمدن نتایج اصلی پایان‌نامه تأثیرگذار بوده‌اند در مقالات مستخرج از پایان‌نامه رعایت می‌گردد.
- در کلیه مراحل انجام این پایان‌نامه، در مواردی که از موجود زنده (یا بافت‌های آن‌ها) استفاده شده است ضوابط و اصول اخلاقی رعایت شده است.
- در کلیه مراحل انجام این پایان‌نامه، در مواردی که به حوزه اطلاعات شخصی افراد دسترسی یافته یا استفاده شده است اصل رازداری، ضوابط و اصول اخلاق انسانی رعایت شده است. **تاریخ**

امضای دانشجو

مالکیت نتایج و حق نشر

- کلیه حقوق معنوی این اثر و محصولات آن (مقالات مستخرج، کتاب، برنامه‌های رایانه‌ای، نرم‌افزارها و تجهیزات ساخته شده است) متعلق به دانشگاه صنعتی شاهرود است. این مطلب باید به نحو مقتضی در تولیدات علمی مربوطه ذکر شود.
- استفاده از اطلاعات و نتایج موجود در پایان‌نامه بدون ذکر مرجع مجاز نمی‌باشد.

چکیده

امروزه در کاربردهای مختلف همچون سامانه‌های کنترل ترافیک، سنسورهای هواشناسی، سیستم‌های پزشکی، شبکه‌های اجتماعی و غیره، با سرعت بسیار زیادی داده تولید می‌شود. به این داده‌ها، جریان‌داده اطلاق شده و تحلیل آنها نوعا نیازمند روش‌های تکاملی و افزایشی می‌باشد. خوشه‌بندی یکی از روش‌های متداول تحلیل و یادگیری است و الگوریتم‌های متعددی در این حوزه ارائه شده است. خوشه‌بندی جریان داده‌ها به دلایل مختلفی هم‌چون عدم دسترسی همزمان به تمامی داده، تغییرات خوشه‌ها، و غیره، روش‌هایی متفاوت از خوشه‌بندی سنتی را طلب می‌کند.

در این پایان‌نامه الگوریتم خوشه‌بندی مبتنی بر مدل ارائه خواهد شد که با استفاده از تئوری احتمال شرطی و براساس اطلاعات پیشین، خوشه‌بندی احتمالاتی را انجام می‌دهد. الگوریتم پیشنهادی با داده-های مصنوعی و داده‌های واقعی آزمایش می‌شود و در ادامه با روش‌های FCM و Gustafson-Kessel مقایسه خواهد شد. براساس نتایج حاصل از مقایسه، این الگوریتم دقت خوشه‌بندی را نسبت به الگوریتم-های ذکر شده، بهبود می‌دهد و همچنین مقاومت بیشتری در مقابل نویز دارد.

با توجه به ماهیت افزایشی جریان‌داده، الگوریتم پیشنهادی برای این نوع داده‌ها توسعه داده شده و بعد از پیاده‌سازی با استفاده از چارچوب محاسباتی خوشه‌ای Spark، ارزیابی می‌شود. با پیاده‌سازی موازی الگوریتم، زمان اجرا کاهش می‌یابد. آزمایشات نشان می‌دهندکه الگوریتم پیشنهادی مقایس‌پذیری لازم برای کار با حجم بالای داده را دارد.

کلمات کلیدی: خوشه‌بندی، خوشه‌بندی احتمالاتی، جریان‌داده، خوشه‌بندی جریان‌داده، محاسبات خوشه‌ای

فهرست مطالب

شکل ها

جدول ها

فصل اول

مقدمه

۱-۱- مقدمه

امروزه در کاربردهای مختلف همچون سامانه‌های کنترل ترافیک، سنسورهای هواشناسی، سیستم‌های پزشکی، شبکه‌های اجتماعی و غیره، با سرعت بسیار زیادی داده تولید می‌شود. این داده‌ها، جریان‌داده اطلاق شده و تحلیل آنها نوعا نیازمند روش‌های تکاملی و افزایشی می‌باشد(Bones 2016) .

مواردی چون حجیم بودن داده‌ها، تغییر مفهوم، ترتیبی و تک گذر بودن بودن دسترسی به داده‌ها، از جمله چالش‌های تحلیل جریان داده‌ها محسوب می‌شوند. خوشه‌بندی یکی از روش های متداول تحلیل داده‌ها است و الگوریتم‌های زیادی در این حوزه ارائه شده است. هدف خوشه‌بندی کشف گروه‌هایی در داده است که شباهت اعضای درون گروه زیاد و شباهت بین گروه‌ها کم باشد. در این پایان‌نامه الگوریتم خوشه‌بندی ارائه می‌شود که با استفاده از تئوری احتمال شرطی، براساس اطلاعات پیشین خوشه‌بندی احتمالاتی را انجام می‌دهد. اختصاص هر داده به خوشه‌ها براساس احتمال شباهت هر داده به پارامترهای خوشه‌ها صورت می‌گیرد.

با توجه به حجیم بودن داده‌ها و سرعت ورود آنها، پردازش و تحلیل جریان داده پیچیده می‌باشد. یکی راه‌های کاهش پیچیدگی استفاده از پردازش موازی در قالب محاسبات خوشه‌ای است. در محاسبات خوشه‌ای، با پخش کردن داده‌ها روی چندین مرکز محاسباتی و اجرای بخشی از محاسبات به صورت مستقل روی هر مرکز، الگوریتم‌ها به صورت توزیع‌شده اجرا شده و سپس تجمیع نتایج صورت می‌گیرد (Panigrahi, Lenka 2016).

یکی از مدل‌های محاسبات خوشه‌ای مدل نگاشت-کاهش[1] است که در این مدل پردازش روی داده‌ها به صورت موازی در گام نگاشت انجام شده و تجمیع نتایج در گام کاهش انجام می‌شود. در این

[1] Map-Reduce

پایان‌نامه الگوریتم پیشنهادی با ابزار Apache Spark پیاده‌سازی شده است. به این ترتیب زمان اجرای الگوریتم برای داده‌های حجیم کاهش می یابد.

۱-۲- کلان‌داده

به حجم بزرگی از داده‌ها با ساختار متنوع، کلان‌داده[1] گفته می‌شود، که با گذشت زمان در حال رشد و تغییر هستند. بدیهی است برای پردازش این نوع داده‌ها نمی‌توان از روش‌های سنتی استفاده کرد. عملیات اصلی پردازش این نوع داده‌ها مربوط به برداشت و جمع‌آوری، ذخیره‌سازی، جست‌وجو، اشتراک‌گذاری، تحلیل و نمایش آن‌ها است.

کلان‌داده عموماً به‌خاطر تولید و جمع‌آوری داده‌ها از مجموعه بزرگی از تجهیزات و ابزارهای مختلفی مانند گوشی‌های موبایل، حسگرهای محیطی، لاگ نرم‌افزارهای مختلف، دوربین‌ها، میکروفون‌ها، شبکه‌های حسگر بی‌سیم و... می‌باشد و در طول زمان، حجم این داده‌ها با سرعت بالایی در حال افزایش است.

داده‌ها می‌توانند داده‌های تجاری، پزشکی، امنیتی، هواشناسی، ژنتیک، شبیه‌سازی‌های پیچیده فیزیکی، تحقیقات زیست‌شناسی و محیطی، جست‌وجوی اینترنت، تحلیل‌های اقتصادی و مالی و تجاری باشند. به همین دلیل روی حجم بالای داده‌ها، می‌تواند تحلیل‌های بهتر و بیشتری صورت گیرد و دانش بیشتری از این تحلیل‌ها بدست آید.

[1] BigData

۱-۳- چالش‌های کلان‌داده

۵ عملیات اصلی روی داده‌ها را می‌توان به صورت ذیل نام برد

۱. جمع‌آوری
۲. ذخیره‌سازی
۳. جستجو
۴. به اشتراک‌گذاری
۵. تحلیل

که این عملیات بر روی کلان‌داده، هر کدام به نوعی مشکلات اساسی را به همراه دارد، که در ادامه به بررسی آنها می‌پردازیم.

- حجم داده[1]: حجم داده با سرعت بسیار زیادی در حال افزایش است. این داده از منابع مختلفی نظیر شبکه‌های اجتماعی، لاگ سرورهای وب، جریان‌های ترافیک، تصاویر ماهواره‌ای، جریان‌های صوتی، تراکنش‌های بانکی، محتوای صفحات وب، اسناد دولتی و ... با سرعت بسیار بالایی تولید می‌شود.
- نرخ تولید[2]: نرخ تولید به صورت نمایی در حال تولید است.
- تنوع[1]: در جمع‌آوری داده‌ها از منابع مختلف، داده‌هایی با ساختارهای مختلف گردآوری می‌شود. مثلا ساختار مرورگرها می‌تواند xml و json باشد و یا اینکه در داده‌های دوربین‌های ترافیکی، داده‌ها به صورت ویدئو و عکس می‌باشند.

[1] Volume
[2] Velocity

- صحت[۲]: با توجه به اینکه داده‌ها از منابع مختلف دریافت می‌شوند، تمامی این داده‌ها ممکن است درست نباشند و نمی‌توان به همه آن‌ها اعتماد کرد. مثلا در شبکه‌های اجتماعی هر فرد اطلاعات شخصی خود را در آن ثبت می‌کند و ممکن است این اطلاعات در مورد آن فرد، صحت نداشته باشد.
- اعتبار[۳]: گاهی با وجود صحت داده‌ها، امکان استفاده‌ی آن‌ها وجود ندارد. به طور مثال داده‌هایی که تاریخ انقضا داشته باشند.
- نوسان[۴]: در طول زمان ارزش داده‌ها ممکن است تغییر کند. مثلا در تحلیل‌های اقتصادی و تجاری این تغییرات از اهمیت بالایی برخوردار هستند. نگهداری طولانی مدت این اطلاعات هزینه بر است.
- نمایش[۵]: به دلیل وجود ارتباطات مختلف و پیچیده بین داده‌ها و همچنین بالا بودن حجم داده‌ها، نمایش این ارتباطات و نتایج تحلیل داده‌های حجیم، یکی از چالش‌هایی است که امروزه با آن مواجه هستیم.

۱-۴- جریان‌داده

جریان‌داده نوعی از کلان‌داده است، که ماهیت افزایشی در طول زمان دارند. این داده‌ها دنباله‌ای مرتب و نامتناهی هستند. نحوه‌ی دستیابی به آنها به ترتیب ورود است و تنها یکبار مورد دسترسی قرار می‌گیرند.

---

[1] Variety
[2] Veracity
[3] Validity
[4] Volatility
[5] Visualization

استفاده از این نوع داده‌ها چالش‌های زیر را در پی دارد:

- حجیم بودن داده‌ها
- تغییر مفاهیم و نمایش مدل‌ها به دلیل ورود پیوسته و سرعت بالای ورود داده‌ها
- ترتیبی بودن دسترسی داده‌ها
- تک‌گذر بودن؛ دسترسی به هر داده فقط یکبار امکان‌پذیر است
- مقایس‌پذیری الگوریتم‌ها؛ مقایس‌پذیری اغلب به عنوان یک مشخصه در یک الگوریتم خوشه‌بندی جریان‌داده‌ای بیان می‌شود.

۱-۵- تعریف مساله

همان‌طور که در بخش (۱-۲) کلان‌داده و (۱-۴) جریان‌داده گفته شد، داده در طول زمان به صورت افزایشی تولید می‌شوند و پردازش این داده‌ها مسئله بسیار مهمی می‌باشد. یکی از پردازش‌ها، خوشه‌بندی است. الگوریتم‌های خوشه‌بندی زیادی ارائه شده است. اما به دلایل هزینه‌های بالای ذخیره‌سازی و پردازش آن‌ها نمی‌توان با استفاده از روش‌های سنتی، جریان‌داده را به صورت یکجا خوشه‌بندی کرد.

مسئله اصلی ارائه الگوریتم خوشه‌بندی مناسب و مطابق با ماهیت افزایشی جریان‌داده می‌باشد.

منظور از ارائه این الگوریتم کاهش پیچیدگی و هزینه تحلیل داده است.

۱-۵-۱- چالش‌های مسئله

الگوریتم‌های خوشه‌بندی جریان‌داده با چالش‌های زیر روبرو هستند:

- مشخص نبودن تعداد خوشه‌ها
- نوع داده‌ها؛ مختلف بودن نوع ساختار داده‌ها

- ابعاد بالای داده‌ها: هنگامی‌که ابعاد داده‌ها زیاد می‌شود، احتمال وجود نداشتن بعضی از ویژگی‌ها برای داده نیز افزایش پیدا می‌کند، با توجه به معیارهای تشابه، تعداد داده‌های مشابه کم می‌شود و الگوریتم خوشه‌بندی قادر به خوشه‌بندی دقیق این داده‌ها نمی‌باشد.
- ارزیابی خوشه‌ها: معیارهایی برای خوشه‌بندی مناسب می‌باشد.
- پیش‌پردازش جریان‌داده: پیش پردازش داده‌ها زمان زیادی نیاز دارد.
- اشکال دلخواه خوشه‌ها: در داده جریانی ممکن است اشکال خوشه‌ها متفاوت باشد، مانند باید کروی و یا جداپذیرخطی و دایره‌ای باشد.
- شناسایی نقاط پرت

۱-۶- اهداف پایان‌نامه

به دلیل اینکه جریان‌داده در طول زمان به‌صورت افزایشی تولید می‌شود، ذخیره‌سازی و پردازش این نوع داده‌ها به‌صورت یکجا هزینه‌بر است. هدف این پایان‌نامه ارائه الگوریتمی مبتنی بر مدل می‌باشد. با استفاده از این روش نیازی به ذخیره‌سازی کل داده نیست و می‌توان کل داده را به صورت یک مدل نگهداری کرد. همچنین این الگوریتم به صورت افزایشی توسعه داده می‌شود. در روند اجرای این الگوریتم با استفاده از پنجره‌هایی در واحد زمان، داده‌ها به تدریج اضافه می‌شوند و سپس مراکز خوشه‌بندی بروزرسانی می‌گردد. در این پایان‌نامه روش‌های موجود خوشه‌بندی را بررسی می‌کنیم. روش ارائه شده در چارچوب محاسبات خوشه‌ای با ابزار آپاچی اسپارک پیاده می‌شود و در نتیجه زمان پردازش کاهش می‌یابد.

۱-۷- ساختار پایان‌نامه

این پایان‌نامه شامل ۵ فصل می‌باشد. در فصل اول مقدمه‌ای از پایان‌نامه بیان شده است .فصل دوم مروری بر کارهای گذشته است که به تعریف خوشه‌بندی و انواع آن پرداخته شده است و در ادامه انواع خوشه‌بندی جریان داده و تعریف محاسبات خوشه‌ای و ابزارهای محاسبات خوشه‌ای و همچنین بررسی ابزار آپاچی اسپارک بیان شده است. در فصل سوم الگوریتم پیشنهادی برای حالتی که داده‌ها، دسته‌ای و افزایشی باشد ارائه شده است. در فصل چهارم آزمایشات مربوط به این الگوریتم پیشنهادی و آزمایشات مربوط به پیاده‌سازی روی ابزار اسپارک توضیح داده شده است. در فصل پنجم نتایج و کارهای آینده شرح داده شده است.

۲- فصل دوم

مروری بر کارهای گذشته

۲-۱- مقدمه

در این فصل تعریف خوشه‌بندی و انواع آن بیان می‌شود. سپس در مورد خوشه‌بندی جریان داده و خوشه‌بندی افزایشی توضیح داده می‌شود. در بخش بعدی درمورد محاسبات خوشه‌ای و ابزارهای آن نیز اشاره‌ای خواهد شد و در آخر ابزار محاسبات خوشه‌ای اسپارک شرح داده خواهد شد.

۲-۲- تعریف خوشه‌بندی

خوشه‌بندی یا تحلیل خوشه[1]، به دسته‌بندی نمونه‌هایی که اعضای آن از بعضی جهات به هم شبیه هستند، گفته می‌شود. خوشه‌بندی، زیر شاخه‌ی داده کاوی و یادگیری ماشین و یک روش بدون ناظر است (Robert 2014).

۲-۳- انواع خوشه‌بندی

روش‌های خوشه‌بندی را به طور کلی می‌توان به دسته‌های زیر تقسیم‌بندی کرد:

۲-۳-۱- خوشه‌بندی سلسله مراتبی

در این روش از خوشه‌بندی، مجموعه داده براساس معیار تشابه و دو عملیات اصلی ادغام و تقسیم، خوشه‌بندی می‌شود.

این خوشه‌بندی به دو روش بالا به پایین و پایین به بالا انجام می‌پذیرد(Corpet 1988). پایین به بالا ابتدا هر نمونه را داخل یک خوشه مجزا قرار می‌دهد، سپس در هر مرحله خوشه‌هایی که بیشترین شباهت را بهم دارند، ادغام می‌کند. این عملیات تا به یک خوشه‌بندی مناسبی برسد، ادامه می‌یابد. از این دسته الگوریتم‌ها می‌توان به Single-Link، Average-Link، اشاره کرد(Jain 2010). روش بالا به پایین همه

[1] Clustering

داده‌ها را در یک خوشه قرار می‌دهد و در هر تکرار این خوشه، تقسیم به خوشه‌های کوچکتر می‌شوند، این روند تا جایی که به یک خوشه‌بندی مناسبی برسد، ادامه می‌یابد.

۲-۳-۲- خوشه‌بندی مبتنی بر تراکم یا چگالی داده‌ها

این الگوریتم‌ها، براساس تراکم نمونه‌های همسایه، تقسیم می‌شوند. از جمله‌ی این الگوریتم‌ها DBSCAN، OPTICS ، DENCLUE می‌باشد (Jain 2010).

۲-۳-۳- خوشه‌بندی مبتنی بر گرید

در این نوع خوشه‌بندی فضای نمونه‌ها، به صورت سلول‌های هم اندازه تقسیم‌بندی می‌شوند. سپس خوشه‌بندی بر روی این ساختار انجام می‌گیرد. از مزیت اصلی این روش‌ها، می‌توان زمان پردازش را نام برد. معمولا این الگوریتم‌ها برای سرعت بخشیدن به الگوریتم‌های دیگر استفاده می‌شوند. از جمله الگوریتم‌های این روش، STING و CLIQUE می‌باشد.

۲-۳-۴- خوشه‌بندی مبتنی بر مدل

این روش برای هر خوشه مدلی در نظر می‌گیرد و هر نمونه برحسب شباهت به مدل، به خوشه مورد نظر تخصیص می‌یابد. در این روش خوشه‌بندی، از راهکارهای آماری و شبکه‌های عصبی استفاده می‌شود. از این نوع الگوریتم‌ها می‌توان EM[۱] (Dempster, Laird 1977) را نام برد.

GMM[۲] (Ben-Israel , Iyigun 2008) یکی از این روش‌ها است، هنگامی که فضای نمونه چندین توزیع آماری داشته باشد می‌توان وزن هر توزیع و پارامترهای آن را تخمین زد. این مسئله با EM حل می‌شود. شباهت داده‌ها نسبت به پارامتری توزیع‌های با استفاده از لگاریتم شباهت[۳] بدست می‌آید.

[1] Expectation maximum
[2] Gaussian Mixture Model
[3] Maximum Likelihood

۲-۳-۵- خوشه‌بندی افرازی

خوشه‌بندی افرازی (یا بر اساس تکرار)، ابتدا نقاطی را به عنوان نماینده هر کلاس در نظر می‌گیرد و سپس براساس یک معیار شباهت، هر داده را به یک خوشه تخصیص می‌دهد. این نوع خوشه‌بندی را می‌توان به دو گروه انحصاری[۱] و یا هم‌پوشی[۲] تقسیم کرد(Jain 2010). خوشه‌بندی انحصاری هر داده دقیقا به یک خوشه تعلق می‌گیرد مانند الگوریتم‌های K-means و K-medoids، اما خوشه‌بندی هم‌پوشی برای هر داده نسبت به همه خوشه‌ها درجه تعلقی در نظر می گیرد. این درجه تعلق به صورت فازی است. معروفترین الگوریتم از نوع خوشه‌بندی، الگوریتم FCM[۳] را می‌توان ذکر نمود(Wang1983). مشتقات زیادی براساس این الگوریتم‌ها ارائه شده است(Choi , Chung 2017) .

۲-۳-۶- مشتقات FCM

تابع هزینه الگوریتم FCM به صورت زیر می‌باشد:

$$\underset{u}{Max}\sum_{j=1}^{c}\sum_{i=1}^{n} u_{ij}^{m} \parallel x_i - v_j \parallel^2 \qquad \text{رابطه (۲-۱)}$$

که در آن U یک درجه عضویت، $V = \{v_1, \ldots, v_n\}$ مجموعه از c مرکز کلاس، $m > 1$ ثابت فازی‌سازی است. در FCM از فاصله اقلیدسی[۴] برای معیار شباهت استفاده می‌کند. U ماتریس عضویت است که می‌توان عکس فاصله بیان گردد یعنی اگر فاصله زیاد شود درجه تعلق کم می شود و اگر فاصله کم شود درجه تعلق زیاد می‌شود. از FCM الگوریتم‌های زیادی مشتق شده است.

یکی از این مشتقات FCM، Gustafson-Kessel (Gustafson , Kessel 1979) است، که در این روش برای معیار تشابه، از فاصله ماهالانوبیس[۵] استفاده شده است. رابطه (۲-۳) تابع هزینه می باشد.

---

[2] Exclusive or Hard Clustering
[3] Overlapping or Soft Clustering
[4] Fuzzy c-means (FCM)
[5] Euclidean
[5] Mahalanobis Distance

$$\underset{u}{Max}\sum_{j=1}^{c}\sum_{i=1}^{n}u_{ij}^{m}d(x_i,v_j)$$ رابطه (۲-۳)

فاصله از رابطه (۲-۴) و ماتریس A_i را با استفاده از رابطه (۲-۵) بدست می‌آید. ماتریس A_i بیانگر شکل خوشه است.

$$d(x_i,v_j)=(x_i-v_j)^T A_i(x_i-v_j)$$ رابطه (۲-۴)

$$A_i=\frac{\alpha|F_i|^{1/n}}{F_i}$$ رابطه (۲-۵)

$$F_i=\frac{\sum_{i=1}^{n}u_{ij}{}^{m}(x_i-v_j)(x_i-v_j)^T}{\sum_{i=1}^{n}u_{ij}{}^{m}}$$ رابطه (۲-۶)

در ادامه روش دیگری به نام Gath-Geva (Gath , Geva 1989) ارائه شد. رابطه (۲-۳) تابع هزینه می-باشد. فاصله از رابطه (۲-۳) بدست می‌آید.

$$d(x_i,v_j)=\frac{|F_i|^{1/2}}{\alpha_i}\exp\left(\frac{(x_i-v_j)^T F_i^{-1}(x_i-v_j)}{2}\right)$$ رابطه (۲-۷)

این روش نیز از دانش پیشین استفاده می‌کند. رابطه (۲-۸) نشان دهنده آن می‌باشد.

$$h(j\,|\,x_i)=\frac{1/d(x_i,v_j)}{\sum_{k=1}^{K}1/d(x_i,v_k)}$$ رابطه (۲-۸)

$$\alpha_i=\frac{1}{N}\sum_{j=1}^{N}h(i\,|\,x_j)$$ رابطه (۲-۹)

α_iاز دانش پیشین بدست می‌آید.

روش‌هایی نیز از الگوریتم FCM و Gath-Geva ایده گرفته‌اند، در این روش عدم شباهت FCM، یعنی فاصله هر داده تا مرکز را با شباهت احتمالی الگوریتم Gath-Geva، یعنی شباهت هر داده به هر خوشه، ترکیب شده است(Chatzis 2011).

۲-۳-۷- دانش‌پیشین

درصورتی که دانشی قبل از خوشه‌بندی داشته باشیم، تاثیر زیاد در روند خوشه‌بندی دارد. تاکنون استفاده از دانش‌پیشین در کارهای گذشته، به شکل استخراج دانش از خوشه‌بندی داده‌های اولیه به عنوان پارامترهای ورودی الگوریتم بوده است. الگوریتم‌های خوشه‌بندی مانند Kmeans و FCM نیز از دانش پیشین استفاده می‌کنند، تعداد کلاستر و همچنین نقاط اولیه خوشه‌بندی نیز بسیار مهم است. همچنین در الگوریتم Gath-Geva که از دانش‌پیشین برای مقداردهی اولیه به پارامترهای الگوریتم K-means استفاده می‌کند(Gath , Geva 1989) .

الگوریتم‌هایی که از دانش‌پیشین برای خوشه‌بندی استفاده می‌نمایند، داده‌هایی که باید در یک خوشه باشند را به صورت قید must-linked و داده‌هایی که نباید در یک خوشه باشند را به صورت cannot-linked قرار می‌دهند(Klein 2002)، که به این نوع قیدها در سطح شی گفته می‌شودکه با استفاده از این نوع قیدها روش Kmeans را بهبود داده است. همچنین (Wagstaff, Cardie 2001) نیز قید اندازه خوشه، به عنوان دانش پیشین به الگوریتم Kmeans اضافه کرده و خوشه‌بندی را بهبود داده است.

۲-۴- انواع خوشه‌بندی جریان‌داده

برخی الگوریتم‌های خوشه‌بندی جریان‌داده، تعمیم یافته روش‌های سنتی هستند (Bhatnagar, Kaur 2014)(Bhatnagar 2014)، که می‌توانند به سه گروه زیر تقسیم‌بندی شوند:

۲-۴-۱- الگوریتم‌هایی براساس فاصله و چگالی

در روند این الگوریتم‌ها، از یک مجموعه‌ای از نقاط اولیه استفاده می‌شود، معیار شباهت، فاصله یا چگالی است. داده به مجموعه‌های کوچکی به نام خوشه‌های میکرو تقسیم می‌شود و نقاط اولیه از این مجموعه‌ها بدست می‌آید. مجموعه نقاط اولیه، به عنوان نماینده‌ی هر خوشه در نظر گرفته می‌شود. سپس با استفاده از آن کل داده خوشه‌بندی می‌شوند.

در الگوریتم‌های مبتنی بر فاصله، شکل خوشه‌ها منحنی است و در مقابل نویز قوی نیستند. در حالی که الگوریتم‌های مبتنی بر چگالی شکل خوشه‌ی مشخصی ندارند. در این مورد الگوریتم‌های CluStream(Aggarwal , Han 2003) و DenStream (Ruiz , Menasalvas 2009) و C-denStream(Ruiz , Menasalvas 2009) را می‌توان نام برد.

۲-۴-۲- روش‌های مبتنی بر شبکه

محبوبیت این روش‌ها به دلیل پیچیدگی محاسباتی کم آن است. این روش به دو نوع FGG[1] (Aldinucci Coppola 2006 , وDGG (Lee, Lee 2008) تقسیم‌بندی می‌شود. FGG توسط کاربر اندازه سلول‌ها را مشخص می‌کند، اما در DGG اندازه سلول‌ها براساس ویژگی داده‌های فعلی است. DGG قادر است ساختار بهتری از داده‌ها ارائه دهد، در حالی که حافظه‌ی بالاتر و هزینه‌ی محاسباتی بیشتری در مقایسه با FGG تحمیل می‌کند.

۲-۴-۳- روش مبتنی بر آمار

روش‌های آماری به دو دسته پارامتری و غیرپارامتری تقسیم می‌شوند. اخیرا این روش‌ها برای خوشه‌بندی جریان‌داده‌ها مورد توجه قرار گرفته‌اند (Dang, Lee 2009) (Song,Wang 2004). روش‌های خوشه-

[1] Future Generation Grids

بندی مبتنی بر پارامتر نیازمند فرضیات اولیه برای برآورد احتمالات ناشناخته و توابع چگالی خود می-باشد، درحالی‌که روش‌های بدون‌پارامتر نیازمند فرضیات اولیه نیست.

۲-۴-۴- خوشه‌بندی افزایشی

به دلیل افزایش تدریجی داده‌ها و محدودیت حافظه و همچنین قدرت پردازش، الگوریتم‌های زیادی ارائه شده‌اند. این الگوریتم‌ها در طول زمان عمر خود، در حال یادگیری و همواره در حال توسعه مرزهای تصمیم‌گیری می‌باشند و همچنین تغییر مفهوم[1] را به همراه دارند.

دو روش برای خوشه‌بندی افزایشی وجود دارد(Ade, Deshmukh 2013):

۱- روش تجمیع داده‌ها

۲- روش یادگیری گروهی

از جمله الگوریتم‌های سنتی تعمیم یافته می‌توان kmeans (Chakraborty, Nagwani 2011) ، DBSCAN(Chakraborty, Nagwani 2014) که در این روش‌ها، الگوریتم‌های سنتی را به صورت افزایشی ارائه داده‌اند و همچنین خوشه‌بندی فازی افزایشی(Aghabozorgi, Saybani 2012) (Wang, Chen 2014) و GMM (Engel, Heinen 2010) را نام برد.

یکی از روش‌های خوشه‌بندی افزایشی، برحسب مدل هسته‌ی طیفی افزایشی[2] (IKSC) می‌باشد که این الگوریتم مدلش را در طول زمان تغییر می‌کند (Langone, Agudelo 2014).

۲-۵- محاسبات خوشه‌ای

یکی از روش‌های پردازش موازی محاسبات خوشه‌ای است. محاسبات روی چندین خوشه[1] یا ماشین-های مختلف انجام می‌شود. این محاسبات طوری توزیع می‌شوند که کاربر تصور می‌کند روی یک ماشین انجام شده است(Morrison 2003).

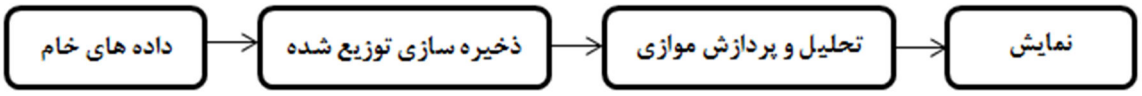
[1] Concepts Drift
[2] Incremental kernel spectral clustering

از مزایای محاسبات خوشه‌ای می‌توان موارد ذیل را نام برد:

- ذخیره‌سازی خوشه‌ای[2]:در این تکنیک، فضای فیزیکی به صورت توزیع شده روی ماشین‌های مختلفی می-باشد و با بخش واحد، مدیریت می‌شود (Yeo, Buyya 2006).
- دسترسی بالا[3] : اگر ماشینی از سیستم خارج شود، ماشین‌های دیگر جایگزین آن می‌شوند و کاربر متوجه خارج شدن ماشین از سیستم نمی‌شود. در این صورت علاوه بر دسترسی با سرعت بالاتری انجام می‌شود، قابلیت اطمینان دسترسی نیز افزایش می‌یابد.
- تقسیم‌بندی بار[4] : بار محاسباتی و پردازشی و ذخیره سازی روی همه ماشین‌ها توزیع می‌شود و در نتیجه توازن بار برقرار است.
- بالا بردن کارایی[5] : پردازش‌های موازی باعث بالا رفتن کارایی سیستم می‌شوند. به دلیل توزیع بار محاسباتی و استفاده حداکثر از منابع ماشین‌ها، پردازش در زمان کمتری انجام می‌شود.

۲-۶- ابزارهای محاسبات خوشه‌ای

سیستم‌های محاسبات خوشه‌ای از دو قسمت اصلی ذخیره‌سازی توزیع شده و تحلیل و پردازش موازی و توزیع شده تشکیل شده‌اند که روند اجرای پردازش موازی در شکل (۲-۱) نشان داده شده است.



شکل(۲-۱)- روند اجرای پردازش موازی

ابزارهای مختلفی برای محاسبات خوشه‌ای بوجود آمده است، در این بخش چندین نمونه از آن شرح داده می‌شود:

---

[1] cluster
[2] Storage Clusters
[3] High Availability
[4] Load Balancing
[5] High Performance

۲-۶-۱- ابزارهای پردازش کلان‌داده

بعضی از ابزارها در جدول(۲-۱) توضیحاتی داده شده است (Kamburugamuve, Fox 2013).

جدول(۲-۱)- ابزارهای محاسبات خوشه‌ای

آنالیز داده	**Mahout**	mahout چارچوب نرم‌افزاری برای پردازش‌های موازی است. این ابزار، برای مقیاس‌پذیری و تجاری کردن تکنیک‌های یادگیری ماشین جهت تحلیل داده‌ها در مقیاس بزرگ و هوشمند انجام می‌گیرد. که در این ابزار الگوریتم‌های یادگیری ماشین به‌خوبی قابل اجرا است.
پردازش داده	**Apache Spark**	یک چارچوب محاسباتی برای کلان‌داده و جریان‌داده است.
	Storm	یکی دیگر از ابزارهای محاسبات خوشه‌ای برای پردازش توزیع شده است. علاوه بر این، مقیاس‌پذیری و تحمل خطا برای اجرای برنامه‌های رقابتی وجود دارد.
NOSql	**Hbase**	ابزاری برای پایگاه داده Nosql است که در کنار هدوپ استفاده می شود. دسترسی به داده در هدوپ به صورت ترتیبی است. در این ابزار به صورت تصادفی انجام می‌گیرد و به همین دلیل سرعت بهتری دارد.
	Cansedra	این ابزار سیستم توزیع شده‌ی کوچک و جامعی است که ارتباط داخلی آن به صورت نقطه به نقطه[1] است.
	Accumula	این ابزار در کنار Hbase استفاده می‌شود و جهت سطح دسترسی کاربران مورد استفاده قرار می‌گیرد.
هماهنگ کننده	**Zookeeper**	این ابزار برای مدیریت نقل و انتقالات و ارتباط بین کار[1]ها مورد

[1] pair to pair

توزیعات		استفاده می‌شود.
خلاصه‌سازی سطح بالا	**Hive**	انباره داده‌ای است که با دستورات Hql می‌توان پرس‌وجو روی داده‌ها را انجام داد.
	Pig	ابزاری برای پیش‌پردازش است که هر نوع داده‌ای به هر فرمتی را تبدیل می‌کند.

- **هدوپ**[۲]

هدوپ یک چارچوب نرم‌افزاری متن‌باز، برای برنامه‌های توزیع شده است که از دو بخش اصلی فایل سیستم توزیع شده به نام HDFS[۳] (فایل سیستم توزیع شده هدوپ) و همچنین سیستم پردازش موازی با استفاده از روش نگاشت-کاهش[۴] تشکیل شده است(Xu, Liang 2012). هدوپ برای اجرای کار[۵] روی سیستم، برنامه‌ها را روی ماشین‌ها توزیع می‌کند و بعد از پردازش داده‌ها نتایج را جمع‌آوری می‌کند و سپس در خروجی نمایش می‌دهد.

روش نگاشت-کاهش که امکان پردازش موازی را می‌دهد، از دو قسمت اصلی تشکیل شده است:

گام نگاشت[۶]: گره اصلی[۷] علمیات یک برنامه را به قسمت‌های کوچکتری تقسیم می‌کند و سپس نتایج را به سرور اصلی می‌فرستد.

گام کاهش[۸]: عملیات خروجی به وسیله گره اصلی بر روی نتایجی که از گره‌های کار بدست آمده، انجام می‌شود و آنها را نمایش می‌دهد.

---

[1] job
[2] Hadoop
[3] Hadoop Distribition File System
[4] Map-Reduce
[5] job
[6] map
[7] Master Node
[8] reduce

۲-۶-۲- آپاچی اسپارک[1]

یک چارچوب محاسباتی برای کلان‌داده و جریان‌داده است. بزرگترین برتری آن این است که مجموعه داده را حین پردازش در حافظه نگه می‌دارد(Karau, Konwinski 2015) (Zaharia 2016). به همین دلیل نسبت به هدوپ، حدود ۱۰۰ برابر کارایی بهتری دارد. این ابزار به خوبی با هدوپ و یارن[2] کار می-کند و نیز می‌تواند از تکنیک نگاشت- کاهش[3] بهره ببرد. در ابزارهای محاسبات خوشه‌ای دیگر مانند هدوپ، برای هر یک از پردازش‌ها مانند sql و یادگیری ماشین و...، یک موتور پردازشی مجزا وجود دارد ولی یکی از مزیت‌های اسپارک این است که همه این موتورها را در یک موتور عمومی ترکیب کرده و کار را برای کاربر آسان نموده است. یکی دیگر از محبوبیت‌های اسپارک اجرای الگوریتم‌های تکرار شونده، مانند kmeans و تحلیل تعاملی می‌باشد. در الگوریتم تحلیل تعاملی، کاربر مجموعه‌ای از پرس‌وجوهای اکتشافی را روی مجموعه داده انجام می‌دهد. اسپارک محیطی برای کاربران فراهم کرده است که به راحتی بتوانند با استفاده از چند زبان اسکالا، جاوا و پایتون کدنویسی کنند، که یکی از مزیت‌های آن محسوب می‌شود. اسپارک شامل ماژول‌هایی است که شکل(۲-۳) آنها را نشان می‌دهد.

شکل(۲-۳) – اجزاهای اسپارک ((Karau, Konwinski 2015) صفحه ۳)

[1] Apache Spark
[2] yarn

۲-۶-۳- هسته اسپارک[1]

مسئولیت‌های اصلی هسته اسپارک زمانبندی وظیفه، مدیریت حافظه، رفع خطا و ارتباط با سیستم-های ذخیره‌سازی با دیگر اجزای اسپارک می‌باشد. مجموعه داده‌ی توزیع شده[2]، اصلی‌ترین قسمت اسپارک است که دو وظیفه مهم توزیع محاسبات روی گره‌ها و پردازش موازی روی اسپارک را برعهده دارد.

۲-۶-۴- زبان پرس و جو اسپارک[3]

این قسمت برای پرس‌وجو روی داده‌های ساختار یافته است که براساس hive انجام می‌شود.

۲-۶-۵- پردازش جریان داده[4]

یکی از محبوبت‌های اسپارک به دلیل وجود همزمانِ پردازش داده با ورود داده می‌باشد داده‌هایی که در طول زمان وارد سیستم می‌شوند تا پردازش بر روی آنها صورت گیرد، Dstream[5] می‌باشد. شکل شکل(۲-۴) یک نمونه از Dstream پنجره‌گذاری شده است. همانطور که نشان داده می‌شود این داده‌ها، در طی زمان‌های مشخص، پنجره‌گذاری شده‌اند.

شکل (۲-۴) – Dstream های پنجره‌گذاری شده ((Karau, Konwinski 2015 صفحه ۱۸۷)

[1] SparkCore
[2] Resilient Distributed Datasets(RDD)
[3] Sql Shark
[1] Streaming Spark
[5] Discretized Streams

۲-۶-۶- کتابخانه یادگیری ماشین[1]

یکی از اجزا اسپارک کتابخانه‌هایی هستند که برای روش‌های یادگیری ماشین مانند الگوریتم‌های رگرسیون و خوشه‌بندی و طبقه‌بندی استفاده می‌شوند و این الگوریتم‌ها به صورت موازی و توزیع شده پیاده‌سازی شده‌اند.

۲-۶-۷- پردازش گراف[2]

یکی دیگر از اجزا اسپارک کتابخانه‌هایی برای پردازش‌های گراف و عملگرهای روی گراف می‌باشند که به صورت پردازش موازی و توزیع شده می‌باشد.

۲-۶-۸- مدیریت خوشه

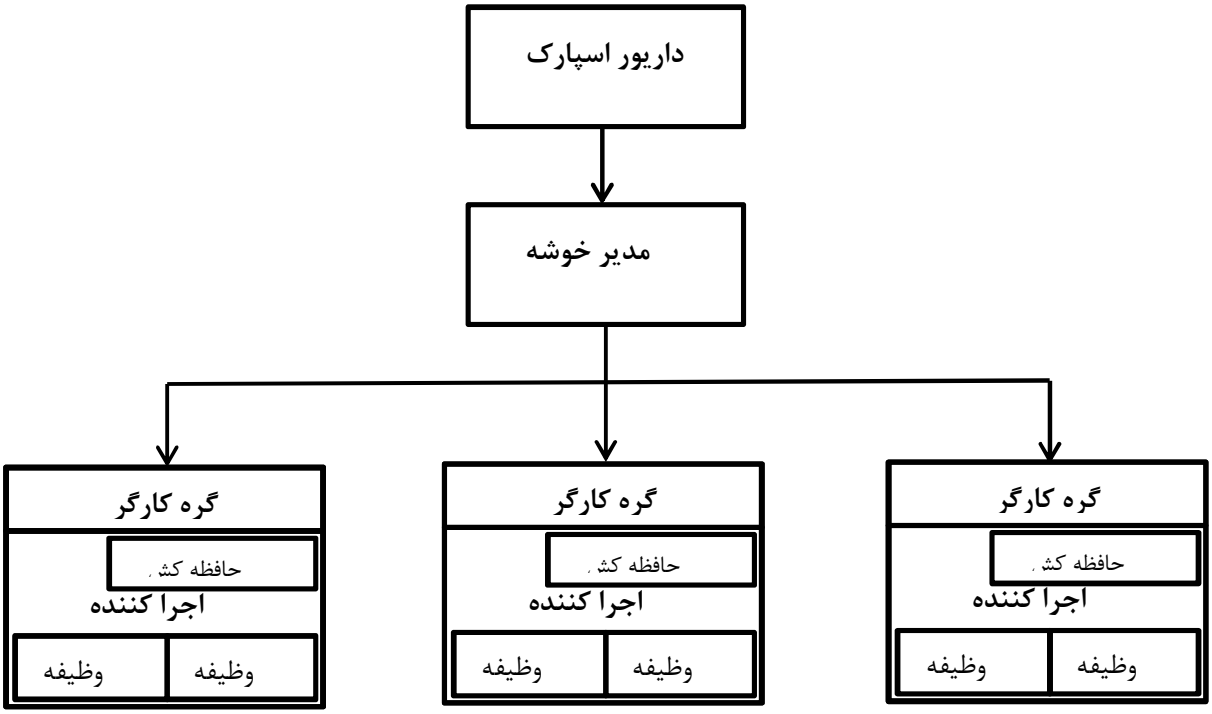
این بخش از اسپارک برای مدیریت خوشه‌های محاسباتی است. مدیریت خوشه‌ها می‌تواند به سه روش یارن، Apache Mesos و یا با استفاده از زمانبندی خود اسپارک صورت گیرد.

۲-۶-۹- معماری

اسپارک شامل دو قسمت مدیر[3] و درایور[4] و اجراکننده (گره کارگر) می‌باشد. شکل (۲-۵) معماری اسپارک را نشان می‌دهد.

---

[1] MLib
[4] GraphX
[3] master
[4] driver

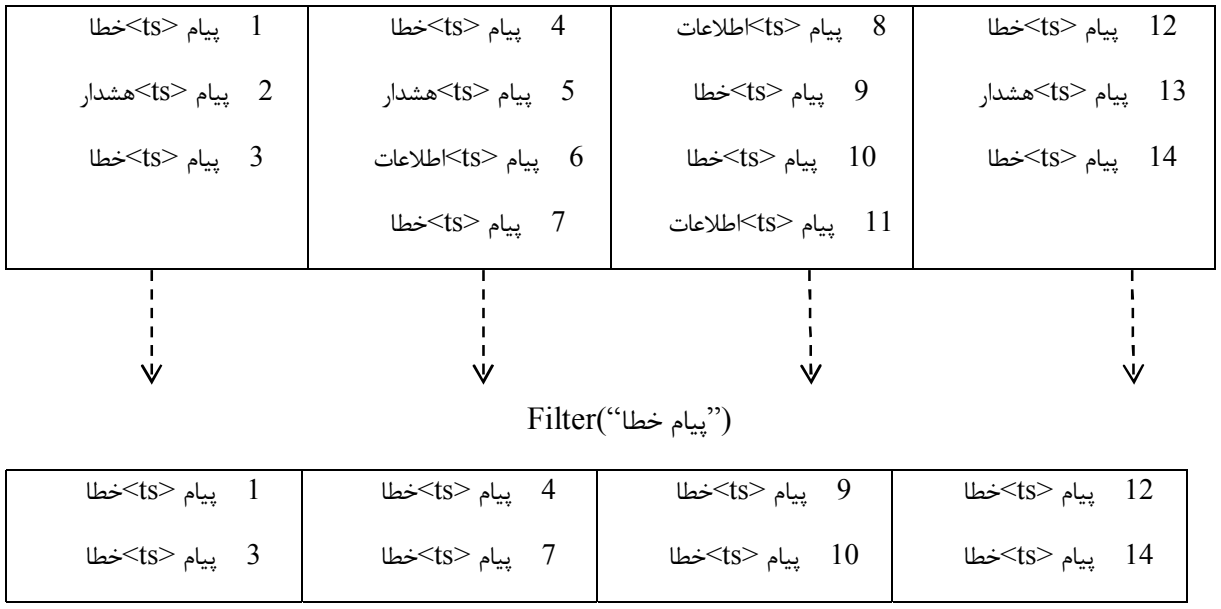


شکل(۲-۵) – معماری اسپارک ((Karau, Konwinski 2015 صفحه ۱۵)

این بخش مدیریت خوشه‌ها را برعهده دارد. در قسمت درایور اسپارک، شی‌ای به نام sc از جنس SparkContext وجود دارد. وظیفه این شی اتصال برنامه به بخش مدیریت خوشه‌های اسپارک می‌باشد.

وظیفه اصلی مدیر خوشه تبدیل برنامه به واحدهای کوچکتر است، برنامه به RDD هایی تقسیم می-شود که روی این RDD می‌توان دستورات تبدیلی و اعمالی چون collect و count انجام داد. دستوراتی مانند map یا فیلترکردن داده‌ها روی RDD هستند. در این صورت داده را بر روی پارتیشن‌ها نگاشت می-شود. هر RDD می‌توان یک پارتیشن یا چند پارتیشن باشد. در شکل(۲-۶) یک RDD با ۴ پارتیشن نشان داده شده است که متن‌هایی که کلمه "پیام خطا" در آن باشد را فیلتر می‌کند و نتیجه را در یک RDD دیگر نشان می‌دهد.

مدیر خوشه برای انجام کار، گرافی بدون دور جهت‌دار منطقی از عملیات را ایجاد می‌کند. زمانی که راه‌انداز شروع به اجرا می‌شود، این گراف منطقی به یک طرح اجرای فیزیکی تبدیل می‌شود.



شکل (۲-۶) – فیلتر کردن "پیام خطا" ((Karau, Konwinski 2015) صفحه ۱۸۷)

۳- فصل سوم

روش پیشنهادی

۳-۱- مقدمه

در مورد جریان‌داده در فصل‌های گذشته توضیح داده شد، جریان‌داده در طول زمان به صورت تدریجی تولید می‌شود. یکی از راهکارهای خوشه‌بندی این داده‌ها، این است که محور زمان به بازه‌های مشخصی تقسیم شود، به این ترتیب داده‌ها بلوک‌بندی خواهد شد.

روش‌های خوشه‌بندی‌های زیادی در این زمینه ارائه شده است. در این فصل از پایان‌نامه روشی ارائه خواهد شد که مبتنی بر مدل و افزار می‌باشد.

مطابق مطالب بیان شده در بخش (۲-۳-۴) فصل ۲، الگوریتم‌های مبتنی بر مدل برای هر خوشه یک مدلی در نظر می‌گیرد و هر نمونه برحسب شباهت به مدل، به خوشه مورد نظر تخصیص می‌یابد.

همانطور که در بخش (۲-۳-۵) فصل ۲ خوشه‌بندی افراز بیان شد، در هر افراز، نمونه‌ها به طور قطعی به یک خوشه تخصیص داده می‌شوند. هر خوشه یک نماینده دارد، این نماینده می‌تواند مرکز خوشه باشد. در هر افراز، هر نقطه به نزدیکترین نماینده‌ی خوشه تخصیص داده می‌شود. نماینده هر خوشه دوباره محاسبه و بروزرسانی می‌شود و ممکن است نماینده خوشه تغییر کند. سپس دوباره نمونه‌ها به خوشه‌ها تخصیص داده می‌شود.

الگوریتم ارائه شده به طور کلی از دو بخش تشکیل شده است. در بخش اول نمونه‌ها به صورت یکجا وجود دارند، و برای هر خوشه مدلی در نظر می‌گیرد و میزان شباهت هر نمونه به پارامترهای مدل خوشه تعیین می‌گردد. در بخش دوم الگوریتم پیشنهادی برای جریان‌داده توسعه داده خواهد شد، با ورود هر نمونه پارامترهای هر خوشه جدید پارامترهای مدل خوشه‌ها محاسبه و بروزرسانی می‌شود.

۳-۲- خوشه‌بندی

مطابق مطالب بیان شده در بخش (۲-۴) فصل ۲، یک خوشه، مجموعه‌ای از نمونه $X = \{x_1, x_2, ..., x_n \mid x_i \in \mathbb{R}^d\}$ است که از بعضی جهات شباهت دارند.

هر نمونه‌ی x_i، $1 < i < n$، d بعدی است. معیار شباهت بین نمونه‌ها ممکن است "حداقل فاصله" باشد، $d(x, y)$ فاصله بین نمونه‌های x و y است(Ben-Israel, Iyigun 2008). اگرمعیار شباهت فاصله اقلیدسی درنظر گرفته شود، فاصله بین نمونه‌ها $d(x,y) = \| x - y \|, \forall x, y \in \mathbb{R}^n$ می‌باشد. این فاصله را می‌توان با فاصله ماهالانوبیس[۱] و روش‌های دیگر نیز بدست آید.

یکی از چالش‌های مهم در خوشه‌بندی نمونه‌ها، تعیین تعداد خوشه‌هاست. تعداد خوشه‌ها را با c نشان می‌دهند.

همانطور که در بخش (۲-۳-۵) فصل ۲ خوشه‌بندی افراز بیان شد، در هر افراز، نمونه به طور قطعی به یک خوشه تخصیص داده می‌شود. هر خوشه یک نماینده دارد، این نماینده می‌تواند مرکز خوشه باشد. در هر افراز، هر نقطه به نزدیکترین نماینده‌ی خوشه تخصیص داده می‌شود. نماینده‌ی هر خوشه دوباره محاسبه و بروزرسانی می‌شود و ممکن است نماینده خوشه تغییر کند. سپس دوباره نمونه‌ها به خوشه‌ها تخصیص داده می‌شود.

رابطه‌ی بین عضویت نمونه به هر خوشه و فاصله‌ی نمونه تا نماینده‌ی خوشه را می‌توان به صورت $p_j(x) d_j(x,v)$ در نظر گرفت. $d_j(x,v)$ فاصله نقطه x از v نماینده‌ی خوشه j ام ، $p_j(x)$ درجه‌ی عضویت نمونه x به خوشه j ام است. درجه عضویت عددی بین ۰ و ۱ است. در بعضی الگوریتم‌ها، این درجه عضویت را با u نمایش می‌دهند. با توجه به رابطه‌ی (۳-۱)، جمع همه‌ی u_{ij} برای همه خوشه‌ها برابر ۱ می‌باشد.

---

[۱] Mahalanobis ***Distance***

رابطه (۳-۱)

$$\sum_{j=1}^{c} u_{ij} = 1$$

در این پژوهش، فاصله‌ی نمونه تا نماینده خوشه را به صورت تابع احتمالی خواهد شد.

۳-۳- الگوریتم پیشنهادی

۳-۳-۱- رویکرد کلی الگوریتم

همانطور که در فصل قبل گفته شد، خوشه‌بندی پیداکردن نمونه‌هایی که از بعضی جهات بهم شبیه هستند، نمونه‌های یک خوشه می‌توانند با یک پارامتر همسان توصیف می‌شوند. الگوریتم پیشنهادی مبتنی بر مدل است. در این روش بجای نگهداری خود نمونه‌ها، تنها پارامترهای مدل آن نگه‌داری می‌شود. مدل خوشه‌بندی پیشنهادی هر توزیع گوسی را به عنوان یک خوشه در نظر می‌گیرد، که پارامترهای آن میانگین و کواریانس نمونه‌ها می‌باشد. ایده‌ی اصلی این الگوریتم محاسبه شباهت نمونه به پارامترهای هرخوشه و اضافه کردن دانش-پیشین به تابع هزینه می‌باشد.

- **خوشه‌بندی**

$X = \{x_1, x_2, ..., x_n \mid x_i \in \mathbb{R}^d\}$ است که از بعضی جهات شباهت دارند. هر نمونه‌ی x_i، $1 < i < n$، d بعدی است. نمونه‌های درون خوشه با X_j به صورت $X_j = \{x_1, x_2, ..., x_{n_j} \mid x_i \subset \mathbb{R}^d\}$، که d ابعاد نمونه است، نشان داده می‌شود، در این صورت اگر داده ها به c خوشه نا همپوشان تقسیم شوند، پارامترهای خوشه j ام را θ_j می-نامند. به‌عبارتی هر خوشه با پارامترهای آن توصیف می‌شوند، می‌توان آن را به صورت رابطه (۳-۲) نشان داد:

رابطه (۳-۲)

$$f(\theta_j \mid X_j)$$

تابع $f(\theta_j \mid X_j)$ را می‌توان تابع شباهت θ_j به X_j دانست ، در اینصورت رابطه (۳-۳) تابع هزینه‌ی شباهت بین نمونه‌های X_j و پارامترهای خوشه θ_j را می‌باشد.

$$Max_{\theta_j} \sum_{j=1}^{c} \mathrm{f}(\theta_j | X_j)$$ رابطه (۳-۳)

اگر نمونه‌های هر خوشه بصورت X_j ها جدا شوند، پیدا کردن آن آسان است اما بدست آوردن نمونه‌های متعلق به خوشه X_j کار ساده‌ای نیست، با توجه به مطالب گفته شده، تابع هزینه جدیدی تشکیل می‌شود.

۳-۳-۲- تابع هزینه

هر نمونه به صورت x_i نشان داده می‌شود، n تعدادکل نمونه‌ها می‌باشد. تابع هزینه رابطه (۳-۴) با حداکثر کردن شباهت هر نمونه به پارامتر θ_j بدست می‌آید.

$$\underset{\theta_j, u}{Max} \sum_{j=1}^{c} \sum_{i=1}^{n} u_{ij}^{m} f(\theta_j | x_i)$$ رابطه (۳-۴)

این الگوریتم، زیر مجموعه الگوریتم‌های تکراری است، با این تفاوت که بجای $d(x,v)$ از تابع شباهت $f(\theta_j | x_i)$ استفاده می‌کند. متغیر u_{ij} احتمال عضویت هر نمونه x_i به خوشه j ام است و این مقدار بین $0 \le u_{ij} \le 1$ می‌باشد که در آن m متغییر فازی‌سازی است.

$f(\theta_j | x_i)$ تابع احتمال شرطی که احتمال اتفاق افتادن θ_j به شرط x_i است ، این تابع احتمال شباهت نمونه به پارامترهای هر خوشه را بیان می‌کند. این تابع را می‌توان به صورت رابطه (۳-۵) نیز نشان داد:

$$\sum_{j=1}^{c} \sum_{i=1}^{n} u_{ij}^{m} f(x_i | \theta_j) f(\theta_j)$$ رابطه(۳-۵)

$f(x_i | \theta_j)$ تابع احتمال نمونه x_i به شرط پارامتر θ_j است، شباهت پارامتر θ_j به هر نمونه x_i را بیان می‌کند، به این تابع، تابع شباهت گفته می‌شود و $f(\theta_j)$ تابع احتمال پیشین که از اطلاعات قبلی بدست می‌آید. بدیهی است که هر چقدر $f(\theta_j)$ دقیقتر و قوی‌تر باشد، نتیجه خوشه‌بندی دقیق‌تر است.

همانطور که گفته شد، در این الگوریتم توزیع نمونه‌های خوشه، یک توزیع گوسی با پارامترهای میانگین نمونه‌ها (μ) و کواریانس آن‌ها (Σ) می‌باشد.

تابع شباهت به صورت رابطه (۳-۶) است:

$$f(x_i \mid \theta_j) = \frac{1}{(2\pi)^{d/2} \mid \Sigma_j \mid^{1/2}} \exp(-0.5(x_i - \mu_j)^T \Sigma_j^{-1}(x_i - \mu_j)) \qquad \text{رابطه(۳-۶)}$$

در رابطه (۳-۶) پارامترهای μ، میانگین و Σ، کواریانس خوشه j ام تابع شباهت است. در این رابطه می توان تعلق هرنمونه به هر خوشه را بدست آورد. تابع دانش پیشین برای هر خوشه به صورت زیر خواهد بود.

$$f(\theta_j) = N(\mu_{\theta_j}, \Sigma_{\theta_j})$$

$$= \frac{1}{(2\pi)^{d/2} \mid \Sigma_{\theta_j} \mid^{1/2}} (-0.5(\mu_j - \mu_{\theta_j})^T \Sigma_j^{-1}(\mu_j - \mu_{\theta_j})) \qquad \text{رابطه(۳-۷)}$$

در رابطه (۳-۷)پارامترهای μ_{θ_j}، میانگین و Σ_{θ_j}، کواریانس خوشه jام توزیع احتمال پیشین است. با جایگزین کردن تابع چگالی احتمال رابطه‌های (۳-۶) و (۳-۷) در رابطه (۳-۵)، رابطه (۳-۸) بدست می‌آید:

$$\sum_{j=1}^{c}\sum_{i=1}^{n} u_{ij}^m \ \mathrm{N}\left(\mu_j, \Sigma_j\right)\mathrm{N}\left(\mu_{\theta_j}, \Sigma_{\theta_j}\right) =$$

$$\frac{1}{(2\pi)^{d/2} \mid \Sigma_j \mid^{1/2} \mid \Sigma_{\theta_j} \mid^{1/2}} \qquad \text{رابطه (۳-۸)}$$

$$\exp(-0.5((x_i - \mu_j)^T \Sigma_j^{-1}(x_i - \mu_j) + (\mu_j - \mu_{\theta_j})^T \Sigma_j^{-1}(\mu_j - \mu_{\theta_j})))$$

رابطه (۳-۸) تابع هدف مسئله می‌باشد. با کمینه کردن این تابع، کمترین هزینه برای خوشه‌بندی را خواهیم داشت.

برای حل این مسئله، از روش گرادیان نزولی استفاده می‌کنیم و تابع لاگرانژ آن را تشکیل می‌دهیم و نیز محدودیتی در رابطه (۳-۱) اضافه می‌کنیم و پارامترهای آن را حداقل می‌کنیم. رابطه (۳-۹) تابع لاگرانژ می-باشد:

$$L = \frac{u_{ij}^{m}}{(2\pi)^{d/2} \mid \Sigma_j \mid^{1/2} \mid \Sigma_{\theta_j} \mid^{1/2}} \exp(-0.5((x_i - \mu_j)^T \Sigma_j^{-1}(x_i - \mu_j) + (\mu_j - \mu_{\theta_j})^T \Sigma_j^{-1}(\mu_j - \mu_{\theta_j}))) - \sum_{i=1}^{n} \lambda_i (\sum_{j=1}^{c} u_{ij} - 1)$$

رابطه(۳-۹)

۳-۳-۳- بدست آوردن پارامترهای خوشه

– میانگین هر خوشه

مکان توزیع با میانگین بیان می‌شود. در الگوریتم پیشنهادی میانگین خوشه، نماینده‌ی خوشه را بیان می-کند.

برای حداقل کردن تابع هزینه، نسبت به پارامترهای توزیع مشتق گرفته می‌شود.

$$\frac{\partial L}{\partial \mu_j} = 0$$

$$\sum_{i=1}^{n} \frac{u_{ij}^{m}}{(2\pi)^{d/2} \mid \Sigma_j \mid^{1/2} \mid \Sigma_{\theta_j} \mid^{1/2}} ((-0.5((-2)\Sigma_j^{-1}(x_i - \mu_j)) + (-0.5((2)\Sigma_{\theta_j}^{-1}(\mu_j - \mu_{\theta_j}))) \exp(-0.5((x_i - \mu_j)^T \Sigma_j^{-1}(x_i - \mu_j) + (\mu_j - \mu_{\theta_j})^T \Sigma_j^{-1}(\mu_j - \mu_{\theta_j})))$$

رابطه (۳-۱۱)

در رابطه (۳-۱۲)، متغیر کمکی f_{ij} استفاده می‌شود تا روابط نهایی ساده‌تر بیان شوند.

$$f_{ij} = \exp(-0.5((x_i - \mu_j)^T \Sigma_j^{-1}(x_i - \mu_j) + (\mu_j - \mu_{\theta_j})^T \Sigma_j^{-1}(\mu_j - \mu_{\theta_j})))$$

رابطه (۳-۱۲)

با جایگذاری رابطه‌ی(۳-۱۲) در (۳-۱۱) می توان محاسبه μ_j راحتر باشد.

$$\mu_j = \left(\frac{\Sigma_j^{-1}}{\Sigma_j^{-1} + \Sigma_{\theta_j}^{-1}} \times \frac{\sum_{i=1}^{n} u_{ij}^m f_{ij} x_i}{\sum_{i=1}^{n} u_{ij}^m f_{ij}} \right) + \left(\frac{\Sigma_{\theta_j}^{-1}}{\Sigma_j^{-1} + \Sigma_{\theta_j}^{-1}} \times \frac{\sum_{i=1}^{n} u_{ij}^m f_{ij} \mu_{\theta_j}}{\sum_{i=1}^{n} u_{ij}^m f_{ij}} \right) \qquad \text{رابطه (۳-۱۲)}$$

در رابطه(۳-۱۴) متغیر کمکیS در نظر گرفته می‌شود:

$$S = \frac{\Sigma_j^{-1}}{\Sigma_j^{-1} + \Sigma_{\theta_j}^{-1}} \qquad \text{رابطه (۳-۱۴)}$$

همچین 1-S نیز به صورت زیر بدست می‌آید:

$$1 - S = \frac{\Sigma_{\theta_j}^{-1}}{\Sigma_j^{-1} + \Sigma_{\theta_j}^{-1}} \qquad \text{رابطه (۳-۱۵)}$$

رابطه (۳-۱۵) با جایگزین کردن S و 1-S در رابطه (۳-۱۴)، رابطه (۳-۱۶) بدست می‌آید:

$$\mu_j = \left((S) \times \frac{\sum_{i=1}^{n} u_{ij}^m f_{ij} x_i}{\sum_{i=1}^{n} u_{ij}^m f_{ij}} \right) + \left((1-S) \times \frac{\sum_{i=1}^{n} u_{ij}^m f_{ij} \mu_j}{\sum_{i=1}^{n} u_{ij}^m f_{ij}} \right) \qquad \text{رابطه(۳-۱۶)}$$

جمع دو ضریب S و 1-S، یک می‌شود. در این رابطه اگر S به یک نزدیکتر باشد، ارزش نمونه‌های جدید بیشتر خواهد شد و اگر به صفر نزدیک باشد، ارزش دانش پیشین بیشتر است. اگر از جنبه احتمالاتی به این ضرایب نگاه شود، می‌توان به پراکندگی بین نمونه‌های جدید و دانش‌پیشین پی‌برد.

- **کواریانس داده های تخمینی**

پارامتر دیگری که تابع توزیع را بیان می‌کند، کواریانس نمونه‌ها است. این پارامتر پراکنندگی نمونه‌ها و شکل خوشه را نشان می‌دهد. ماتریس کواریانس به صورت مربعی است که ابعاد آن برابر ابعاد نمونه‌ها است، قطر اصلی آن نشان دهنده‌ی پراکندگی افقی و قطر فرعی پراکندگی نمونه در بعد عمودی را نشان می‌دهد، این ماتریس بیانگر شکل خوشه است.

در این بخش مشتق تابع لاگرانژ براساس ماتریس کواریانس تخمینی روی کل نمونه‌های Σ_j گرفته می-شود. ماتریس کواریانس به صورت زیر است:

$$\Sigma_j = \frac{\sum_{i=1}^{n} u_{ij}^m \left(x_i - \mu_j\right)\left(x_i - \mu_j\right)^T f_{ij}}{\sum_{i=1}^{n} u_{ij}^m f_{ij}}$$ رابطه(۳-۱۷)

در رابطه(۳-۱۷)، $\sum_{i=1}^{n} u_{ij}^m f_{ij}$ نرمال‌کننده است.

– تابع عضویت

u_{ij} درجه عضویت که بیانگر میزان عضویت نمونه x_i به توزیع چگالی خوشه است. اگر درجه عضویت یک نمونه از توزیع چگالی خوشه برابر با صفر باشد، x_i کاملاً از توزیع خوشه خارج است و اگر درجه عضویت x_i برابر با یک باشد، x_i کاملاً در خوشه قرار دارد. اگر درجه عضویت یک نمونه ما بین صفر و یک باشد، این عدد بیانگر درجه عضویت تدریجی است.

در این مرحله ، از تابع لاگرانژ نسبت به ماتریس عضویت u_{ij} مشتق گرفته می‌شود. اثبات آن به صورت روابط (۳-۱۸)، (۳-۱۹)، (۳-۲۰) است:

$$\frac{\partial L}{\partial u_{ij}} = 0$$ رابطه (۳-۱۸)

$$\frac{\partial L}{\partial u_{ij}} = m u_{ij}^{m-1} f_{ij} - \lambda_i = 0$$ رابطه (۳-۱۹)

$$u_{ij} = \left(\frac{\lambda_i}{m f_{ij}}\right)^{\frac{1}{m-1}}$$ رابطه (۳-۲۰)

با توجه به رابطه (۳-۱) که محدودیت رابطه (۳-۹) است، خواهیم داشت:

$$\sum_{i=1}^{n} u_{ij} = 1$$ رابطه (۳-۲۱)

$$\sum_{j=1}^{c}\left(\frac{\lambda_i}{mf_{ij}}\right)^{\frac{1}{\mathrm{m}-1}}=1$$ رابطه (۳-۲۲)

$$\lambda_i^{\frac{1}{\mathrm{m}-1}}\sum_{i=1}^{c}\left(\frac{1}{mf_{ij}}\right)^{\frac{1}{\mathrm{m}-1}}=1$$ رابطه (۳-۲۳)

از طرفی می‌توان λ_i را به این صورت محاسبه کرد:

$$\lambda_i=\left(\frac{1}{\sum_{k=1}^{c}\left(\frac{1}{mf_{kj}}\right)^{\frac{1}{\mathrm{m}-1}}}\right)^{\mathrm{m}-1}$$ رابطه (۳-۲۴)

با توجه به رابطه (۳-۱۹) و رابطه (۳-۲۴) ، رابطه (۳-۲۵) بدست می‌آید:

$$mu_{ij}^{m-1}f_{ij}=\left(\frac{1}{\sum_{k=1}^{c}\left(\frac{1}{mf_{kj}}\right)^{\frac{1}{\mathrm{m}-1}}}\right)^{\mathrm{m}-1}$$ رابطه (۳-۲۵)

سپس خواهیم داشت:

$$u_{ij}=\frac{1}{\sum_{k=1}^{c}\left(\frac{mf_{ij}}{mf_{kj}}\right)^{\frac{1}{\mathrm{m}-1}}}$$ رابطه(۳-۲۶)

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{f_{ij}}{f_{kj}} \right)^{\frac{1}{m-1}}}$$ رابطه (۳-۲۷)

– میانگین دانش‌پیشین

از تابع لاگرانژ، نسبت به میانگین دانش‌پیشین مشتق گرفته می شود:

$$\frac{\partial L}{\partial \mu_{\theta_j}} = 0$$ رابطه (۳-۲۸)

سپس خواهیم داشت:

$$\mu_{\theta_j} = \mu_j$$ رابطه (۳-۲۹)

میانگین دانش‌پیشین با توجه به رابطه (۳-۲۹)، همان میانگین بدست آمده از تکرار قبلی است.

– کواریانس دانش‌پیشین

رابطه (۳-۳۱) از مشتق تابع لاگرانژ نسبت به کواریانس دانش‌پیشین (Σ_{θ_j}) بدست می‌آید.

$$\frac{\partial L}{\partial \Sigma_{\theta_j}} = 0$$ رابطه (۳-۲۹)

سپس خواهیم داشت:

$$\Sigma_{\theta_j} = \left(\mu_j - \mu_{\theta_j} \right) \left(\mu_j - \mu_{\theta_j} \right)^T$$ رابطه(۳-۳۱)

۳-۳-۴- پیاده‌سازی الگوریتم خوشه‌بندی دسته‌ای

در صورتی‌که نمونه‌ها به صورت یکجا باشد برای هر خوشه یک توزیع گوسی در نظر می‌گیرد. سپس برای پیاده‌سازی الگوریتم با توجه به اولویت‌های آورده شده در جدول (۳-۱)، خوشه‌بندی انجام می‌شود.

در مرحله‌ی اول مقداردهی اولیه تمامی پارامترها μ_j و μ_{θ_j} و Σ_j و Σ_{θ_j} و u_{ij} و f_{ij} انجام می‌شود و u_{ij} و f_{ij} برای تمامی نمونه‌ها بدست می‌آید. در مرحله‌ی دوم با توجه به روابط گفته شده به ترتیب Σ_j و Σ_{θ_j} و μ_j و μ_{θ_j} محاسبه می‌شوند و بعد از یک مرحله افراز تمامی پارامترها بروزرسانی می‌شوند و با یک مقدار آستانه روی مقادیر عضویت نمونه بررسی می‌شود، در صورتی‌که از مقدار آستانه بزرگ‌تر باشد به مرحله اول می‌رود و در غیر این‌صورت به خوشه‌بندی مناسبی رسیده است و برنامه به اتمام می‌رسد.

جدول (۳-۱) – الگوریتم خوشه‌بندی دسته‌ای

۱	مقداردهی اولیه تمامی پارامترها μ_j و μ_{θ_j} و Σ_j و Σ_{θ_j} و u_{ij} و f_{ij}
۲	$f_{ij} = \exp(-0.5((x_i - \mu_j)^T \Sigma_j^{-1}(x_i - \mu_j) + (\mu_j - \mu_{\theta_j})^T \Sigma_j^{-1}(\mu_j - \mu_{\theta_j})))$
۳	$u_{ij} = \frac{1}{\sum_{k=1}^{c}\left(\frac{f_{ij}}{f_{kj}}\right)^{\frac{1}{m-1}}}$
٤	$\Sigma_j = \frac{\sum_{i=1}^{n} u_{ij}^m (x_i - \mu_j)(x_i - \mu_j)^T f_{ij}}{\sum_{i=1}^{n} u_{ij}^m f_{ij}}$
٥	$\Sigma_{\theta_j} = (\mu_j - \mu_{\theta_j})(\mu_j - \mu_{\theta_j})^T$
٦	$\mu_{\theta_j} = \mu_j$
۷	$\mu_j = \left(\frac{\Sigma_j^{-1}}{\Sigma_j^{-1} + \Sigma_{\theta_j}^{-1}} \times \frac{\sum_{i=1}^{n} u_{ij}^m f_{ij} x_i}{\sum_{i=1}^{n} u_{ij}^m f_{ij}}\right) + \left(\frac{\Sigma_{\theta_j}^{-1}}{\Sigma_j^{-1} + \Sigma_{\theta_j}^{-1}} \times \frac{\sum_{i=1}^{n} u_{ij}^m f_{ij} \mu_j}{\sum_{i=1}^{n} u_{ij}^m f_{ij}}\right)$

۳-۴- الگوریتم خوشه‌بندی افزایشی

۳-۴-۱- رویکرد الگوریتم افزایشی

در ادامه‌ی این فصل، با توجه به ماهیت افزایشی جریان‌داده، الگوریتم پیشنهادی برای این نوع داده‌ها توسعه داده می شود. خوشه‌بندی این داده‌ها، به این صورت است که محور زمان به بازه‌های مشخصی تقسیم شود، به این ترتیب داده‌ها بلوک‌بندی خواهند شد. نمونه‌ی جدید که وارد می‌شود با استفاده از الگوریتم خوشه‌بندی افزایشی تشخیص داده می‌شود به کدام خوشه تعلق دارد. بنابراین تعداد نمونه از n به n+1 تغییر پیدا می‌کند.

با توجه به تغییر میانگین جدید($\hat{\mu}_j$) و کواریانس تخمینی جدید ($\hat{\Sigma}_j$) و میانگین دانش‌پیشین جدید ($\hat{\mu}_{\theta_j}$) و کواریانس دانش‌پیشین جدید را ($\hat{\Sigma}_{\theta_j}$) و تابع عضویت جدید ($\hat{u}_{ij}$) روابط تغییر می‌کنند.

در این روش تنها روابط $\hat{\Sigma}_j$ و $\hat{\mu}_j$ تغییر می‌کنند و روابط $\hat{\Sigma}_{\theta_j}$ و $\hat{\mu}_{\theta_j}$ و $\hat{u}_{ij}$ تنها بروزرسانی خواهند شد.

۳-۴-۲- محاسبه پارامترهای خوشه

- **میانگین خوشه**

با استفاده از رابطه (۳-۳۲) میانگین قبلی بدست آمده، رابطه میانگین جدید برای یک نمونه جدید بدست می‌آید.

$$\hat{\mu}_j = \left((S) \times \frac{\sum_{i=1}^{n+1} u_{ij}^m f_{ij} x_i}{\sum_{i=1}^{n+1} u_{ij}^m f_{ij}} \right) + \left((1-S) \times \frac{\sum_{i=1}^{n+1} u_{ij}^m f_{ij} \mu_{\theta_j}}{\sum_{i=1}^{n+1} u_{ij}^m f_{ij}} \right) \qquad \text{رابطه (۳-۳۲)}$$

رابطه (۳-۳۱) نمونه‌های قبلی را از نمونه‌های جدید جدا می‌کند، سپس رابطه (۳-۳۲) بدست می‌آید:

$$\hat{\mu}_j = \left((S)\times(\frac{\sum_{i=1}^{n} u_{ij}^m f_{ij} x_i}{\sum_{i=1}^{n+1} u_{ij}^m f_{ij}} + \frac{u_{n+1\,j}^m f_{n+1\,j} x_{n+1}}{\sum_{i=1}^{n+1} u_{ij}^m f_{ij}}) \right) + \left((1-S)\times(\frac{\sum_{i=1}^{n} u_{ij}^m f_{ij} \mu_{\theta_j}}{\sum_{i=1}^{n+1} u_{ij}^m f_{ij}} + \frac{u_{n+1\,j}^m f_{n+1\,j} \mu_{\theta_j}}{\sum_{i=1}^{n+1} u_{ij}^m f_{ij}}) \right)$$

رابطه (۳-۳۳)

برای ساده‌سازی رابطه (۳-۳۲)، $\frac{\sum_{i=1}^{n} u_{ij}^m f_{ij}}{\sum_{i=1}^{n} u_{ij}^m f_{ij}}$ در مولفه‌های عملگر جمع ضرب می‌شود و (۳-۳۴) بدست می‌آید:

$$\hat{\mu}_j = \left((S)\times(\frac{\sum_{i=1}^{n} u_{ij}^m f_{ij} x_i}{\sum_{i=1}^{n} u_{ij}^m f_{ij}} \times \frac{\sum_{i=1}^{n} u_{ij}^m f_{ij}}{\sum_{i=1}^{n+1} u_{ij}^m f_{ij}}) + (\frac{u_{n+1\,j}^m f_{n+1\,j}}{\sum_{i=1}^{n+1} u_{ij}^m f_{ij}} \times x_{n+1}) \right) + \left((1-S)\times(\frac{\sum_{i=1}^{n} u_{ij}^m f_{ij} \mu_{\theta_j}}{\sum_{i=1}^{n} u_{ij}^m f_{ij}} \times \frac{\sum_{i=1}^{n} u_{ij}^m f_{ij}}{\sum_{i=1}^{n+1} u_{ij}^m f_{ij}}) + \frac{u_{n+1\,j}^m f_{n+1\,j}}{\sum_{i=1}^{n+1} u_{ij}^m f_{ij}} \times \mu_{\theta_j}) \right)$$

رابطه (۳-۳۴)

اگر رابطه (۳-۳۴)، را به کسرهای جدا بشکنیم، خواهیم داشت:

$$\mu_1 = \frac{\sum_{i=1}^{n} u_{ij}^m x_i f_{ij}}{\sum_{i=1}^{n} u_{ij}^m f_{ij}}$$

رابطه (۳-۳۵)

$$\mu_2 = \frac{\sum_{i=1}^{n} u_{ij}^m \mu_{\theta_j} f_{ij}}{\sum_{i=1}^{n} u_{ij}^m f_{ij}}$$

رابطه (۳-۳۶)

دو مقدار μ_1 و μ_2 از نمونه‌های قبلی بدست آمده است.

متغیر α تعریف می‌شود:

$$\alpha = \frac{\sum_{i=1}^{n} u_{ij}^m x_i f_{ij}}{\sum_{i=1}^{n+1} u_{ij}^m f_{ij}}$$

رابطه (۳-۳۷)

و همچنین متغیر $1-\alpha$ نیز تعریف می‌شود:

$$1-\alpha_1 = \frac{\hat{u}_j^m x_i f_{n+1j}}{\sum_{i=1}^{n+1} u_{ij}^m f_{ij}}$$ رابطه (۳-۳۸)

اکنون با استفاده از ، روابط (۳-۳۵)، (۳-۳۶)، (۳-۳۷)و (۳-۳۸)، رابطه(۳-۳۴) را می‌توان به صورت ساده شده بیان می شود:

$$\hat{\mu}_j = (S \times (\mu_1.(\alpha_1) + (1-\alpha_1)x_{n+1})) + (\mu_2.(\alpha_2) + (1-\alpha_2)\mu_{\theta_j})$$ رابطه (۳-۳۹)

– کواریانس

با استفاده از رابطه (۳-۴۰) کواریانس قبلی بدست آمده، رابطه کواریانس جدید برای یک نمونه‌ی جدید بدست می‌آید.

$$\Sigma_j = \frac{\sum_{i=1}^{n} u_{ij}^m \left(x_i - \mu_j\right)\left(x_i - \mu_j\right)^T f_{ij}}{\sum_{i=1}^{n} u_{ij}^m f_{ij}}$$ رابطه (۳-۴۰)

$$\hat{\Sigma}_j = \frac{\sum_{i=1}^{n+1} u_{ij}^m \left(x_i - \mu_j\right)\left(x_i - \mu_j\right)^T f_{ij}}{\sum_{i=1}^{n+1} u_{ij}^m f_{ij}}$$ رابطه (۳-۴۱)

رابطه (۳-۳۱) نمونه‌های قبلی را از نمونه‌های جدید جدا می‌کند، سپس رابطه (۳-۳۲) بدست می‌آید.

$$\hat{\Sigma}_j = \frac{\sum_{i=1}^{n} u_{ij}^m \left(x_i - \mu_j\right)\left(x_i - \mu_j\right)^T f_{ij}}{\sum_{i=1}^{n+1} u_{ij}^m f_{ij}} + \frac{u_{n+1j}^m \left(x_{n+1} - \mu_j\right)\left(x_{n+1} - \mu_j\right)^T f_{n+1j}}{\sum_{i=1}^{n+1} u_{ij}^m f_{ij}}$$ رابطه(۳-۴۲)

و متغیر α تعریف می‌شود:

$$\alpha = \frac{\sum_{i=1}^{n} u_{ij}^m \mu_{\theta_j} f_{ij}}{\sum_{i=1}^{n} u_{ij}^m f_{ij}}$$ رابطه (۳-۴۳)

و متغیر $1-\alpha$ تعریف می‌شود:

رابطه (۳-۴۴)

$$1-\alpha = \frac{u_{n+1j}^{m}\mu_{\theta_j} f_{n+1j}}{u_{n+1j}^{m} f_{n+1j}}$$

اکنون با استفاده از ، روابط (۳-۴۳) و (۳-۴۴)، رابطه(۳-۴۵) را می‌توان به صورت ساده‌تر بیان شود:

رابطه(۳-۴۵)

$$\hat{\Sigma}_j = (\Sigma_j \times (\alpha) + (1-\alpha)(x_{n+1} - \mu_j)(x_{n+1} - \mu_j)^T$$

– کواریانس دانش‌پیشین

رابطه‌ی $\Sigma_{\theta_j} = \sum_{i=1}^{n}(x_{n+1} - \mu_j)(x_{n+1} - \mu_j)^T$ فقط به کل نمونه‌ها بستگی دارد، پس نمونه جدید را به نمونه‌های قبلی اضافه می‌کنیم و رابطه (۳-۴۶) بدست می‌آید:

رابطه(۳-۴۶)

$$\hat{\Sigma}_{\theta_j} = \left(\hat{\mu}_j - \hat{\mu}_{\theta_j}\right)\left(\hat{\mu}_j - \hat{\mu}_{\theta_j}\right)^T$$

۳-۴-۳- تعمیم الگوریتم برای تعداد داده‌های بیشتر

با توجه به پارامترهای بدست آمده بخش (۲-۱۱-۲) الگوریتم خوشه‌بندی افزایشی، اگر یک نمونه جدید وارد الگوریتم شود. می‌توان برای m نمونه‌ی جدید نیز تعمیم داد.

۳-۴-۴- پیاده‌سازی الگوریتم خوشه‌بندی افزایشی

ابتدا نمونه‌ها با استفاده از بخش اول الگوریتم ارائه شده، خوشه‌بندی می‌شود و برای هر خوشه، پارامترهای توزیع گوسی بدست می‌آید. با تولید داده‌های تدریجی، از این بخش از الگوریتم افزایشی استفاده می‌شود، سپس برای پیاده‌سازی الگوریتم با توجه به اولویت‌های آورده شده در جدول (۳-۲) خوشه‌بندی انجام می‌شود.

در مرحله‌ی اول مقداردهی اولیه تمامی پارامترها μ_j و μ_{θ_j} و Σ_j و Σ_{θ_j} و u_{ij} و f_{ij} انجام می‌شود و u_{ij} و f_{ij} برای تمامی نمونه‌ها بدست می‌آید. در مرحله‌ی دوم با توجه به روابط گفته شده به ترتیب Σ_j و Σ_{θ_j} و μ_j و μ_{θ_j} محاسبه می‌شوند و بعد از یک مرحله‌ی افراز تمامی پارامترها بروزرسانی می‌شوند و با یک مقدار آستانه روی مقادیر عضویت نمونه بررسی می‌شود، در صورتی‌که از مقدار آستانه بزرگ‌تر باشد به مرحله اول می‌رود و در غیر اینصورت به خوشه‌بندی مناسبی رسیده است و برنامه به اتمام می‌رسد.

جدول (۳-۲)- الگوریتم خوشه‌بندی افزایشی

۱	مقداردهی اولیه تمامی پارامترها μ_j و μ_{θ_j} و Σ_j و Σ_{θ_j} و u_{ij} و f_{ij}
۲	$\hat{f}_{ij} = \exp(-0.5((x_i - \mu_j)^T \Sigma_j^{-1}(x_i - \mu_j) + (\mu_j - \mu_{\theta_j})^T \Sigma_j^{-1}(\mu_j - \mu_{\theta_j})))$
۳	$\hat{u}_{ij} = \frac{1}{\sum_{k=1}^{c}\left(\frac{f_{ij}}{f_{kj}}\right)^{\frac{1}{m-1}}}$
٤	$\hat{\Sigma}_j = (\Sigma_j \times (\alpha) + (1-\alpha)(x_{n+1} - \mu_j)(x_{n+1} - \mu_j)^T$
٥	$\hat{\Sigma}_{\theta_j} = \left(\hat{\mu}_j - \hat{\mu}_{\theta_j}\right)\left(\hat{\mu}_j - \hat{\mu}_{\theta_j}\right)^T$
٦	$\hat{\mu}_{\theta_j} = \hat{\mu}_j$
۷	$\hat{\mu}_j = (S \times (\mu_1.(\alpha) + (1-\alpha)x_{n+1})) + (\mu_2.(\alpha) + (1-\alpha)\mu_{\theta_j})$

۳-۴-۱- نتیجه‌گیری

در این فصل یک الگوریتم جداپذیرخطی مبتنی بر مدل ارائه شد. الگوریتم ارائه شده، به دلیل استفاده از دانش‌پیشین، میزان شباهت دقیقتری را بدست می‌آورد و خوشه‌بندی بهتری انجام می‌دهد. بیشترین تاثیر دانش‌پیشین در کواریانس‌ها است. اگر نمونه نویزی در بین نمونه‌ها باشد، تابع شباهتِ آن به خوشه‌ها نزدیک به

صفر می‌شود و در نتیجه مقدار تابع عضویت نیز نزدیک به صفر خواهد شد. بنابراین آن نمونه را نویز تشخیص می‌دهد.

۴- فصل چهارم

آزمایشات و ارزیابی

۴-۱- مقدمه

در این فصل از پایان‌نامه نتایج آزمایش‌های انجام شده برای ارزیابی روش پیشنهادی که در فصل سوم تشریح گردید، ارائه و بررسی خواهد شد. در ابتدای فصل معیارهای خوشه‌بندی بیان می‌شوند. در بخش بعدی محیط، شرایط و پایگاه داده‌ی آزمایشات توضیح داده خواهد شد. سپس نتایج و ارزیابی الگوریتم، برای کاهش زمان خوشه‌بندی به صورت توزیع شده در محیط اسپارک (یکی از ابزارهای محاسبات خوشه-ای) استفاده می‌گردد.

۴-۲- معیارهای خوشه‌بندی

در الگوریتم‌های خوشه‌بندی، خوشه‌ای مناسب است که تا حد امکان نمونه‌های داخل خوشه بیشترین شباهت به یکدیگر و حداکثر فشردگی[۱] را داشته باشند و نسبت به نمونه‌های خوشه‌های دیگر بیشترین تفاوت را داشته باشد و تا حد امکان جدایی[۲] آنها نیز زیاد باشد. معیار استفاده شده، به شرح زیر است و رابطه‌ی ریاضی آنها در جدول (۴-۱) نشان داده شده است.

Partition Coefficien (Bezdek 1973) : اگر الگوریتمی معیار Partition Coefficient بزرگتری داشته باشد، الگوریتم بهتری برای خوشه‌بندی است. مقدار این معیار بین 1/c و ۱ می‌باشد.

Separation Index (S) (Bensaid, Hall 1996): اگر الگوریتمی معیار **Separation Index کوچکتری** داشته‌باشد، الگوریتم مناسب‌تری برای خوشه‌بندی است.

Dunn's Index (DI) (Dunn 1973): با اجرای خوشه‌بندی الگوریتمی که معیار Dunn's Index کمتری داشته باشد، الگوریتم بهتری برای خوشه‌بندی است. برای تشخیص خوشه‌های کوچکتر، این

[۱] Compactness

[۲] Separation

معیار، معیار مناسبی است به دلیل اینکه واریانس بین خوشه‌ها کوچک‌تر می‌باشد. معیار Alternative Dunn Index (ADI) توسعه یافته معیار Dunn Index است، ولی محاسبات کمتری دارد.

ماتریس آشفتگی[1](Fawcett 2006): ماتریسی است که برای سنجش دقت[2] و صحت[3] نتایج الگوریتم استفاده می‌شود. این ماتریس بیشتر برای ارزیابی الگوریتم‌های دسته‌بندی بکار برده می‌شود. اما در این فصل، پایگاه داده‌های آزمایشی استاندارد استفاده شده، که دسته‌ی واقعی آنها مشخص است. به همین دلیل ماتریس آشفتگی معیار خوبی برای ارزیابی است. ستون‌های این ماتریس، داده‌های واقعی و سطرهای ماتریس نتایج الگوریتم است.

جدول (۴-۱) – رابطه ریاضی معیارهای ارزیابی

معیار	فرمول
Partition coefficient	$V_{PC}(U) = \frac{1}{n}(\sum_{k=1}^{n}\sum_{i=1}^{c} {u_{ik}}^2)$
Alternative Dunn Index (ADI)	$DI(c) = \min_{i\in c}\left\{\min_{j\in c, i\neq j}\left\{\frac{\min_{x_i\in C_i, x_j\in C_j} \lvert d(y,\mu_j) - d(x,\mu_j)\rvert}{\max_{j\in C}\{\max_{x,y\in C} d(x,y)\}}\right\}\right\}$
Dunn's Index (DI)	$DI(c) = \min_{i\in c}\left\{\min_{j\in c, i\neq j}\left\{\frac{\min_{x\in C_i, y\in C_j} d(x,y)}{\max_{j\in C}\{\max_{x,y\in C} d(x,y)\}}\right\}\right\}$
Separation Index (S)	$S(c) = \frac{\sum_{i=1}^{c}\sum_{i=1}^{n} u_{ij}^2 \lVert x_j - \mu_j\rVert^2}{N \min_{i,k} \lVert \mu_k - \mu_j\rVert^2}$

[2] Confusion Matrix

[3] Precision

[3] Recall

۴-۲-۱- پایگاه داده

پایگاه داده‌هایی که الگوریتم پیشنهادی روی آنها اجرا شده است در جدول (۴-۲) مشاهده می‌شوند. داده‌های مصنوعی تولید شده در پایگاه داده دارای ۵۰۰۰ نمونه، دو ویژگی عددی، با ۱۰ خوشه می‌باشد. مجموعه داده IRIS، Blood، Glass و Ecoli مجموعه داده‌های استاندارد از مجموعه داده UCI است.

جدول (۴-۲) - پایگاه داده‌هایی الگوریتم پیشنهادی روی آن ها آزمایش شده است

پایگاه داده	تعداد نمونه ها	تعداد ویژگی‌ها	تعداد خوشه
داده های تولید شده	۵۰۰۰	۲	۱۰
IRIS[1]	۱۵۰	۴	۳
Blood[2](BL)	۷۴۸	۴	۲
Glass[3](GL)	۲۱۴	۹	۶
Ecoli[4](ECO)	۳۳۶	۷	۸

۴-۲-۲- محیط پیاده‌سازی

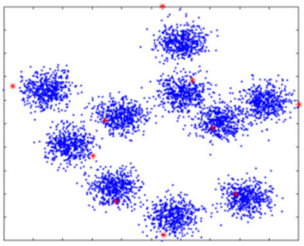
ابتدا الگوریتم دسته‌ای و افزایشی در محیط نرم‌افزار متلب پیاده‌سازی شده است. الگوریتم پیشنهادی با الگوریتم‌های FCM (Wang 1983) و GK (Gustafson 1979) مقایسه شده است. با توجه به حساس بودن این الگوریتم‌ها به نقاط اولیه، نقاط شروع یکسانی برای آزمایشات در نظر گرفته شد. در آزمایش پایگاه داده تولیدی، نقاط اولیه برای هر خوشه در جدول (۴-۳) نشان داده میشود.

[1] https://archive.ics.uci.edu/ml/datasets/iris
[2] https://archive.ics.uci.edu/ml/datasets/Blood+Transfusion+Service+Center
[3] https://archive.ics.uci.edu/ml/datasets/Glass+Identification
[4] https://archive.ics.uci.edu/ml/datasets/Ecoli

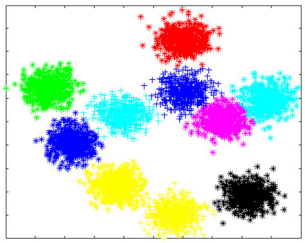
۴-۳- آزمایشات با داده دسته‌ای

۴-۳-۱- داده‌های تولید شده

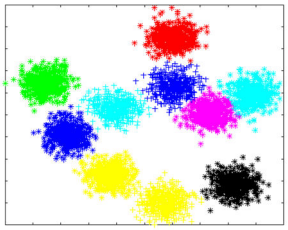
داده‌های تولید شده در شکل (۴-۱) نشان داده شده‌اند. نقاط اولیه در این شکل با رنگ قرمز نشان داده شده است. همان‌طور که مشاهده می‌شود، ۱۰ خوشه در این داده‌ها موجود است. شکل (۴-۱- ب) خروجی الگوریتم FCM بر روی داده‌های شکل (۴-۱- الف) می‌باشد، در شکل (۴-۱- ج) خروجی الگوریتم GK نشان داده شده و خروجی الگوریتم پیشنهادی در شکل (د) نمایش داده شده است. همان‌طور که در شکل‌های (ب) و (ج) و (د) نشان داده خروجی هر سه الگوریتم تا حد زیادی مشابه است و از نظر بصری تفاوتی در آنها مشاهده نمی‌شود. با این وجود، تفاوت اندک در مرز خوشه‌ها موجب تفاوت در شاخص‌های ارزیابی خواهد شد که در ادامه نشان داده شده‌اند.



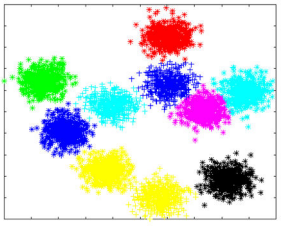
شکل (الف)



شکل (ب)



شکل (ج)



شکل (د)

شکل (۴-۱) – (الف) نقاط اولیه برای خوشه‌بندی- (ب) خوشه‌بندی با الگوریتم FCM- (ج) – خوشه‌بندی با الگوریتم GK - شکل (د) – خوشه‌بندی با الگوریتم BFC

جدول (۴-۴) معیارهای خوشه‌بندی PC و SI و DI و ADI را برای الگوریتم‌های FCM و GK و BFC روی مجموعه داده تولیدی نشان می‌دهد. در مقایسه، اعمال معیار های خوشه‌بندی PC و SI و DI روی الگوریتم پیشنهادی نتایج بهتری را نشان می‌دهد.

جدول (۴-۳) - معیارهای خوشه‌بندی برای داده‌های تولیدی

مجموعه داده / معیارهای ارزیابی خوشه	PC	SI	DI	ADI
الگوریتم				
FCM	0.7170	1.7770e-05	0.0125	8.7587e-04
GK	0.7136	1.7781e-05	0.0066	2.8033e-04
BFC	**0.9993**	**1.0299e-05**	**0.0125**	5.4760e-04

IRIS -۲-۳-۴

الگوریتم پیشنهادی روی مجموعه داده‌های IRIS آزمایش شده است. جدول (۴-۵) نقاط اولیه الگوریتم را نشان می‌دهد. نتایج آزمایش در جدول (۴-۶) نشان داده شده، معیار PC، برای الگوریتم پیشنهادی بهتر از الگوریتم‌های FCM و GK عمل می‌کند. برای معیار PC الگوریتم پیشنهادی با الگوریتم GathGava تقریبا برابر است.

همچنین در جدول(۴-۷) ماتریس آشفتگی را برای این پایگاه داده بدست آوردیم. ماتریس بدست‌آمده برای FCM و GK به صورت مشابه هستند و ماتریس آشفتگی الگوریتم پیشنهادی و Gava -Gath تقریبا مشابه است.

جدول (۴-۵) - نقاط اولیه برای پایگاه داده IRIS

	بعد۱	بعد۲	بعد۳	بعد۴
خوشه ۱	۰,۹۴۳۰	۰,۴۲۴۹	۰,۴۲۱۱	۰,۳۹۸۱
خوشه ۲	۰,۱۲۱۵	۰,۸۷۸۵	-۰,۰۴۵۴	۰,۵۸۳۲
خوشه۳	۰,۵۵۰۸	۰,۷۳۳۸	۰.۸۰۹۸	۰,۷۷۴۵

جدول (۴-۶) - معیارهای خوشه‌بندی برای مجموعه داده IRIS

IRIS **مجموعه داده**

الگوریتم	PC	SI	DI	ADI
FCM	0:7425	0:0063	0:0347	0:0090
KG	0:7277	0:0049	0:0430	0:0194
BFC	**0.9817**	0:0056	0:0694	0:0034

جدول (۴-۷) - ماتریس آشفتگی برای نتایج اجرای GKو GG و FCM وBFC برروی پایگاه داده IRIS

GK	**FCM**	**GG**	**BFC**
47 0 12	46 0 12	0 50 0	45 0 0
0 50 0	0 50 0	45 0 0	0 50 0
3 0 38	4 0 38	5 0 50	5 0 50

با توجه به ماتریس آشفتگی جدول(۴-۷)، تعداد عناصری که درست تشخیص داده شده را نسب به کل ماتریس آشفتگی برای الگوریتم‌های GK و GG و FCM وBFC در جدول (۴-۸) نشان می‌دهد. الگوریتم ارائه شده و الگوریتم GG بیشترین تشخیص درست، نسبت به الگوریتم‌های دیگر دارند.

جدول (۴-۸) - نتایج بدست آمده از تشخیص درست خوشه‌بندی از اجرای GKو GG و FCM وBFC برروی پایگاه داده IRIS

GK	**0.9**	**FCM**	**0.89**
GG	**0.96**	**BFC**	**0.96**

glass -۳-۳-۴

الگوریتم ارائه شده روی داده glass آزمایش شده است. نتایج آن در جدول (۴-۹) مشاهده می‌شود. معیارهای PC و S برای الگوریتم پیشنهادی بهتر از FCM و GK عمل می‌کند.

جدول (۴-۹)- معیارهای خوشه‌بندی برای مجموعه داده glass

الگوریتم	PC[۱]	S[۲]	DI[۳]	ADI[۴]
GK	0. 0.8006 + 0.0000i	0.1727 + 0.0295i	0.0373	2.6952e-04
FCM	0.4186	0.0070	0.0248	1.1090e-05
BFCM	0.9993	0.0020	0.0877	0.0011

ecoli -۴-۳-۴

الگوریتم ارائه شده برای داده‌های ecoli آزمایش شد. نتایج آن در جدول (۴-۱۰) نشان داده شده است. معیارهای PC و S برای الگوریتم پیشنهادی بهتر از FCM و GK عمل می‌کند.

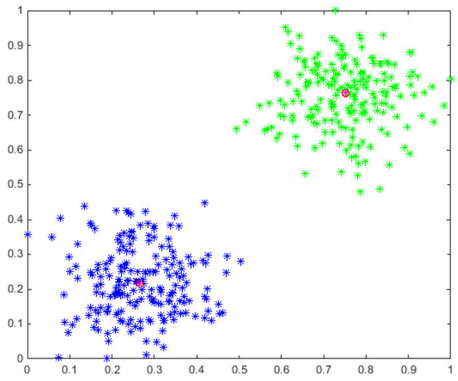
جدول(۴-۱۰)- معیارهای خوشه‌بندی برای مجموعه داده ecoli

الگوریتم	PC[۵]	S[۶]	DI[۷]	ADI[۸]
GK	0.5361 + 0.0000i	0.0030 + 0.0006i	0.0308	1.7108e-06
FCM	0.3031	0.0057	0.0325	4.1917e-04
BFCM	0.9958	0.0011	0.0624	1.6073e-04
GG	0.9398 - 0.0071i	Nan	0.0292	1.5678e+19

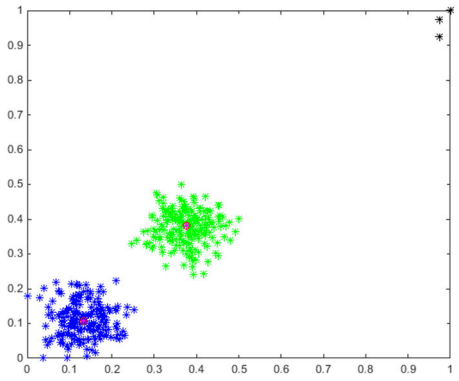
[۱] Partition coefficient
[۲] Separation Index
[۳] Dunn's Index
[۴] Alternative Dunn Iindex
[۵] Partition coefficient
[۶] Separation Index
[۷] Dunn's Index
[۸] Alternative Dunn Iindex

۴-۴- داده‌های پرت

شکل (۴-۲) (الف) خوشه‌بندی بدون داده‌ی پرت و (ب) خوشه‌بندی با داده‌ی پرت نشان می‌دهد. نمونه‌ها را با الگوریتم ارائه شده خوشه‌بندی می‌کنیم. الگوریتم داده‌های پرت را به درستی تشخیص می‌دهد. به دلیل اینکه مقدار تابع عضویت داده‌های پرت به هر دو خوشه تقریبا برابر صفر است، این داده‌ها نقش خاصی در تعیین پارامترهای مدل خوشه‌بندی ندارند و الگوریتم آنها را به عنوان داده‌ی پرت در نظر می‌گیرد.



شکل(ب)



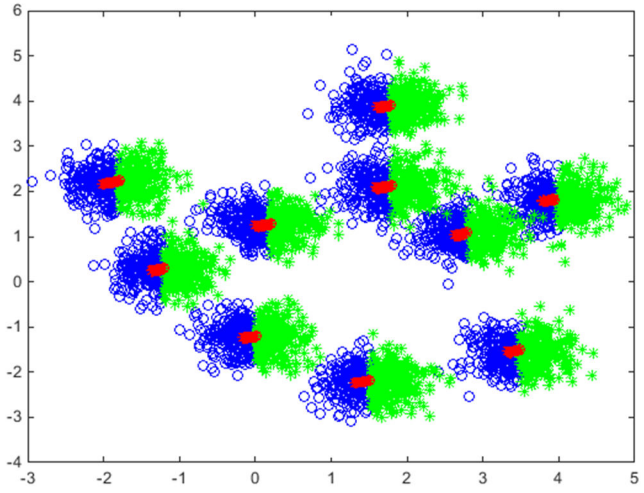
شکل (الف)

شکل (۴-۲) الف)خوشه‌بندی بدون داده‌ی پرت – ب) خوشه‌بندی با داده‌ی پرت

۴-۵- آزمایش الگوریتم افزایشی

این بخش از آزمایش‌ها، روی داده‌های تولیدی انجام شده است. این داده‌ها به دو بخش تقسیم شده و نیمه‌ی اول به صورت یکجا با الگوریتم دسته‌ای آزمایش شده است و نیمه‌ی دوم داده‌ها که در شکل(۴-۳) با رنگ **سبز** مشخص شده به طور تدریجی به الگوریتم معرفی شده و به طور افزایشی خوشه‌بندی می‌شود. در شکل (۴-۳) میانگین هر خوشه با ستاره **قرمز** رنگ مشخص شده و مراکز به سمت داده‌های

جدید در حرکت است. مراکز نهایی هر خوشه‌ای که با الگوریتم افزایشی بدست آمده، با مراکز الگوریتم اولیه در جدول (۴-۱۱) مقایسه شده است.

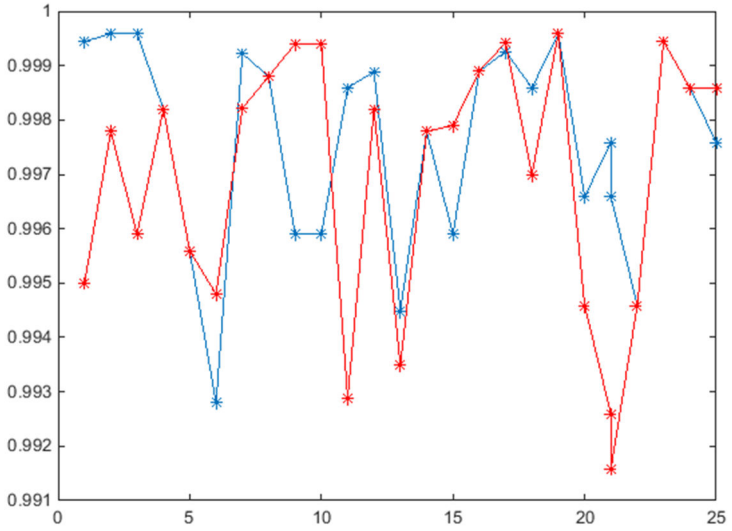


شکل (۴-۳) الگوریتم خوشه‌بندی افزایشی

مختصات اصلی دسته‌هایی که با الگوریتم خوشه‌بندی افزایشی (**Inc-BFC**) بدست‌آمده را با الگوریتم خوشه‌بندی دسته‌ای (**BFC**) برای دو بعد X و Y مقایسه کرده‌ایم. مراکز هر دو دسته برای هر دو الگوریتم تقریبا نزدیک بهم می‌باشد. در شکل (۴-۴) مقادیر معیار PC برای الگوریتم دسته با رنگ **آبی** و افزایشی با رنگ **قرمز** مشخص شده است، به طوری که در هر بار ۱۰۰ داده به الگوریتم معرفی شود. همان‌طور که مشخص است این معیار برای هر دو الگوریتم تقریبا مشابه است.

جدول (۴-۱۱)- مراکز بدست آمده خوشه‌بندی افزایشی (Inc-BFC) و مراکز خوشه‌بندی دسته‌ای (BFC) برای هر دو بعد داده مصنوعی

خوشه	Inc-BFC X	Inc-BFC Y	BFC X	BFC Y
۱	-1.2134	0.2976	-1.3178	0.2978
۲	-1.7976	2.2314	-1.6044	2.3054
۳	0.0121	-1.1880	-0.0386	-1.1975
۴	2.7880	1.1055	2.8030	1.1227
۵	3.4782	-1.5108	3.4228	-1.5542
۶	3.9293	1.7840	3.7916	1.8331
۷	1.8112	3.9196	1.8396	3.7513
۸	0.2007	1.2903	0.1219	1.3137
۹	1.5035	-2.1797	1.6310	-2.3668
۱۰	1.7956	1.9998	1.8825	2.0668



شکل (۴-۴) مقایسه معیارPC الگوریتم دسته و افزایشی

۴-۶- آزمایشات الگوریتم پیشنهادی به صورت توزیع شده

در این قسمت الگوریتم بر روی چارچوب نرم‌افزاری اسپارک پیاده‌سازی و اجرا شده است. در پیاده‌سازی این الگوریتم تمامی محاسباتی که برای روی داده انجام شده را در گام نگاشت[۱] قرار داده‌ایم و سپس با استفاده از گام کاهش[۲] که مدیر[۳] اجرا می‌کند، عملیات مربوط به تجمیع نتایج خروجی اجرا می‌شود.

۴-۷- الگورریتم پیشنهادی نگاشت- کاهش

عملیات گام نگاشت الگوریتم ارائه شده را در جدول (۴-۱۲) نشان داده‌ایم. همان‌طور که مشاهده می‌شود، عملیاتی که روی همه داده وجود دارد روی تمام سیستم ها نگاشت می‌شود و این باعث می‌شود این عملیات توزیع شود و به صورت بهینه اجرا گردد. در قسمت کاهش بعد از اینکه نتایج توزیع‌شده

[1] map
[2] reduce
[3] master

بدست آمد، روی نود مدیر تجمیع انجام می‌گیرد. عملیاتی که چنین ماهیتی دارند در جدول(۴-۱۳) نشان داده شده است.

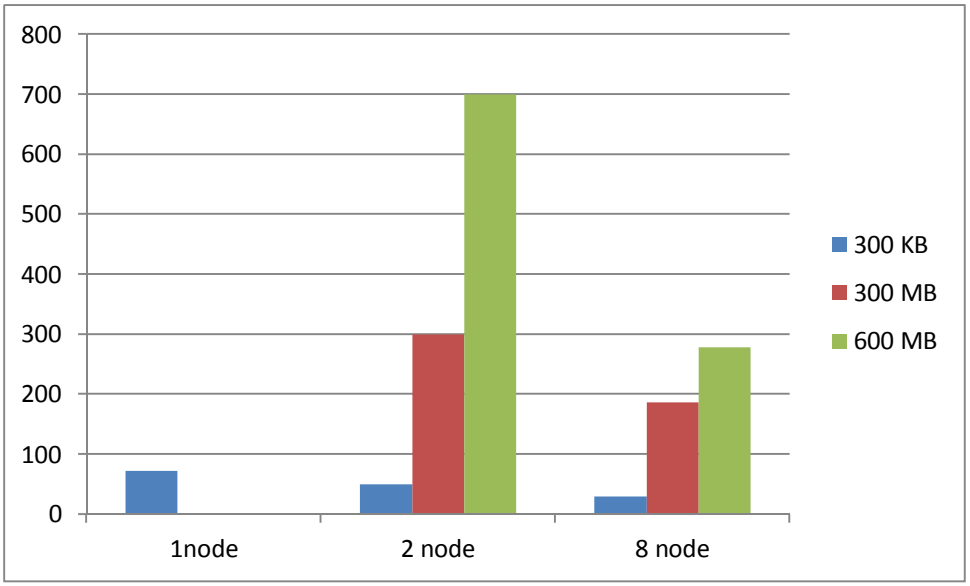
جدول (۴-۱۲)- عملیات مربوط به مرحله نگاشت

۱	$f_{ij} = \exp(-0.5((x_i - \mu_j)^T \Sigma_j^{-1}(x_i - \mu_j) + (\mu_j - \mu_{\theta_j})^T \Sigma_j^{-1}(\mu_j - \mu_{\theta_j})))$
۲	$u_{ij} = \frac{1}{\sum_{k=1}^{c}\left(\frac{f_{ij}}{f_{kj}}\right)^{\frac{1}{m-1}}}$
۳	$\left(x_i - \mu_j\right)\left(x_i - \mu_j\right)^T$
۴	$\sum_{i=1}^{n} u_{ij}^m f_{ij} x_i$
۵	$\sum_{i=1}^{n} u_{ij}^m f_{ij} \mu_j$

جدول (۴-۱۳)- عملیات مربوط به مرحله کاهش

۱	$\sum_{i=1}^{n} u_{ij}^m f_{ij}$
۲	$\Sigma_{\theta_j} = \left(\mu_j - \mu_{\theta_j}\right)\left(\mu_j - \mu_{\theta_j}\right)^T$
۳	$\mu_{\theta_j} = \mu_j$
۴	$\frac{\Sigma_j^{-1}}{\Sigma_j^{-1} + \Sigma_{\theta_j}^{-1}}$
۵	$\frac{\Sigma_j^{-1}}{\Sigma_j^{-1} + \Sigma_{\theta_j}^{-1}}$
۶	محاسبه Σ_j و μ_j

شکل (۴-۵) زمان اجرای الگوریتم پیشنهادی برای ۳۰ کیلوبایت داده و ۳۰۰ مگابایت داده و ۶۰۰ مگابایت داده روی ۱ نود کامپیوتر و ۲ نود و ۸ نود را نشان می دهد. به نسبت ۸ نود زمان پردازش کمتری دارد.



شکل (۴-۵) اجرای الگوریتم پیشنهادی با اسپارک روی نودها

۵- فصل پنجم

نتایج و کارهای آینده

۵-۱- نتیجه‌گیری

در این پایان نامه ابتدا به بررسی مفهوم جریان‌داده و چالش‌های تحلیل آن پرداختیم. خوشه‌بندی یکی از راه‌های متداول تحلیل داده است. روش‌های خوشه‌بندی جریان داده در فصل دوم بیان شد. در فصل سوم پایان‌نامه الگوریتمی مبتنی بر مدل ارائه شد. الگوریتم ارائه شده یک الگوریتم جداپذیر خطی می‌باشد که به دلیل استفاده از دانش‌پیشین، خوشه‌بندی بهتری را انجام می‌دهد. با در نظر گرفتن توزیع گوسی برای خوشه‌ها، مشاهده شد که بیشترین تاثیر دانش‌پیشین در کواریانس خوشه‌ها می‌باشد.

آزمایشات روی داده‌های مصنوعی و واقعی نشان می دهد که الگوریتم ارائه شده با دقت بهتری نسبت به الگوریتم‌های FCM و GK داده را خوشه‌بندی می‌کند. یکی از مزیت‌های الگوریتم تاثیر کم نقاط پرت است در خوشه‌بندی می‌باشد.

الگوریتم برای ورود داده های جریانی که داده به صورت تدریجی تولید می شوند، توسعه داده شد. به دلیل حجم بالای داده، برای کاهش زمان پردازش از محاسبات خوشه‌ای استفاده شد. ابزار اسپارک برای این موضوع انتخاب شد که الگوریتم با داده حجیم آزمایش گردید و کاهش زمان اجرا بررسی گردید.

۵-۲- کارهای آینده

در این بخش به بیان برخی کارهایی می‌پردازیم که می توان در آینده براساس الگوریتم پیشنهادی انجام داد.

۵-۲-۱- خوشه‌بندی جدید

در اکثر الگوریتم‌ها، تعداد خوشه‌های اولیه از قبل مشخص شده‌اند و تا انتهای اجرای الگوریتم تغییر نمی‌کنند. اما حین اجرای الگوریتم پیشنهادی، اگر خوشه‌ای جدیدی اضافه شود، قابل شناسایی می‌باشد. به این منظور می توان نمونه‌هایی که نویز تشخیص داده می‌شود را درون بافری قرار داد و نسبت به زمان

به آنها وزنی تخصیص داده داد. در صورتی که تعداد آنها از یک آستانه بیشتر شد، آنها را به عنوان یک خوشه جدید معرفی نمود.

۵-۲-۲- ارزش‌دهی به نمونه‌ها در طول زمان

با توجه به ماهیت افزایشی جریان‌داده، و تغییر مفهوم در طول زمان، ارزش نمونه‌های قدیم‌تر کاهش می‌یابد. این ارزش را می تواند به صورت نمایی بیان کرد.

اگر نمونه‌ها به صورت پنجره‌هایی به طول Δt باشد. Δt اختلاف زمان فعلی و زمان شروع پنجره است. وزن نمونه‌ها در صورتی که زمان نمونه را نیز در نظر بگیریم، به این صورت رابطه (۵-۱) بیان گردد:

$$f(\Delta t) = \lambda^{\Delta t} \qquad (0 < \lambda < 1)$$ رابطه (۵-۱)

مدل خوشه‌های جدید، نسبت به مدل خوشه‌های قدیمی‌تر ارزش بیشتری دارند (Balamurugan , Visalatchi 2016) (Chi, Song 2009) این مفهوم می‌تواند به عنوان قید در تابع هدف اضافه کرد.

۵-۲-۳- بهینه‌سازی الگوریتم افزایشی

با استفاده از روش‌هایی که برای بهینه کردن میانگین افزایشی (Shenmaier 2016) وجود دارد، می-توان مراکز خوشه و کورایانس آنها را بهینه نمود.

مراجع

[1] RR Ade and PR Deshmukh. Methods for incremental learning: a survey. International Journal of Data Mining & Knowledge Management Process, 3(4):119, 2013.

[2] Charu C Aggarwal, Jiawei Han, Jianyong Wang, and Philip S Yu.A framework for clustering evolving data streams. In Proceedings of the 29th international conference on Very large data bases-Volume 29, pages 81-92. VLDB Endowment, 2003.1

[3] Saeed Aghabozorgi, Mahmoud Reza Saybani, and Teh Ying Wah. Incremental clustering of time-series by fuzzy clustering. Journal of Information Science and Engineering, 28(4):671-688, 2012.

[4] Marco Aldinucci, Massimo Coppola, Sonia Campa, Marco Danelutto,Marco Vanneschi, and Corrado Zoccolo. Structured implementation of component-based grid programming environments. In Future Generation Grids, pages 217-239. Springer, 2006.

[5] M Balamurugan and L Visalatchi. Self-constructing clusters in dynamic heterogeneous multi typed network. In Intelligent Systems and Control (ISCO), 2016 10th International Conference on, pages 1-5. IEEE, 2016.

[6] Adi Ben-Israel and Cem Iyigun. Probabilistic d-clustering. Journal of Classi_cation, 25(1):5, 2008.

[7] Amine M Bensaid, Lawrence O Hall, James C Bezdek, Laurence PClarke, Martin L Silbiger, John A Arrington, and Reed F Murtagh.Validity-guided (re) clustering with applications to image segmentation.IEEE Transactions on Fuzzy Systems, 4(2):112-123, 1996.

[8] James C Bezdek. Pattern recognition with fuzzy objective function algorithms. Springer Science & Business Media, 2013.

[9] Vasudha Bhatnagar, Sharanjit Kaur, and Sharma Chakravarthy. Clustering data streams using grid-based synopsis. Knowledge and information systems, 41(1):127-152, 2014.

[10] Christian C Bones, Luciana AS Romani, and Elaine PM de Sousa.Improving multivariate data streams clustering. Procedia Computer Science, 80:461-471, 2016.

[11] Manfred Borovcnik, Hans-Joachim Bentz, and Ramesh Kapadia. A probabilistic perspective. In Chance encounters: Probability in education, pages 27-71. Springer, 1991.

[12] Sanjay Chakraborty and Naresh Kumar Nagwani. Analysis and study of incremental dbscan clustering algorithm. arXiv preprint arXiv:1406.4754, 2014.
[13] Sanjay Chakraborty and NK Nagwani. Analysis and study of incremental k-means clustering algorithm. In High Performance Architecture and Grid Computing, pages 338-341. Springer, 2011.2
[14] Yun Chi, Xiaodan Song, Dengyong Zhou, Koji Hino, and Belle L Tseng. On evolutionary spectral clustering. ACM Transactions on Knowledge Discovery from Data (TKDD), 3(4):17, 2009.
[15] Florence Corpet. Multiple sequence alignment with hierarchical clustering. Nucleic acids research, 16(22):10881-10890, 1988.
[16] Xuan Hong Dang, Vincent CS Lee, Wee Keong Ng, and Kok Leong Ong. Incremental and adaptive clustering stream data over sliding window. In International Conference on Database and Expert Systems Applications, pages 660-674. Springer, 2009.
[17] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. Journal of the royal statistical society. Series B (methodological), pages 1-38, 1977.
[18] Joseph C Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. 1973.
[19] Paulo Martins Engel and Milton Roberto Heinen. Incremental learning of multivariate gaussian mixture models. In Brazilian Symposium on Arti_cial Intelligence, pages 82-91. Springer, 2010.
[20] Tom Fawcett. An introduction to roc analysis. Pattern recognition letters, 27(8):861-874, 2006.
[21] Isak Gath and Amir B. Geva. Unsupervised optimal fuzzy clustering. IEEE Transactions on pattern analysis and machine intelligence,11(7):773-780, 1989.
[22] Taylor C Glenn, Alina Zare, and Paul D Gader. Bayesian fuzzy clustering. IEEE Transactions on Fuzzy Systems, 23(5):1545-1561, 2015.
[23] Donald E Gustafson and William C Kessel. Fuzzy clustering with a fuzzy covariance matrix. In Decision and Control including the 17th Symposium on Adaptive Processes, 1978 IEEE Conference on, pages761-766. IEEE, 1979.
[24] Richard Hyde, Plamen Angelov, and AR MacKenzie. Fully online clustering of evolving data streams into arbitrarily shaped clusters. Information Sciences, 382:96-114, 2017.
[25] Anil K Jain. Data clustering: 50 years beyond k-means. Pattern recognition letters, 31(8):651-666, 2010.3

[26] Supun Kamburugamuve, Geo_rey Fox, David Leake, and Judy Qiu. Survey of apache big data stack. Indiana University, Tech. Rep., 2013.
[27] Holden Karau, Andy Konwinski, Patrick Wendell, and Matei Zaharia. Learning spark: lightning-fast big data analysis. " O'Reilly Media, Inc.",2015.
[28] Imran Khan, Joshua Z Huang, and Kamen Ivanov. Incremental density based ensemble clustering over evolving data streams. Neurocomputing,191:34-43, 2016.
[29] Jae Woo Lee and Won Suk Lee. A coarse-grain grid-based subspace clustering method for online multi-dimensional data streams. In Proceedings of the 17th ACM conference on Information and knowledge management, pages 1521-1522. ACM, 2008.
[30] Richard S Morrison. Cluster computing: Architectures, operating systems, parallel processing & programming languages. GNU General Public Licence, 2003.
[31] Carlos Ruiz, Ernestina Menasalvas, and Myra Spiliopoulou. C-denstream: Using domain knowledge on a data stream. In International Conference on Discovery Science, pages 287-301. Springer, 2009.
[32] Indrajit Saha and Ujjwal Maulik. Incremental learning based multi objective fuzzy clustering for categorical data. Information Sciences,267:35-57, 2014.
[33] Vladimir Shenmaier. An approximation algorithm for the Euclidean incremental median problem. Discrete Optimization, 22:312-327, 2016.
[34] Mingzhou Song and HongbinWang. Incremental estimation of Gaussian mixture models for online data stream clustering. In Proceedings of the international conference on bioinformatics and its applications, 2004.
[35] Yangtao Wang, Lihui Chen, and Jian-Ping Mei. Incremental fuzzy clustering with multiple medoids for large data. IEEE Transactions on Fuzzy Systems, 22(6):1557-1568, 2014.
[36] JunWu Xu and JunLing Liang. Research on distributed le system with hadoop. In Network Computing and Information Security, pages148-155. Springer, 2012.4
[37] Chee Shin Yeo, Rajkumar Buyya, Hossein Pourreza, Rasit Eskicioglu, Peter Graham, and Frank Sommers. Cluster computing: High performance, high-availability, and high-throughput processing on a network of computers. In Handbook of nature-inspired and innovative computing, pages 521-551. Springer, 2006.

[38] Laiwen Zheng, Hong Huo, Yiyou Guo, and Tao Fang. Supervised adaptive incremental clustering for data stream of chunks. Neurocomputing,219:502-517, 2017.
[39] Wagstaff K, Cardie C, Rogers S, Schrödl S. Constrained k-means clustering with background knowledge. ICML ,pages 577-584 , 2001.

پیوست

ضمایم اثبات روابط ریاضی فصل سوم الگوریتم پیشنهادی

اثبات رابطه کواریانس

$$\frac{\partial L}{\partial \Sigma_j} =$$

$$\frac{-1}{2}\sum_{i=1}^{n} u_{ij}^{\ m} \frac{\left|\Sigma_j\right|\ \Sigma_j^{-1}\ \left|\Sigma_{\theta_j}\right|^{\frac{-3}{2}}}{(2\pi)^{\frac{d}{2}}\left|\Sigma_j\right|^{\frac{1}{2}}\left|\Sigma_{\theta_j}\right|^{\frac{1}{2}}}\ exp\left(\begin{pmatrix}\left(-0.5\left(x_i-\mu_j\right)^T\Sigma_j^{-1}\left(x_i-\mu_j\right)\right)+\\ \left(-0.5\left(\mu_{j-}\mu_{\theta_j}\right)^T\Sigma_{\theta_j}^{-1}\left(\mu_j-\mu_{\theta_j}\right)\right)\end{pmatrix}\right) -$$

$$\frac{1}{2}\sum_{i=1}^{n}\frac{u_{ij}^{\ m}}{(2\pi)^{\frac{d}{2}}\left|\Sigma_j\right|^{\frac{1}{2}}\left|\Sigma_{\theta_j}\right|^{\frac{1}{2}}}\Sigma_j^{-1}\ \left(x_i-\mu_j\right)\left(x_i-\mu_j\right)^T\ \Sigma_j^{-1}\left(\begin{pmatrix}\left(-0.5\left(x_i-\mu_j\right)^T\Sigma_j^{-1}\left(x_i-\mu_j\right)\right)+\\ \left(-0.5\left(\mu_j-\mu_{\theta_j}\right)^T\Sigma_{\theta_j}^{-1}\left(\mu_j-\mu_{\theta_j}\right)\right)\end{pmatrix}\right)=0$$

$$\Sigma_j^{-1}=\frac{\sum_{i=1}^{n}u_{ij}^{m}f_{ij}}{\sum_{i=1}^{n}u_{ij}^{m}\left(x_i-\mu_j\right)\left(x_i-\mu_j\right)^T f_{ij}}$$

$$\Sigma_{\ j}=\frac{\sum_{i=1}^{n}u_{ij}^{m}\left(x_i-\mu_j\right)\left(x_i-\mu_j\right)^T f_{ij}}{\sum_{i=1}^{n}u_{ij}^{m}f_{ij}}$$

اثبات رابطه میانگین دانش پیشین

$$\frac{\partial L}{\partial \mu_{\theta_j}} =$$

$$\sum_{i=1}^{n}\frac{u_{ij}^{m}}{(2\pi)^{\frac{d}{2}}\left|\Sigma_j\right|^{\frac{1}{2}}\left|\Sigma_{\theta_j}\right|^{\frac{1}{2}}}(0.5\left((-2)\Sigma_{\theta_j}^{-1}\left(\mu_j-\mu_{\theta_j}\right)\right))exp(\left(0.5\left(x_i-\mu_j\right)^T\Sigma_j^{-1}\left(x_i-\mu_j\right)\right)+\left(0.5\left(\mu_j-\mu_{\theta_j}\right)^T\Sigma_{\theta_j}^{-1}\left(\mu_j-\mu_{\theta}\right.\right.$$

$$\sum_{i=1}^{n}\frac{u_{ij}^{m}}{(2\pi)^{\frac{d}{2}}\left|\Sigma_j\right|^{\frac{1}{2}}\left|\Sigma_{\theta_j}\right|^{\frac{1}{2}}}\left(\ \Sigma_{\theta_j}^{-1}\mu_{\theta_j}-\ \Sigma_{\theta_j}^{-1}\mu_j\right)exp\left(\begin{pmatrix}\left(0.5\left(x_i-\mu_j\right)^T\Sigma_j^{-1}\left(x_i-\mu_j\right)\right)+\\ \left(0.5\left(\mu_j-\mu_{\theta_j}\right)^T\Sigma_{\theta_j}^{-1}\left(\mu_j-\mu_{\theta_j}\right)\right)\end{pmatrix}\right)=0$$

$$\left(\ \Sigma_{\theta_j}^{-1}\mu_{\theta_j}-\ \Sigma_{\theta_j}^{-1}\mu_j\ \right)\frac{1}{(2\pi)^{\frac{d}{2}}\left|\Sigma_j\right|_{\frac{1}{2}}\left|\Sigma_{\theta_j}\right|^{\frac{1}{2}}}\sum_{i=1}^{n}u_{ij}^{m}\ exp\left(\begin{pmatrix}\left(0.5\left(x_i-\mu_j\right)^T\Sigma_j^{-1}\left(x_i-\mu_j\right)\right)+\\ \left(0.5\left(\mu_j-\mu_{\theta_j}\right)^T\Sigma_{\theta_j}^{-1}\left(\mu_j-\mu_{\theta_j}\right)\right)\end{pmatrix}\right)=\ 0$$

$$\Sigma_{\theta_j}^{-1}\mu_{\theta_j}-\ \Sigma_{\theta_j}^{-1}\mu_j=0$$

$$\mu_{\theta_j}=\mu_j$$

اثبات رابطه کواریانس دانش پیشین

$$\frac{\partial L}{\partial \Sigma_{\theta_j}} =$$

$$\frac{-1}{2}\sum_{i=1}^{n} u_{ij}^{m} \frac{\left|\Sigma_{\theta_j}\right| \ \Sigma_{\theta_j}^{-1} \left|\Sigma_{\theta_j}\right|^{\frac{-3}{2}}}{(2\pi)^{\frac{d}{2}} \left|\Sigma_j\right|^{\frac{1}{2}} \left|\Sigma_{\theta_j}\right|^{\frac{1}{2}}} \ exp\left(\begin{array}{l}\left(0.5\left(x_i-\mu_j\right)^T \Sigma_j^{-1}\left(x_i-\mu_j\right)\right)+ \\ \left(0.5\left(\mu_{j-}\mu_{\theta_j}\right)^T \Sigma_{\theta_j}^{-1}\left(\mu_j-\mu_{\theta_j}\right)\right)\end{array}\right)-$$

$$\frac{1}{2}\sum_{i=1}^{n} \frac{u_{ij}^{m}}{(2\pi)^{\frac{d}{2}} \left|\Sigma_j\right|^{\frac{1}{2}} \left|\Sigma_{\theta_j}\right|^{\frac{1}{2}}} \Sigma_{\theta_j}^{-1}\left(\mu_j-\mu_{\theta_j}\right)\left(\mu_j-\mu_{\theta_j}\right)^T \ \Sigma_{\theta_j}^{-1} exp\left(\begin{array}{l}\left(0.5\left(x_i-\mu_j\right)^T \Sigma_j^{-1}\left(x_i-\mu_j\right)\right)+ \\ \left(0.5\left(\mu_j-\mu_{\theta_j}\right)^T \Sigma_{\theta_j}^{-1}\left(\mu_j-\mu_{\theta_j}\right)\right)\end{array}\right) = 0$$

$$\Sigma_{\theta_j}^{-1} = \frac{1}{\left(\mu_j-\mu_{\theta_j}\right)\left(\mu_j-\mu_{\theta_j}\right)^T}$$

$$\Sigma_{\theta_j} = \left(\mu_j-\mu_{\theta_j}\right)\left(\mu_j-\mu_{\theta_j}\right)^T$$

ضمایم کد اسکالا الگوریتم روی اسپارک

```
import org.apache.spark.rdd.RDD
import org.apache.spark.SparkContext
import breeze.numerics_.
import breeze.linalg_.
object BFC extends Serializable}
  def calculateFij(dataX:  breeze.linalg.DenseMatrix[Double]  , Mu: breeze.linalg.DenseMatrix[Double]  ,
Mu_teta: breeze.linalg.DenseMatrix[Double] ,covt : breeze.linalg.DenseMatrix[Double]   , cov_teta :
breeze.linalg.DenseMatrix[Double]    ) : Double} =

  var tmp = (dataX).-(Mu (
  var t1 = tmp.*(covt).*(tmp.t(

  val tmpA = Mu.-(Mu_teta(
  var t2 = tmpA.*(cov_teta).*(tmpA.t(
----------------------------------------------------------------------------------------------------------//
-
//  return function
----------------------------------------------------------------------------------------------------------//
-----
  Math.exp(((t1 .+ (t2)).*(- 0.5)).toArray(0((
{

----------------------------------------------------------------------------------------------------------//
  def calculateUF ( u : Double , f :Double  , m: Double ): Double} =
    Math.pow(u,m) * f
{

----------------------------------------------------------------------------------------------------------///
  def calculateSigma  (  u : Double  ,  dataX:  breeze.linalg.DenseMatrix[Double] , Mu:
breeze.linalg.DenseMatrix[Double]  ,  f : Double  , m : Int ): breeze.linalg.DenseMatrix[Double} = [
    var tmp = (dataX.-(Mu( (
    var tmp2= tmp.t
    val tmp3= tmp2.*(tmp(
    tmp3.*(Math.pow ( u , m  )* f(
{

----------------------------------------------------------------------------------------------------------///
  def calculateSigma_teta  (  Mu:  breeze.linalg.DenseMatrix[Double]  ,Mu_teta:
breeze.linalg.DenseMatrix[Double]  ): breeze.linalg.DenseMatrix[Double} = [
    var tmp = ( Mu .-(Mu_teta( (
    var tmp2= tmp.t
)   tmp2.*(tmp((
{
```

```
-----------------------------------------------------------------------------------------------------------------------//
    def calculatS( Sigma : breeze.linalg.DenseMatrix[Double]  , Sigma_teta :
breeze.linalg.DenseMatrix[Double] , err : breeze.linalg.DenseMatrix[Double]  ) :
breeze.linalg.DenseMatrix[Double} = [
      Sigma.+(err).*(inv ((Sigma.+( Sigma_teta )).+(err(((
{
-----------------------------------------------------------------------------------------------------------------------//
    def calculatSAx( Sigma : breeze.linalg.DenseMatrix[Double]  , Sigma_teta :
breeze.linalg.DenseMatrix[Double] , err : breeze.linalg.DenseMatrix[Double]  ) :
breeze.linalg.DenseMatrix[Double} = [
)    Sigma_teta.+(err)).*(inv ( (Sigma.+( Sigma_teta ))+err((
{
-----------------------------------------------------------------------------------------------------------------------//
    def calculateMu ( dataX : breeze.linalg.DenseMatrix[Double]  ,  Muteta :
breeze.linalg.DenseMatrix[Double] ,  U:Double  ,F :Double , m: Int ) : (breeze.linalg.DenseMatrix[Double],
breeze.linalg.DenseMatrix[Double} = (  [
      var UF =  Math.pow( U ,m)  *F
)   dataX.*(UF) , Muteta.*(UF((
{
-----------------------------------------------------------------------------------------------------------------------//
    def execute(dataX: org.apache.spark.rdd.RDD[breeze.linalg.DenseMatrix[Double]], data :
Array[breeze.linalg.DenseMatrix[Double]]   , Mu: Array[breeze.linalg.DenseMatrix[Double]  ] , Mu_teta:
Array[ breeze.linalg.DenseMatrix[Double]  ] , covt: Array[breeze.linalg.DenseMatrix[Double]  ] , cov_teta:
Array [breeze.linalg.DenseMatrix[Double} = ( [[
      iterat(dataX,data ,Mu, Mu_teta ,covt , cov_teta , 12(
{
-----------------------------------------------------------------------------------------------------------------------//
    def calculateMu1 ( Mu1 : breeze.linalg.DenseMatrix[Double]  ,  Mu2:
breeze.linalg.DenseMatrix[Double] ,  S :  breeze.linalg.DenseMatrix[Double] , SS :
breeze.linalg.DenseMatrix[Double]  , SumUF : Double ) : breeze.linalg.DenseMatrix[Double} = [
)    S.*(((Mu1)./(SumUF)).t)).+ ((SS).*(((Mu2)./(SumUF)).t((
{
-----------------------------------------------------------------------------------------------------------------------//
    def calculateParam ( x: breeze.linalg.DenseMatrix[Double] , Mu:
Array[breeze.linalg.DenseMatrix[Double]  ] , Mu_teta: Array[ breeze.linalg.DenseMatrix[Double]  ] , Cov:
Array[breeze.linalg.DenseMatrix[Double]] , Cov_teta: Array[breeze.linalg.DenseMatrix[Double]] , j: Int , m :
Int) : (Array[Double ], Array[Double ] ,Double, breeze.linalg.DenseMatrix[Double] ,
(breeze.linalg.DenseMatrix[Double], breeze.linalg.DenseMatrix[Double} = ((
      var F= Array (calculateFij( x  , Mu(0) , Mu_teta(0) ,Cov(0) , Cov_teta(0)  ) , calculateFij( x  , Mu(1) ,
Mu_teta(1) ,Cov(1) , Cov_teta(1(( (
      var sumation = F(0) + F(1(
      var U =Array (F(0)/sumation, F(1)/sumation(
      var UF= BFC.calculateUF(U(j),F(j),m(
      var UFD = BFC.calculateSigma (U(j) , x , Mu(j) , F(j)  , m(
      var UFM = BFC.calculateMu( x , Mu_teta(j), U(j), F(j),m(
```

```
) F,U,UF,UFD, UFM  (
{
-------------------------------------------------------------------------------------------------------------//
  def iterat(dataRDD: org.apache.spark.rdd.RDD[breeze.linalg.DenseMatrix[Double]] , data :
Array[breeze.linalg.DenseMatrix[Double]] , Mu: Array[breeze.linalg.DenseMatrix[Double]] , Mu_teta:
Array[breeze.linalg.DenseMatrix[Double]] ,Cov: Array[breeze.linalg.DenseMatrix[Double]] , Cov_teta:
Array [breeze.linalg.DenseMatrix[Double]] , numberOfIterate: Int):
Array[breeze.linalg.DenseMatrix[Double} = [[
    if (numberOfIterate == 0}(
     Mu
{
    else }
      // var j = 0
     for (j <- 0 to 1}(
       var m =2
       var err = DenseMatrix((0.001,0.0),(0.0,0.001((
       var FUX= dataRDD.map(x => BFC.calculateParam ( x, Mu , Mu_teta ,Cov , Cov_teta ,j ,2((
       //println(FUX.first(

       val  sumUF  =FUX.map (x=> x._3 ).reduce; (_+_)
       var Mu1 = FUX.map(x => x._5._1  ).reduce(_+_)
       var Mu2 = FUX.map(x => x._5._2  ).reduce(_+_)

       val  UFD  =FUX.map (x=> x._4 ).reduce(_+_)/sumUF         ;
       Cov(j) = inv(UFD.+(err;((
-------------------------------------------------------------------//
       val UFD1 =   BFC.calculateSigma_teta ( Mu(j)  , Mu_teta(j)) / sumUF
-------------------------------------------------------------------//
       Cov_teta(j) = inv(UFD1.+(err;((
-------------------------------------------------------------------//
       val S = BFC.calculatS(Cov(j),Cov_teta(j),err;(
       val SS = BFC.calculatSAx(Cov(j), Cov_teta(j) ,err;(
-------------------------------------------------------------------//
  //     var Mu1 = FUX.map(x => x._5._1  ).reduce(_+_)
  //     var Mu2 = FUX.map(x => x._5._2  ).reduce(_+_)
       Mu_teta(j) =DenseMatrix(Mu(j).toArray(0) , Mu(j).toArray(1) ).t
-------------------------------------------------------------------//
       Mu(j) =BFC.calculateMu1(Mu1 , Mu2,SS,S,sumUF).t
-------------------------------------------------------------------//
{
    iterat(dataRDD,data,  Mu, Mu_teta ,Cov , Cov_teta , numberOfIterate - 1(
{
{
```

Abstract

Today, in various applications such as traffic control systems, weather sensors, medical systems, social networks, etc. data is produced at a rapid rate. This phenomenon is referred to as data stream, and its analysis typically requires evolutionary and incremental methods. Clustering is among the common methods of data analysis and mining, and several algorithms have been presented in this area. Clustering data streams requires methods different from classic approaches due to reasons such as lack of simultaneous access to all data, cluster changes, etc.

This thesis presents a model-based clustering algorithm that uses conditional probability theory and prior knowledge for a probabilistic clustering. The proposed algorithm is analyzed with artificial and real data, and later compared with FCM and Gustafson-Kessel methods. According to the results, the proposed algorithm improves the accuracy of clustering and demonstrates more resistance to noise.

Given the incremental nature of the data stream, the proposed algorithm is extended for this type of data and evaluated after implementation using the Spark cluster computing framework. With the parallel implementation of the algorithm, the runtime is reduced. The experiments show the scalability of the proposed algorithm for handling large amounts of data.

Keywords: Clustering, Probabilistic Clustering, DataStream, DataStream Clustering

Shahrood University of Technology

Faculty of computer

M.Sc. Thesis in Artificial Intelligence

Improving Clustering Algorithms for Big Data Using Cluster Computing

By: Zohreh Ferydoon Moghadam

Supervisor:

Dr. H. Mashayekhi

Advisor

Dr. V. Abolghasemi

February 2017