الگوریتم‌های خوشه‌بندی جریان متن و جریان داده

مریم خیرخواه‌زاده [۱] ، معصومه خیرخواه‌زاده[۲]

چکیده

مسئله خوشه‌بندی. مسئله دشواری در زمینه جریان داده بشمار می‌رود. این مشکل به دلیل حجم زیاد داده دریافتی از یک جریان است که باعث ناکارآمد شدن الگوریتم‌های سنتی شده است.

مسئله خوشه‌بندی اخیرا در زمینه جریان داده‌های عددی، بسیار مورد مطالعه قرار گرفته است. ولی، تحقیقات در زمینه خوشه‌بندی جریان متن، هنوز در آغاز راه است. خوشه‌بندی جریان متن کاربردهایی از قبیل فیلترسازی گروه خبری، خزش متن، سازمان‌دهی متن و تشخیص و ردیابی موضوع دارد.

در این مقاله تعدادی از الگوریتم‌های خوشه‌بندی جریان داده و خوشه‌بندی جریان متن بحث می‌شود.

کلمات کلیدی

جریان داده، جریان متن، خوشه‌بندی جریان، مدل هموارساز مفهومی.

Introduction to text and data Stream Clustering Algorithms

Maryam,kheirkhahzadeh; Masoumeh kheirkhahzadeh;

ABSTRACT

The clustering problem is a difficult problem for the data stream domain. This is because the large volumes of data arriving in a stream renders most traditional algorithms too ineffieient.

The clustering problem has recently been studied in the context of numeric data streams. But, the text data streams clustering research is only on the underway stage. Clustering text streams has a number of application as news group filtering, text crawling, document organization and topic detection and tracing etc.

In this paper, the number of algorithms for clustering data streams and text streams are discussed.

KEYWORDS

Data Streams, text Streams, Stream Clustering, Semantic Smoothing Model.

۱. مقدمه

امروزه خوشه‌بندی جریان داده یکی از نیازهای اساسی است که در کاربردهایی مثل تشخیص حمله به شبکه، جستجو، شبکه‌های حسگر، فیلترسازی گروه‌های خبری و ... مطرح می‌شود. جریان داده، دنباله‌ای پیوسته از داده‌ها است که به همان ترتیب دریافت، مورد دسترسی قرار می‌گیرند. بنابراین امکان دسترسی تصادفی به داده‌های دریافتی وجود ندارد. علت این است که حافظه نسبت به جریان کوچک است. در نتیجه تنها امکان ذخیره تعداد محدودی از نقاط داده، یا یک خلاصه آماری از داده‌ها وجود دارد. طبیعت پویای جریان داده و سرعت دریافت آن محدودیت‌های زمان و فضا را بر کاربردهای جریان داده اعمال می‌کند که طراحی الگوریتم‌های جریان داده را دشوار می‌سازد.

خوشه‌بندی جریان داده، یکی از مسائل پیچیده‌ای است که محققان را با چالش‌های زیادی روبرو ساخته است. سه نیاز اساسی یک الگوریتم خوشه-بندی جریان داده ناشی از محدودیت فضا و زمان، به شرح زیر است[۳]:

۱عضو هیئت علمی دانشگاه آزاد اسلامی- واحد اهواز- ایمیل: kheirkhah@iauahvaz.ac.ir
۲کارشناسی ارشد دانشگاه امیرکبیر- ایمیل: kheirkhahzadeh@gmail.com

- نمایش فشرده خوشه‌ها- جریان داده، یک جریان پیوسته و پر سرعت است، پس به دلیل محدودیت در حافظه و زمان، الگوریتم نمی تواند شرح مفصلی از هر خوشه را نگه دارد. داده‌ها باید برخط[1] پردازش شوند و طوری فشرده شوند که حافظه اصلی گنجایش ذخیره آنها را داشته باشد، چون چک کردن داده‌های حافظه ثانویه به دلیل سرعت پایین این حافظه امکان‌پذیر نیست. الگوریتم نباید اجازه دهد، حجم اطلاعات فشرده با ورود داده‌های جدید به طرز محسوس رشد کند.
- پردازش سریع داده‌های جدید- عملیات مقایسه و اضافه کردن داده‌های جدید نیازمند سرعت و البته داشتن نمایش فشرده‌ای از داده‌هاست که قبلا شرح دادیم.
- تشخیص سریع و از بین بردن outlier- منظور از outlier، نقاطی است که به هیچ یک از خوشه‌هایی که تاکنون یافته‌ایم نزدیک و مشابه نیستند. طبیعتا باید مکانیزمی برای تصمیم‌گیری در مورد برخورد با outlierها وجود داشته باشد. توجه به این داده‌ها در الگوریتم، نتایج خوشه‌بندی را به انحراف می‌کشد.

در این مقاله روش‌های ارائه شده در زمینه خوشه‌بندی جریان داده معرفی شده است. در بخش ۲ برخی تعاریف اولیه را بیان می‌کنیم. در بخش۳به معرفی الگوریتم‌های خوشه‌بندی جریان داده می‌پردازیم. به دلیل اینکه الگوریتم‌های خوشه‌بندی متن جزء الگوریتم‌های با فضای ابعاد بالا هستند، خوشه‌بندی جریان متن با محدودیت‌های بیشتری نسبت به خوشه‌بندی جریان داده روبروست. بنابراین روش‌های خوشه‌بندی جریان متن را در بخش۴جداگانه معرفی کرده‌ایم. در پایان هر یک از بخش‌های ۳ و ۴ الگوریتم‌های مطرح شده را با هم مقایسه می‌کنیم.

۲. پیش زمینه‌ها

در این بخش به بیان مفاهیم اولیه و برخی کلمات کلیدی می‌پردازیم. الگوریتم‌های مبتنی برمشابهت و مبتنی برمدل دو نوع الگوریتم متفاوت برای خوشه‌بندی است که در این بخش شرح می‌دهیم. سپس تقسیم بندی الگوریتم‌های جریان داده به مقیاس‌پذیر و تطبیقی را بررسی می‌کنیم. در پایان نیز روش‌های موجود در زمینه تشخیص و حذف outliers را معرفی می‌کنیم.

۱.۲. الگوریتم‌های مبتنی برمشابهت و مبتنی برمدل

روشهای خوشه‌بندی ارائه شده برای داده‌کاوی به دو دسته تقسیم می شوند: discriminative یا مبتنی بر تشابه[2] و generative یا مبتنی بر مدل[3]. در روش مبتنی بر تشابه هدف رسیدن به مقدار بهینه تابع هدف است. هدف این تابع کمینه کردن میانگین تشابه داده‌های درون هر خوشه و بیشینه کردن تشابه بین خوشه‌هاست. از سوی دیگر روشهای مبتنی بر مدل، به دنبال یادگیری مدلهای مولد از داده‌ها هستند. هر مدل نمایشگر یک گروه خاص از داده‌هاست. یعنی سعی دارد به هر خوشه یک مدل را نسبت بدهد. در واقع روش‌های خوشه‌بندی مبتنی بر مدل، بر اساس احتمال هستند. الگوریتم‌های خوشه‌بندی جریان داده شامل STREAM[۶]، CluStream[۱] و HPStream[۲] که در بخش ۳ معرفی می‌شوند مبتنی برتشابه هستند. در بخش ۴ دو روش خوشه‌بندی جریان متن، stream OSKM[۸] و TF-ICF[۵]، مبتنی برتشابه و سپس الگوریتم OCTS[۱۲]، مبتنی برمدل، معرفی می‌شود.

۲.۲. الگوریتم‌های مقیاس‌پذیر و تطبیقی

تحقیقات اخیر بر خوشه‌بندی جریان داده را می توان به دو دسته تقسیم کرد. روشهای مقیاس‌پذیر[4] و روشهای تطبیقی[5]. به بیان ساده، یک الگوریتم خوشه‌بندی مقیاس‌پذیر، جریان داده را به صورت بخش به بخش، دریافت کرده و خوشه‌بندی را انجام می‌دهد. الگوریتم هر بار با دریافت یک دسته از داده‌ها، عملیات خوشه‌بندی را انجام می‌دهد. هدف در روشهای مقیاس‌پذیر کسب ویژگی‌های کلی داده، در یک دوره طولانی زمانی علی‌رغم اینکه در یک زمان فقط مجموعه کوچکی از داده در حافظه جا می‌گیرد(برای یک دوره زمانی محدود). اما روش‌های تطبیقی، به دنبال تغییرات در ویژگی‌های داده در طول زمان هستند. بنابراین کیفیت الگوریتم تطبیقی به طور محلی روی یک مجموعه نقاط جریان داده سنجیده می شود. به بیان ساده‌تر یک الگوریتم تطبیقی با دریافت هر داده جدید، مشابه‌ترین خوشه به داده را انتخاب کرده و داده را به خوشه موردنظر اضافه می‌کند.

نمونه‌هایی از الگوریتم‌های مقیاس‌پذیر خوشه‌بندی جریان، STREAM[۶] و streaming OSKM[۸] و الگوریتم‌های تطبیقی CluStream[۱]، HPStream[۲] و OCTS[۱۲] است.

متاسفانه الگوریتم‌های مقیاس‌پذیر برای خوشه‌بندی جریان، نامتناهی بودن جریان داده و تغییر پیوسته آن را با گذشت زمان، در نظر نگرفته‌اند. درواقع در چنین روش‌هایی، با یک الگوریتم شبه پیوسته روبرو هستیم.

۳.۲. تشخیص و حذف outliers

همانطور که گفتیم outliers، نقاطی هستند که به هیچ کدام از از خوشه‌هایی که تا بحال یافته‌ایم ارتباطی ندارند. یکی از چالش‌های مهم در زمینه خوشه‌بندی جریان، تشخیص outliers و حذف آنهاست. الگوریتم STREAM[۶]، که هیچ راهکاری در این زمینه در نظر نگرفته است. اما [۱]، روش-هایی برای تشخیص outlier، معرفی کرده است که در اینجا به معرفی این راهکارها می‌پردازیم. ساده‌ترین روش یافتن خوشه حاوی کمترین تعداد داده ورودی نسبت به خوشه‌های دیگر است. روش دیگر انتخاب خوشه حاوی قدیمی‌ترین داده از جریان است به شرطی که این خوشه اخیرا فعال نبوده و داده جدیدی به آن اضافه نشده باشد. اما روش ایده‌آل، محاسبه میانگین مهر زمانی آخرین m داده اضافه شده به هر خوشه است. سپس خوشه‌ای که کمترین میانگین زمانی را دارد باید به عنوان outlier حذف شود. اما ذخیره m مهرزمانی به ازاء هر خوشه، با توجه به محدودیت گنجایش حافظه اصلی، مشکل‌ساز می‌شود. این عمل حافظه لازم را با فاکتور m افزایش می‌دهد. اما روش استفاده شده در CluStream، نگهداری داده راجع به مهرهای زمانی(نه خود مهرهای زمانی) است. می‌توانیم به راحتی میانگین و انحراف معیار استاندارد را به‌ازاء هر خوشه بیابیم. سپس با فرض توزیع نرمال زمان‌های ورود، زمان m/۲nامین درصد را بیابیم. [۱]، این زمان را مهر مرتبط[۶] نامیده است. خوشه‌ای که کمترین مهر مرتبط را دارد، کاندید حذف به عنوان outlier خواهد بود. در صورتی که مقدار این مهر، از یک حد آستانه کمتر باشد، خوشه کاندید، باید حذف شود. به بیان ساده‌تر، در چه زمانی m/۲n درصد داده‌ها وارد هر یک از خوشه‌ها شده است. آن خوشه‌ای که خیلی وقت پیش این تعداد داده را دریافت کرده مدتهاست ورودی نداشته است. پس احتمالا outlier است. HPStream، روش جدیدی در زمینه حذف outlier به کارنگرفته است و خوشه‌ای را که اخیرا کمتر بروزرسانی شده حذف می‌کند.

الگوریتم HPStream، با دریافت داده جدید، فاصله و OCTS، تشابه آن را نسبت به هر یک از خوشه‌ها محاسبه می‌کند. سپس خوشه‌ای که بیشترین فاصله(کمترین تشابه) را با داده جدید دارد تعیین می‌کند. اگر فاصله(تشابه) دورترین خوشه، از یک حد آستانه بیشتر(کمتر) باشد، هر دو الگوریتم فرض را براین می‌گذارند که داده جدید، نشانه‌ای از شروع یک جریان جدید است. این مسئله ناشی از طبیعت پویای جریان است. پس خوشه جدیدی حاوی داده جدید ایجاد می‌شود. اما باید یکی از خوشه‌ها حذف شود. علت محدودیت گنجایش حافظه است. پس نمی‌توانیم اطلاعات تعداد زیادی خوشه را نگه داریم. بعلاوه در یک عملیات خوشه‌بندی تعداد خوشه‌ها ثابت فرض می‌شود. اینکه کدام خوشه باید حذف شود، همان مسئله تعیین و حذف outlier است.

۳. الگوریتم‌های خوشه‌بندی جریان داده

خوشه‌بندی جریان داده، به اعمال الگوریتم خوشه‌بندی به صورت بلادرنگ بر جریان داده می‌پردازد. توانایی پردازش داده در یک گذر و خلاصه‌سازی آن، در حالی که با محدودیت حافظه روبرو هستیم، یک مسئله بحرانی برای خوشه‌بندی جریان است.

چندین روش خوشه‌بندی جریان داده، از قبیل STREAM[۶]، CluStream[۱] و HPStream[۲] اخیرا مطرح شده است. ابتدا به معرفی این سه الگوریتم و در نهایت در بخش ۴.۲ به مقایسه آنها می‌پردازیم.

۱.۳. STREAM

الگوریتم STREAM[۶] ، جریان داده را بخش[۷]، به بخش خوشه‌بندی می‌کند. الگوریتم در هر بار تکرار به اندازه ظرفیت حافظه نقاط ورودی را می خواند، آنها را خوشه‌بندی می کند. مراکز داده‌ها را نگه می‌دارد و داده‌های دیگر را دور می‌ریزد. STREAM در نهایت خوشه‌بندی را روی مراکز خوشه‌های به دست آمده از مراحل مختلف، اعمال می کند. وزن هر مرکز برابر تعداد نقاط درون خوشه مربوطه، در نظر گرفته می شود. مشکل این الگوریتم این است که با افزایش تعداد بسته‌ها، زمان خوشه‌بندی در مرحله دوم افزایش می یابد. این روش سیاستی برای رویارویی با outliers در نظر نگرفته است. الگوریتم STREAM۲[۷]، نیز صورت توسعه یافته‌ای از STREAM است که به کمک جستجوی دودویی سرعت را بهبود بخشیده است. دو الگوریتم ارائه شده در[۶,۷]، نامتناهی بودن جریان داده و تغییر پیوسته آن را با گذشت زمان، در نظر نگرفته‌اند.

۲.۳. CluStream

Aggarwal در سال ۲۰۰۳[۱]، یک چهارچوب کلی برای فرآیند خوشه‌بندی جریان داده به نام CluStream مطرح کرد. ایده او دو مرحله‌ای کردن فرآیند خوشه‌بندی[۸] بود :

- ایجاد خلاصه آماری به صورت برخط

- استفاده از این خلاصه‌ها به صورت برون‌خط[۹]

الگوریتم‌های قبلی خوشه‌بندی، مثل [۶,۷] فرض می کنند که می‌توان خوشه‌بندی را روی کل جریان داده اعمال کرد. چنین روشهایی مسئله خوشه‌بندی را به‌سادگی، به صورت یک روش خوشه‌بندی تک‌گذره در نظر می گیرند. حال اینکه مسئله فراتر از اینهاست. جریان داده باید به صورت یک فرآیند نامتناهی حاوی داده‌هایی که به طور پیوسته با گذشت زمان تغییر می کنند، در نظر گرفته شود. یعنی خوشه‌های ایجاد شده، در طول زمان به تدریج تغییر خواهند کرد. این ویژگی ناشی از طبیعت پویای جریان داده است. طبیعت خوشه‌ها بر اساس دو مسئله شکل می گیرد. اول زمانی که خوشه‌ها ساخته می شوند و دوم بر اساس افق زمانی مورد نظر کاربر. به طور مثال کاربر ممکن است مایل به بررسی خوشه‌های یک ماه گذشته، یک سال گذشته یا دهه قبل باشد. این خوشه‌ها ممکن است با هم متفاوت باشد. الگوریتم خوشه‌بندی باید انعطاف‌پذیر بوده و بتواند خوشه‌ها را در افق‌های زمانی مورد نظر کاربر بیابد. به دلیل محدودیت تک‌گذره الگوریتم‌های جریان داده، اعمال انعطاف‌پذیری به الگوریتم برای یافتن خوشه‌ها در افق‌های زمانی مختلف، بسیار دشوار است. مثلا ممکن است کاربر مایل به مقایسه خوشه‌های این لحظه با لحظه قبل باشد.

CluStream، در مرحله اول خلاصه‌های آماری از جریان داده به صورت ساختارداده‌هایی به نام ریزخوشه[۱۰]، به روش برخط، ایجاد می‌کند. مرحله دوم بر اساس افق زمانی مورد نظر کاربر، از ریزخوشه‌ها استفاده می‌کند و خوشه‌های این دوره زمانی را مشخص می‌کند و چون به صورت برون‌خط عمل می کند، بسیار سریع خواهد بود. بعلاوه در چهارچوب CluStream تمهیداتی جهت تشخیص و رفع outlier در نظر گرفته شده است. از مزایای CluStream، می‌توان به کیفیت بالاتر خوشه‌ها، انعطاف‌پذیری بسیار بالا در یافتن خوشه‌ها در هر افق زمانی مورد نظر کاربر و مقایسه آنها، سرعت بالا به دلیل تقسیم الگوریتم به دو بخش برخط و برون‌خط، استفاده از تکنیک‌های تشخیص outlier اشاره کرد. مشکل CluStream این است که نویسنده فرض کرده است، توزیع ورود داده‌ها همواره نرمال است. در صورتی که طبیعت جریان داده داشتن توزیع متغیر است.

۳.۳. HPSTREAM

بسیاری از جریان‌های داده، حاوی داده‌های با ابعاد بسیار بالا[۱۱] هستند. خوشه‌بندی داده‌های با ابعاد بالا ذاتا بسیار پیچیده است حتی اگر پایگاه داده ایستا باشد. زیرا به فضا و زمان زیادی نیاز دارد. بنابراین در حالت کار با جریان داده تعداد ابعاد بالا چالش بزرگی در خوشه‌بندی ایجاد می‌کند. HPSTREAM[۲]، الگوریتمی برای خوشه‌بندی جریان داده با ابعاد بالا است. به این منظور HPSTREAM، خوشه‌بندی projected را بر جریان داده اعمال کرده است. اگر خوشه‌ها را برای زیرمجموعه‌ای از ابعاد و نه همه آنها پیدا کنیم، خوشه‌ها projected نامیده می شوند. این الگوریتم در مقابل تعداد ابعاد و اندازه جریان داده بسیار، مقیاس‌پذیری بالایی دارد و کیفیت بهتری را نسبت به الگوریتم‌های پیش از خود ارائه می‌دهد. روش‌های پیشین برای خوشه‌بندی projected، به دلیل چندمرحله ای بودن، و پیچیدگی زمانی خیلی بالا مناسب کار با جریان داده نیستند. در این الگوریتم برای پیاده-سازی خوشه‌بندی projected، به ازاء هر خوشه برداری بیتی در نظر گرفته می‌شود. که تعداد ابعاد این بردار با تعداد ابعاد داده‌ها یکسان است. هر بعد از داده که در خوشه‌بندی حضور دارد مقدار بعد متناظر از بردار بیتی آن ۱ و در غیراینصورت ۰ است. با ورود هر داده جدید به یک خوشه، بردار بیتی خوشه تغییر می‌کند.

HPSTREAM از یک ساختار به نام "خوشه فرسودگی[۱۲]" برای نمایش هر خوشه استفاده می‌کند. خوشه فرسودگی از ۳مقدار تشکیل شده است. دو مقدار دو بردار حاصل از مجموع وزندار و مربع مجموع وزندار نقاط درون خوشه است. مقدار سوم مجموع وزن نقاط درون خوشه است. HPSTREAM فرض کرده که هر نقطه وزنی دارد که به وسیله تابع f(t)، تعیین می شود. تابع f(t) را تابع فرسودگی۱۳ می‌نامیم. تابع f(t)، اکیدا نزولی و مقدار آن در بازه (۰,۱) است و به طور یکنواختی با گذشت زمان نزول[۱۴] می‌کند. در این الگوریتم تابع f(t)، نمایی انتخاب شده است، چون تابع نمایی بسیار مناسب حالتی است که بتدریج داده‌های گذشته را کنار بگذاریم. علت اعمال فرسایش طبیعت پویای جریان داده است. داده‌های تازه‌وارد باید نقش موثرتری در خوشه‌بندی جریان داده ایفا کنند.

مزیت این الگوریتم اعمال خوشه‌بندی projected، به جریان داده‌های با ابعاد بالا است. همچنین کیفیت و مقیاس‌پذیری خوبی ارائه داده است.

۴.۳. مقایسه الگوریتم‌های خوشه‌بندی جریان داده

الگوریتم STREAM، نمایش فشرده‌ای از اطلاعات را بکار می‌گیرد. به اینصورت که همواره تنها مراکز خوشه‌ها را نگه می‌دارد و داده‌های ورودی خوشه‌بندی شده، دور ریخته می‌شوند. مشکل این روش افزایش زمان مرحله دوم خوشه‌بندی- یعنی اعمال خوشه‌بندی روی مراکز نگه داشته شده- در صورت طولانی‌بودن جریان داده و در نتیجه بالا بودن تعداد بخش‌های داده‌های ورودی می‌باشد. این الگوریتم، نامتناهی بودن جریان داده و تغییر پیوسته آن را با گذشت زمان، در نظر نگرفته است. CluStream، فرآیند خوشه‌بندی را به دو بخش برخط و برون خط تقسیم می‌کند. خلاصه سازی

داده به صورت برخط و خوشه‌بندی خلاصه داده‌ها، در مرحله برون‌خط انجام می‌شود. این الگوریتم، امکان یافتن خوشه‌های رسیده در افق‌های زمانی مختلف را در اختیار کاربر قرار می‌دهد. بنابراین انعطاف‌پذیری بسیار بالایی نسبت به تغییرات زمانی خوشه‌ها خواهد داشت. بعلاوه برخلاف روش STREAM، که هیچ تدبیری برای تشخیص و حذف outlier نیندیشیده، تکنیکی برای تشخیص outlier در نظر می‌گیرد. نتایج نشان می‌دهد که روش CluStream، نسبت به روش STREAM، کیفیت خوشه‌بندی بالاتری ارائه می‌دهد[۱۰].

HPStream، مفهوم فرسودگی و خوشه‌بندی projected را بخوبی پیاده‌سازی کرده است. در حالتی که تعداد ابعاد نقاط داده بسیار بالاست، خوشه-بندی projected موثر خواهد بود. بویژه در حالت خوشه‌بندی جریان، که سرعت محاسبات اهمیت دارد، این مسئله بیشتر اهمیت می‌یابد.

کیفیت خوشه‌های ایجاد شده توسط این الگوریتم از دو الگوریتم STREAM و CluStream، بالاتر است[۱۰]. برخی ویژگی‌های این سه الگوریتم در جدول شماره (۱) مقایسه شده است. هنوز الگوریتمی بهینه در زمینه خوشه‌بندی جریان داده ارائه داده نشده است. و این به دلیل وسعت مسئله و محدودیت‌هایی است که طراحان الگوریتم جریان داده با آن روبرو هستند

جدول شماره (۱) مقایسه الگوریتم‌های خوشه بندی جریان داده

نام الگوریتم	امکان تشخیص و حذف outlier	انعطاف پذیری در برابر زمان	کار با جریان داده با ابعاد بالا	منسوخ شدن تدریجی داده‌ها با زمان	سرعت پردازش الگوریتم‌ها نسبت به هم
STREAM	خیر	خیر	خیر	خیر	خوب
CluStream	بله	بله	خیر	خیر	بالا (به دلیل دو مرحله برخط و برون‌خط در پردازش)
HPSTREAM	بله	بله	بله	بله	بالا (به دلیل تکنیک خوشه‌بندی projected)

۴- خوشه‌بندی جریان متن

مسئله خوشه‌بندی جریان متن، نسبت به خوشه‌بندی جریان داده‌های عددی، در آغاز راه است و به تازگی مورد توجه محققان بیشتری قرار گرفته است. اکثر روش‌های مبتنی بر مشابهت از روش TF-IDF استفاده می‌کنند. این روش، مبنای تشابه یک متن و یک خوشه را تعداد تکرار کلمات مشابه و تعداد تکرار کلمه در کل مجموعه می‌داند. در بخش ۳.۴ خواهیم دید که روش مناسب‌تری به نام مدل هموارساز مفهومی می‌تواند با در نظر گرفتن ارتباط معنایی متن و خوشه، کیفیت خوشه‌بندی را بهبود دهد.

در این بخش به معرفی الگوریتم‌های خوشه‌بندی جریان متن می‌پردازیم.

۱.۴. streaming OSKM[۱۵]

اگرچه جریان داده‌های متن به صورت جریانی پیوسته در نظر گرفته می شوند، این الگوریتم[۸]، جریان داده را به صورت بخش، بخش خوشه‌بندی می کند- در واقع خوشه‌بندی یک فرآیند شبه-پیوسته[۱۶] خواهد بود. در این روش برداری به نام تاریخچه وجود دارد که در هر مرحله با توجه به این بردار داده‌های مراحل قبل را بر داده‌های جدید، تاثیر می‌دهد. اندازه هر بخش برحسب اندازه بافر تعیین می شود. در این الگوریتم تابع هدف علاوه بر سعی در حداکثرسازی تشابه بردارهای داده درون خوشه با مرکز آن، سعی در ماکزیمم سازی تشابه بردارهای تاریخچه درون خوشه با مرکز آن خوشه دارد. با بردارهای تاریخچه مثل بردارهای داده جدید رفتار می کنیم با این تفاوت که برای آنها وزن قائل می‌شویم. وزن هر بردار تاریخچه با ورود بخش بعدی از جریان داده، به صورت نمایی نزول می کند. و بنابراین می‌بینیم که این وزن نقش یک فاکتور فراموشی را دارد و به دلیل ماهیت متغیر جریان داده به چنین فاکتوری نیاز داریم.

این الگوریتم از روش k-means و تشابه کسینوسی، که یک روش متداول برای خوشه‌بندی متون با ابعاد بالاست استفاده می‌کند. در این الگوریتم، هر متن به صورت یک بردار با طول واحد و با ابعاد بسیار بالا نمایش داده می‌شود. هدف آموزش نزدیکترین مرکز خوشه به نقطه تازه‌وارد x_n است. به این ترتیب این مرکز در جهت درست به نقطه x_n نزدیکتر می شود.

TF-ICF[۱۷] .۲.۴

نوعا الگوریتم‌های خوشه‌بندی مدل فضای برداری(VSM[۱۸])[۵] را برای نمایش متون استفاده می‌کنند. VSM، کلمات را با متن‌ها مرتبط می‌سازد و هر متن به صورت یک بردار از کلمات نمایش داده می‌شود. چون کلمات مختلف درجه اهمیت متفاوت دارند، یک وزن به هر لغت نسبت داده می‌شود[۵]. وزن کلمه‌ها، اغلب براساس تعداد تکرار کلمه در یک متن یا مجموعه‌ای از متن‌ها محاسبه می‌شود. روش‌های تعیین وزن بسیاری برای کلمات معرفی شده است[۵]. در اکثر روش‌های موجود فرض بر این است که همه مجموعه داده موجود و ایستا است. در رهیافت رایج و پرکاربرد TF-IDF، باید از ابتدا تعداد متون حاوی هر کلمه مشخص باشد(DF). این مسئله یک دانش اولیه از داده‌ها را می‌طلبد. بعلاوه امکان تغییر مجموعه در طول محاسبه وجود ندارد.

نیاز به آگاهی از کل مجموعه داده، باعث محدودیت بکارگیری این طرح(TF-IDF)، در کاربردهایی که جریان داده پیوسته مورد تحلیل قرار می‌گیرد، خواهد شد. به ازای هر متن تازه وارد، این محدودیت موجب بروزرسانی فرکانس متن بسیاری از کلمات خواهد شد. در واقع وزن تمام کلماتی که با متون قبلی وارد شده‌اند باید تغییر کند. بنابراین طرح TF-IDF در مورد جریان‌داده‌های پویا قابل استفاده نیست[۵]. [۵]، طرحی برای وزن‌گذاری کلمات به نام TF-ICF ارائه داده است. در این طرح نیازی به اطلاعات راجع به تعداد تکرار کلمه در کل متون نیست. بنابراین زمان پردازش جریان داده خطی می‌شود.

TF-ICF، به جای استفاده از معکوس فرکانس متن، از معکوس فرکانس تکرار در نمونه استفاده کرده است. یعنی در چند متن از مجموعه متن‌های نمونه مورد نظر، این کلمه وجود دارد .TF-ICF فرض می‌کند فرکانس متن برای کلمات در یک زمینه خاص به زبان انگلیسی از یک توزیع تبعیت می‌کند.

الگوریتم TF-ICF سه پرسش زیر مطرح به شرح زیر مطرح کرده است.

- آیا می‌توان توزیع تعداد تکرار متن یک مجموعه کوچکتر را برای مجموعه‌ای بزرگتر تخمین زد؟
- آیا می‌توان توزیع فرکانس متن یک مجموعه را برای دیگری تخمین زد؟
- آیا می‌توان مجموعه‌ای یافت که تقریبا کل کلمات رایج در نوشتن در آن به کار رفته باشد؟

اگر پاسخ سه پرسش بالا مثبت باشد، می‌توان از روش TF-ICF استفاده کرد. به این صورت که به جای معکوس تعداد تکرار در متن از معکوس تعداد تکرار در نمونه استفاده شود. بنابراین نیازی به بروزرسانی بردارهای متون گذشته با رسیدن متون جدید نیست. متاسفانه یافتن پاسخ این سوالات بینهایت دشوار است و بهرحال این روش یک روش تقریبی است

۳.۴. الگوریتم OCTS

اخیرا، [۱۴]، یک متن را سرشار از کلمات مستقل از رده[۱۹] یا "کلمات کلی[۲۰]" و حاوی تعداد اندکی کلمات وابسته به رده[۲۱] یا "کلمات هسته‌ای[۲۲]" دانسته است. این مسئله منجر به کاهش کیفیت خوشه‌بندی می‌شود[۱۱]. الگوریتم‌هایی که از رهیافت TF-IDF، استفاده می‌کنند، چون معیار خوشه‌بندی را تعداد کلمات مشترک بین متن تازه‌وارد و خوشه‌های موجود، می‌دانند، گرفتار این مشکل هستند.

[۱۳]، نشان داده است که مدل هموارساز مفهومی[۲۳] بهتر از TF-IDF عمل می‌کند. [۱۴]، نیز نتایج خوبی از بکارگیری مدل هموارساز مفهومی برای خوشه‌بندی متن ارائه داده است. ایده اصلی رهیافت هموارساز مفهومی، چشم‌پوشی از کلمات کلی(از قبیل کلمات توقف[۲۴]، مثل am، is، that و... که ممکن است در دو متن، متعلق به رده‌های موضوعی مختلف، مشترک باشند) و توجه خاص به کلمات هسته‌ای یافت شده در متن است(از قبیل کلمات موضوعی مرتبط که در یک متن دیده می‌شود). به عبارت دیگر، برای اینکه بتوانیم میزان تشابه متن تازه‌وارد را با هریک از خوشه‌ها بسنجیم و آن را به نزدیکترین خوشه نسبت دهیم، باید بتوانیم ارتباط معنایی کلمات درون متن و کلمات خوشه‌ها را بررسی کنیم و تنها به اشتراک کلمات توجه نکنیم.

۱.۳.۴. مدل هموارساز مفهومی

بسیاری از رهیافت‌های پیشین، برای نمایش ویژگی‌های یک متن، از روش استخراج کلمه و یک بردار از کلمات تکی درون متن استفاده می‌کنند. در چنین مدل‌هایی ممکن است کلمات معنی مبهمی داشته باشند. در مقابل، مدل هموارساز مفهومی از "عبارات چند کلمه‌ای[۲۵]" برای نمایش ویژگی یک متن استفاده می‌کند. عبارت چندکلمه‌ای نمایشگر موضوع متن(امضای موضوع[۲۶]) است. به عنوان مثال عبارت چندکلمه‌ای "fixed star" (نشانگر سیاره)، معنی واضحی دارد اما مشخص نیست کلمه تکی "star" بر اجرام آسمانی دلالت دارد یا یک ستاره سینما۲۷. در مدل هموارساز مفهومی در مرحله آموزش۲۸(یعنی پیش از شروع اجرای الگوریتم)، احتمال $p(w|t_k)$، یعنی احتمال ترجمه عبارت چندکلمه‌ای t_k به کلمات w درون لغت نامه

محاسبه می‌شود تا هنگام اجرای الگوریتم، با توجه به این احتمالات بتوانیم در مورد ارتباط معنایی بین کلمات و عبارات چندکلمه‌ای متن تازه‌وارد و متن‌های درون هر خوشه اظهار نظر کنیم و نزدیکترین خوشه به متن جدید را بیابیم.

به عنوان مثال اگر کلمه "planet"، تعداد تکرار بالایی در یک متن حاوی عبارت چندکلمه‌ای "fixed star"، داشته باشد، پس احتمالا کلمه "planet" و عبارت چندکلمه‌ای "fixed star" باید ارتباط معنایی با هم داشته باشند. پس باید اینگونه ارتباطات را یافته و درجه‌ای از ارتباط را به آن‌ها نسبت دهیم.

بنابراین اگر در فرآیند خوشه‌بندی، با متنی حاوی عبارت چندکلمه‌ای "fixed star" روبرو شویم(و نه "planet")، می توانیم یک احتمال نسبی را به ارتباط کلمه "planet" و آن متن نسبت بدهیم.

۴.۳.۲. معرفی الگوریتم OCTS

[۱۲]، الگوریتمی به نام OCTS، برای خوشه‌بندی جریان متن معرفی کرده است که به ارتباط معنایی متن‌ها در خوشه‌بندی توجه می‌کند و به جای طرح TF-IDF، از مدل هموارساز مفهومی استفاده می‌کند. الگوریتم OCTS برای پیاده‌سازی مدل هموارساز مفهومی، از استخراج عبارات چندکلمه‌ای استفاده کرده است. فرض کنید، متن تازه‌وارد حاوی کلمه wi است و این کلمه در خوشه cj وجود ندارد. در صورتی که کلمه wi، با عبارات چندکلمه‌ای درون خوشه ارتباط داشته باشد، احتمال مشابهت متن تازه‌وارد و cj افزایش می‌یابد.

در حالت برون‌خط، OCTS، ابتدا متن‌های ذخیره شده بر دیسک را به عنوان مجموعه داده آموزش[۲۹] می‌خواند. کلمات و عبارات مجموعه آموزش را استخراج کرده و مدل ترجمه را می‌سازد. یعنی احتمالات ترجمه $p(w|t_k)$ را محاسبه می‌کند[۱۲]. به کمک اولین k متن دریافت شده، k خوشه اولیه را می‌سازد(k تعداد خوشه‌های موردنظر است). سپس فرآیند برخط آغاز می‌شود. با دریافت هر متن جدید، براساس احتمالات محاسبه شده، احتمال مشابهت متن و هریک از خوشه‌ها محاسبه می‌شود. سپس متن جدید به نزدیکترین خوشه نسبت داده می‌شود. البته در صورتی که تشابه متن با نزدیکترین خوشه، از یک حدآستانه کمتر باشد، غیرفعال‌ترین خوشه دور ریخته می‌شود و به جای آن خوشه‌ای حاوی اطلاعات متن جدید ساخته می‌شود. منظور از غیرفعال‌ترین خوشه، خوشه‌ای است که در گذشته‌ای دورتر از سایر خوشه‌ها متن جدید دریافت کرده است. باید دانست علت درنظرگرفتن حدآستانه تشابه و حذف غیرفعال‌ترین خوشه، طبیعت پویای جریان و تغییر آن با زمان است. یعنی ممکن است متن تازه‌وارد که نشانگر جریان جدیدی از متن‌ها با یک موضوع جدید باشد پس قدیمی‌ترین خوشه را حذف می‌کنیم و به جریان جدید توجه می‌کنیم.

۴.۳.۳. ابزارهای استخراج عبارات از متن

[۱۲,۱۴]، برای استخراج عبارات از Xtract[۹] استفاده کرده‌اند. Xtract، نوعی ابزارآماری است که می‌تواند بدون دانش خارجی، عبارات اسمی را در مجموعه‌ای از متن‌ها با دقت ۸۰درصد استخراج کند. بتازگی Xtract، در یک بسته ابزاری به نام dragon Toolkit[۱۵] پیاده‌سازی شده است. Dragon Toolkit، یک بسته مبتنی برجاواست که برای استفاده دانشگاهی در بازیابی اطلاعات و متن‌کاوی(شامل رده‌بندی متن، خوشه‌بندی متن، خلاصه‌سازی متن، و مدلسازی موضوعی) توسعه یافته است.

۴.۴. مقایسه الگوریتم‌های خوشه‌بندی جریان متن

الگوریتم streaming OSKM[۸]، یک الگوریتم مبتنی بر مشابهت است و از طرح TF-IDF استفاده می‌کند. بنابراین کیفیت خوشه‌بندی ضعیفی ارائه می‌دهد[۱۲]. از سوی دیگر الگوریتم TF-ICF[۵] اگرچه سعی در غلبه بر مشکل بکارگیری TF-IDF برای جریان داده دارد، اما به دلیل اینکه از راه‌حلی تقریبی استفاده کرده است، عملا راه‌حل کامل و دقیقی ارائه نداده است. در این روش با در نظر گرفتن یکسری فرضیات، و بررسی ویژگی‌های تعدادی از مجموعه‌های داده، معکوس فرکانس متن تقریب زده شده است.

بهترین الگوریتم ارائه شده، الگوریتم OCTS[۱۲] می‌باشد. این الگوریتم با بکارگیری مدل هموارساز مفهومی، کیفیت خوشه‌بندی جریان متن را بهبود بخشیده است. خلاصه ویژگی‌های الگوریتم‌های بررسی شده در زمینه خوشه‌بندی جریان متن در جدول شماره(۲) مشاهده می‌شود.

جدول شماره (۲) مقایسه الگوریتم‌های خوشه بندی جریان متن

الگوریتم	رهیافت مبنی بر	استفاده از طرح TF-IDF	استخراج کلمات و عبارات	پارامترهای موثر بر تشابه متن‌ها و خوشه‌ها
Streaming OSKM	مشابهت	بله	کلمات تکی	کلمات مشترک
TF-ICF	مشابهت	تقریب TF-IDF	کلمات تکی	کلمات مشترک
OCTS	مدل	خیر	کلمات تکی و عبارات چندکلمه-ای	کلمات مشترک - احتمال ترجمه کلمات به عبارات

۵. معیارهای ارزیابی کیفیت خوشه‌ها

دو نوع معیار خارجی[۳۰] و داخلی[۳۱] برای ارزیابی کیفیت خوشه‌بندی وجود دارد. سه معیار خارجی متداول در ارزیابی کیفیت خوشه‌بندی NMI[۳۲]، purity و entropy است. این معیارها زمانی قابل استفاده است که برچسب رده‌ها را در اختیار داشته باشیم. یعنی از پیش نتایج خوشه‌بندی صحیح را در اختیار داشته باشیم و بتوانیم با نتایج حاصل از الگوریتم خود مقایسه کنیم. امروزه معیار اطلاعات انحصاری(MI)، مورد توجه محققان بسیاری قرار گرفته است. صورت نرمال شده آن NMI نامیده می‌شود. که اعداد حاصل از آن در بازه [۰,۱] است. NMI، تشابه آماری بین خوشه‌های ایجاد شده و برچسب‌های از پیش تعیین شده را اندازه‌گیری می‌کند. این معیار مناسب‌تر از دو معیار دیگر، یعنی purity و entropy است. علت این است که مقدار NMI، لزوما با افزایش تعداد خوشه‌ها افزایش نمی‌یابد اما آن دو معیار اینگونه نیستند. . معیار entropy، و purity نیز کیفیت خوشه-بندی را بصورت عددی در بازه [۰,۱] محاسبه می‌کنند. هرچه کیفیت خوشه‌بندی بهتر باشد، مقدار entropy عددی کمتر و purity و NMI، عددی بیشتر ایجاد می‌کنند. در صورتی که نتایج خوشه‌بندی دقیقا با رده‌های موجود مطابقت کند، entropy به صفر و purity، به یک می‌رسد.

معیارهای داخلی مثل تشابه کسینوسی یک تابع هدف را در نظر می‌گیرند. این تابع عددی نشاندهنده‌ی میزان تشابه داده‌های درون خوشه و عدم تشابه داده‌های خوشه‌های مختلف با یکدیگر ایجاد می‌کند. معیارهای خارجی برای داده‌های متنی مناسب‌تراند. الگوریتم‌های stream OSKM و OCTS دو الگوریتم خوشه‌بندی جریان متن از NMI، که معیاری خارجی استفاده کرده‌اند.

۶. نتیجه‌گیری

هریک از الگوریتم‌های شرح داده شده سعی در رفع یکی از مشکلات و نیازهای جریان دارند اما به دلیل محدودیت‌های کار با جریان داده و همچنین جریان متن، هنوز یک الگوریتم بهینه در زمینه خوشه‌بندی جریان ارائه نشده است. افزایش کیفیت بدون کاهش سرعت و نیاز به فضای بیشتر، همچنین ارائه راه‌حل در تشخیص و حذف outlier، مورد نظر محققان است.

۷. مراجع

[۱] Agrawal, C. C. ,Han, J., Wang, J., *"A framework for clustering evolving data streams"*. Proceedings of ۲۹th international conference on very large data bases, pp. ۸۱–۹۲, ۲۰۰۳.

[۲] Agrawal, C. C. Han, J., Wang, J., Yu, P. S, *"A framework for projected clustering of high dimensional data streams"*. Proceedings of ۳۰th international conference on very large data bases,pp. ۸۵۲–۸۶۳ , ۲۰۰۴.

[۳] Daniel.B, *"Requirements for clustering data streams"*. SIGKDD Explorations , ۲۰۰۲.

[۴] In Jae Myung, ۲۰۰۳, ***Tutorial on maximum likelihood estimation***, Journal of Mathematical Psychology, PP. ۹۰–۱۰۰.

[۵] [۵]Joel W.Reed, Yu Jiao , Thomas E.Potok , *"TF-ICF: A New Term Weighting Scheme for Clustering Dynamic Data Streams"*, proceedings of the ۵th International Conference on Machine Learning and Aplications(ICMLA'۰۶), ۲۰۰۶.

[۶] L. O'Callaghan ,S. Guha, N. Mishra, R. Motwani, *"Clustering Data Streams"*, IEEE FOCS conference, ۲۰۰۰.

[۷] L. . O'Callaghan et al, Guha, *"Streaming-Data Algorithms For High-Quality Clustering"*. ICDE Conference, ۲۰۰۲.

[۸] Shi Zhong, *"Efficient streaming text clustering"*. Neural Networks , ۲۰۰۵.

[۹] Smadja, F.Retrieving collocations from text:Xtract. Computational Linguistics, pp. ۱۴۳–۱۷۷, ۱۹۹۳.

[۱۰] Thanawin R, Komkrit U, Kitsana W, *"E-Stream: Evolution-based Technique for Stream Clustering"*. Springer-verlag Berlin Heidelberg,ADMA, pp.۶۰۵-۶۱۵, ۲۰۰۷.

[۱۱] Yubao Liu, Jiarong Cai , Jian Yin, ۲۰۰۷, *An Improved Semantic Smoothing Model for Model-Based Document Clustering,* Eighth ACIS International Conference on Software Engineering, Artificial Intelligence,Networking, and Parallel/Distributed Computing.

[۱۲] Yubao Liu,Jiarong Cai , Jian Yin , and Wai-chee Fu, "*Clustering Massive Text Data Streams by Semantic Smoothing Model"*, jan , Journal of Computer Science and technology, ۲۰۰۸.

[۱۳] Zhou,X., xiaodan,Z., Lin, X.,Song, "*Context-sensitive Semantic Smoothing for the Language Modeling Approach to Genomic IR"*.In:Proc. ACM SIGIR, ۲۰۰۶.

[۱۴] Zhou,X, Zhou,X,and X.Hu."*semantic smoothing for model-based document clustering"*.In:Proc.ICDM , pp. ۱۱۹۳-۱۱۹۸, ۲۰۰۶.

[۱۵] Zhou, X., Zhang, X., and Hu, X.,۲۰۰۷, The Dragon Toolkit, Data Mining & Bioinformatics Lab,ischool at Drexel University, http://www.ischool.drexel.edu/dmbio/dragontool

---

۱ online
۲ Similarity based
۳ Model based
٤ scalable
٥ adaptive
٦ Relevance stamp
۷ batch
۸ clustering
۹ offline
۱۰ microclusters
۱۱ High-dimensional data streams
۱۲ Fading cluster
۱۳ fading
۱٤ decays
۱٥ Online spherical k-means
۱٦ semi-continuous
۱۷ Term frequency-inverse corpus frequency
۱۸ vector space model
۱۹ Class-independent
۲۰ General words
۲۱ Class-specific
۲۲ Core words
۲۳ Semantic smoothing model
۲٤ Stop words
۲٥ Multiword phrase
۲٦ Topic signature
۲۷ Pop star
۲۸ training
۲۹ Training text data set
۳۰ extrinsic
۳۱ intrinsic
۳۲ Normalized mutual information