



دانشکده مهندسی کامپیوتر

بررسی الگوریتم‌های یادگیری ماشین برای جریان‌های متنی

گزارش سمینار کارشناسی ارشد در رشته مهندسی کامپیوتر
گرایش هوش مصنوعی و رباتیک

وحید خرازی

استاد راهنما

دکتر بهروز مینایی بیدگلی

آبان ۹۵



تقدیم به:

پدر و مادر عزیزم که پیغمبر نگاهشان همیشه مرا
چشم و چراغ خواهد بود.

قدردانی

سپاس خداوندگار حکیم را که با لطف بی‌کران خود، آدمی را زیور عقل آراست.
در آغاز وظیفه خود می‌دانم از زحمات بی‌دریغ استاد راهنمای خود، جناب آقای دکتر مینایی، صمیمانه
تشکر و قدردانی کنم که قطعاً بدون راهنمایی‌های ارزنده ایشان، این مجموعه به انجام نمی‌رسید.

وحید خرازی

آبان ۹۵

چکیده

امروزه ما با منابع تولید داده‌های جریانی روبه‌رو هستیم که داده‌ها را با سرعت بالایی تولید می‌کنند. مدل کردن داده‌های حجیم، پیوسته، سریع و متغیر در زمان با توجه به محدودیت‌های زمانی و منابع محاسباتی نیازمند الگوریتم‌های یادگیری است که با یک‌بار مشاهده‌ی داده و با رعایت محدودیت حافظه اصلی، به صورت بلادرنگ پاسخگو باشند. همچنین به دلیل ماهیت متغیر داده‌ها در این فضا، این الگوریتم‌ها باید توانایی تشخیص رانش مفهوم را داشته باشند. به الگوریتم‌هایی که با داده‌های نامحدود، پویا و گذرا کار می‌کنند «الگوریتم‌های یادگیری داده‌های جریانی» گفته می‌شود. حال اگر منبع تولید این داده‌ها، یک تولید کننده محتوای متنی باشد مساله به جهت ابعاد بالای داده‌های متنی و دشواری‌های پردازش زبان‌های طبیعی، پیچیده‌تر می‌شود.

در سال‌های اخیر، هم از نظر تئوری و هم از نظر عملی، الگوریتم‌های یادگیری ماشین و داده‌کاوری، تمرکز خود را بر روی مجموعه‌های داده‌ای ایستا، یکجا، همجنس، پایدار، معین، محدود و در نهایت مدل‌های ایستا معطوف کرده‌اند در حالی که امروزه، بسیاری ابزارهای کاربردی، خجم بسیار زیادی از داده‌های جریانی را با سرعت بالایی تولید می‌کنند. داده‌کاوی و یادگیری از جریان‌های داده‌ای، به ویژه داده‌های متنی که ابعاد بالایی دارند، یکی از موضوعات داغ تحقیقاتی است که می‌تواند کاربردی باشد.

واژگان کلیدی: یادگیری ماشین، داده‌های جریانی، متن کاوی، پردازش متن

فهرست مطالب

ح	فهرست تصاویر
خ	فهرست جداول
۱	فصل ۱: مقدمه
۵	فصل ۲: مروری بر ادبیات
۵	۲-۱ تعاریف اولیه
۵	۲-۱-۱ جریان داده
۶	۲-۱-۲ مفهوم
۷	۲-۱-۳ رانش مفهوم
۸	۲-۲ پنجره‌های زمانی
۸	۲-۲-۱ پنجره نقطه عطفی
۹	۲-۲-۲ پنجره کشویی
۹	۲-۲-۳ پنجره محوشونده
۹	۲-۲-۴ پنجره یک‌بر
۱۰	۲-۳ رویکردهای محاسباتی
۱۰	۲-۳-۱ یادگیری افزایشی
۱۱	۲-۳-۲ یادگیری دوگامی
۱۲	۲-۴ اعتبار سنجی
۱۳	۲-۵ نرم‌افزارهای کاربردی

۱۴	مخزن‌های داده
----	---------------

فصل ۳: روش‌های یادگیری در داده‌های جریانی

۱۶	درخت هافدین
۱۸	رده‌بند بیزین
۱۹	شبکه عصبی
۲۰	ماشین‌های بردار پشتیبان
۲۰	رده‌بند نزدیک‌ترین همسایگی
۲۱	مجمع رده‌بندها
۲۲	۳-۶-۱ مجمع وزنی رده‌بندها
۲۴	۳-۶-۲ درخت‌های سازگار تصمیم مبتنی بر رویکرد یکی در برابر بقیه
۲۴	۳-۶-۳ مجمع متا-دانش

فصل ۴: جمع‌بندی و نتیجه‌گیری

۲۶	۴-۱ مروری بر روش‌های یادگیری
۳۱	۴-۲ کارهای آینده در جریان‌های متنی
۳۱	۴-۲-۱ انتخاب پویای ویژگی
۳۲	۴-۲-۲ کاوش در جریان‌های متنی

فهرست تصاویر

- ۲-۱ مثالی از تغییر تدریجی منبع S_1 به منبع S_2 . در این شکل کلاس y_1 با رنگ خاکستری و کلاس y_2 با رنگ مشکی مشخص شده است. ۷
- ۲-۲ پنجره‌های مختلف زمانی ۱۰
- ۲-۳ الف- رویکرد افزایشی به یادگیری. ب- رویکرد دو گامی ۱۱
- ۴-۱ مقایسه و نمایش ارتباط روش‌های سنتی یادگیری ماشین با روش‌های یادگیری در داده‌های جریانی ۲۷
- ۴-۲ یک مثال که نشان می‌دهد چگونه ذات ویژگی‌های بااهمیت طی زمان می‌تواند تغییر کند. ۳۱

فهرست جداول

- ۱-۱ مقایسه بین روش‌های یادگیری سنتی و روش‌های یادگیری در محیط داده‌های جریانی . . . ۲
- ۴-۱ مقایسه ظرفیت‌های الگوریتم‌های مختلف یادگیری در داده‌های جریانی ۲۹
- ۴-۲ مزایا و محدودیت‌های روش‌های رده‌بندی برای داده‌های جریانی ۳۰

فصل ۱

مقدمه

روش‌های سنتی یادگیری ماشین، معمولاً روی مخزن‌های داده‌های ساکن و ایستا تمرکز کرده‌اند ولی توسعه فناوری، باعث شده است که داده‌های جریانی تولید شوند و راهی که مردم داده را ذخیره، و پردازش می‌کردند عوض شده است. امروزه بسیاری از سازمان‌ها، میزان زیادی از داده را با سرعت بالایی تولید می‌کند. به عنوان مثال، هر روز، گوگل بیشتر از سه و نیم میلیون جستجو را پاسخ می‌دهد^۱، ماهواره‌های ناسا بیشتر از ۴ ترابایت عکس تولید می‌کنند^۲ و فروشگاه‌های المارت بیشتر از ۲۰ میلیون تراکنش را ثبت می‌کند. مساله‌ای جدید تحقیقاتی داده‌کاوی و یادگیری ماشین این است: «چگونه ما می‌توانیم مجموعه‌ی داده‌ی نامحدودی که سریع تولید می‌شود و در طول زمان تغییری می‌کند را ضمن در نظر گرفتن محدودیت‌های زمانی و منابع محاسباتی مدل کنیم؟».

این مجموعه‌های داده بسیار بزرگ‌تر از آن هستند که روی حافظه‌ی اصلی جا شوند و باید روی حافظه‌های جانبی ذخیره بشوند. به علاوه، دسترسی تصادفی به این داده‌ها که معمولاً در روش‌های سنتی فرض می‌شد که امکان پذیر است، این‌جا بسیار هزینه‌بر است. یک هدف کاوش داده‌های جریانی، ساختن یک فرایند یادگیری است که به صورت خطی با افزایش تعداد نمونه‌ها رشد کند. به هر حال از آنجایی که داده‌ها به شکل پیوسته در گذر هستند، مدلی که ما از روی داده‌های قدیمی می‌سازیم شاید به درد داده‌های جدید نخورد و باید اقر داده‌های منقضی‌شده از بین برود. این که مدل را با داده‌های جدید دوباره آموزش دهیم نیز فکری ناکارآمد و

^۱<http://www.internetlivestats.com/google-search-statistics/>

^۲<http://data.nasa.gov/about/>

جدول ۱-۱: مقایسه بین روش‌های یادگیری سنتی و روش‌های یادگیری در محیط داده‌های جریانی

محدودیت / رویکرد	سنتی	داده‌های جریانی
تعداد مشاهده داده	چندبار	یک بار
زمان	نامحدود	بلادرنگ
حافظه	نامحدود	محدود
تعداد مفهوم	یک مفهوم	چندین مفهوم
تعداد مفهوم	دقیق	تقریبی

ناکافی است پس یک هدف کاوش داده‌های جریانی می‌تواند این باشد که چطور مدل را به صورت افزایشی با دیدن نمونه‌های جدید بروز رسانی کنیم.

داده‌های جریانی با حجم نامتناهی، با اهمیت بودن ترتیب و زمان وقوع و تغییرات پویا شناخته می‌شوند. مثلاً گوگل، میلیون‌های جستجو را روزانه پردازش می‌کند که هر کدامشان در یک زمان مشخص انجام شده‌اند و این جستجوها در طول زمان و براساس موضوعات داغ روز تغییر می‌کنند. جدول ۱-۱ تفاوت ویژگی‌های بین الگوریتم‌های داده‌های جریانی و الگوریتم‌های سنتی را نشان می‌دهد. به عبارت دیگر، اگر بخواهیم محدودیت‌های الگوریتم‌ها در فضای داده‌های جریانی را برشمریم باید از چهار محدودیت نام ببریم:

- یک‌بار مشاهده داده:^۳ برخلاف روش‌های یادگیری سنتی که می‌توانستند مجموعه‌ی داده را مکرراً چند بار پیمایش کنند. الگوریتم‌ها در داده‌های جریانی فقط یک‌بار می‌توانند داده را مشاهده کنند و برگشت به عقبی وجود ندارد. دلیل این موضوع هم آن است که خواندن/نوشتن روی حافظه‌های جانبی بسیار هزینه‌بر تر از حافظه اصلی است. این محدودیت زیادی است و اگر به الگوریتم‌ها اجازه دهیم به جای یک داده چند داده را ببینند و اجازه داشته باشند که چند نمونه به شکل کوتاه مدت ذخیره کنند اوضاع کمی بهتر می‌شود. به عنوان مثال یک الگوریتم می‌تواند یک دسته از نمونه‌ها را برای پردازش‌های داخلی ذخیره کند اما در نهایت باید با گرفتن یک دسته جدید از داده، داده‌های گذشته را پاک کند.

- پاسخ بلادرنگ:^۴ بسیاری از کاربردهای داده‌های جریانی نظیر پیش‌بینی بازار بورس، به پاسخ بلادرنگ نیاز دارند. میزان زمان مورد نیاز برای پردازش داده‌های و رسیدن به تصمیم باید کم باشد.

^۳ Single-pass

^۴ Real-time Response

● محدودیت حافظه: حجم داده‌ای که می‌رسد خیلی بزرگ است و حتی ممکن است نامحدود باشد و ما می‌توانیم تنها تعداد کمی یا خلاصه‌ای از داده‌های جریانی را ذخیره کنیم و ممکن است دیگر به اصل داده دسترسی نداشته باشیم. البته روش‌های تحمینی می‌توانند قابل قبول باشند.

● تشخیص رانش مفهوم: رانش مفهوم زمانی است که الگوها (یا توزیع داده) در طی زمان تغییر پیدا می‌کند.

حال اگر منبع تولید جریان داده، منبعی باشد که متن تولید می‌کند مساله از جهاتی بغرنج‌تر می‌شود. پردازش و مدل‌کردن داده‌های متنی به دلیل غیر ساختیافته بودن^۵، خطاهای سطح بالا^۶ و وجود در فرمت‌های گوناگون بسیار پیچیده‌تر از سایر داده‌هاست. نفرین ابعاد^۷ یکی از معضلات کار با داده‌های متنی است. در برخی از کاربردها، تعداد ویژگی‌هایی که از یک متن استخراج می‌شود چیزی بیش‌تر از دویست‌هزار ویژگی است. بنابراین لازم است که الگوریتم‌هایی در برابر داده‌های جریانی متنی به کار گرفته شوند که قابلیت سازگاری با این تعداد بسیار زیاد ویژگی را داشته باشند و یا این که با روش‌هایی نظیر پیش‌پردازش، انتخاب ویژگی و یا کاهش ابعاد، این داده‌ها را طوری به ورودی الگوریتم‌ها داد که محدودیت‌های زمانی و حافظه‌ای رعایت شود. مضاف بر همه‌ی این مشکلات، در داده‌های متنی تعداد بسیار بیشتری از رویدادهای تعجب‌آور و تغییر موضوعات شگفت‌انگیز مشاهده می‌شود. این موضوع باعث می‌شود که کاوش و یادگیری در داده‌های متنی یک وظیفه دلهره‌آور باشد.

در این فصل به معرفی فضای کار در الگوریتم‌های یادگیری برای داده‌های جریانی پرداختیم، به علاوه مشکلات پیش‌رو زمانی که منبع داده‌ی ما یک منبع تولید کننده متن باشد را معرفی کردیم. در فصل ۲ ابتدا به مرور ادبیات یادگیری در داده‌های جریانی خواهیم پرداخت و مفاهیمی نظیر منبع داده، مفهوم و رانش مفهوم را به شکل ریاضی تعریف می‌کنیم و سپس رویکردهای مختلف در برابر داده‌های جریانی را معرفی می‌کنیم. در این فصل علاوه بر ارایه تعاریف اولیه، مروری بر نرم‌افزارهای داده‌کاوی و یادگیری در فضای جریانی خواهیم داشت و مخزن‌های داده‌ی در دسترس را جهت تحقیقات و آزمایش‌های آتی معرفی خواهیم کرد. در فصل ۳ به معرفی الگوریتم‌های یادگیری ماشین در فضای داده‌های جریانی خواهیم پرداخت. این الگوریتم‌ها عموماً نسخه‌های توسعه‌یافته از الگوریتم‌های سنتی یادگیری ماشین هستند و با چالش‌ها و محدودیت‌های داده‌های جریانی سازگار

^۵Unstructured^۶High Level of Noise^۷Curse of Dimensionality

شده‌اند. به طور کلی در این فصل الگوریتم‌هایی را معرفی خواهیم کرد که تغییر یافته الگوریتم‌های سنتی درخت تصمیم، یادگیرنده‌های بیز، شبکه‌های عصبی، ماشین‌های بردار پشتیبان و مجمع‌های رده‌بندی هستند. نهایتاً در فصل ۴ و پایانی این گزارش، ابتدا به جمع‌بندی روش‌های یادگیری ارایه‌شده و مقایسه و نمایش نسبت الگوریتم‌ها با پدران سنتیشان می‌پردازیم سپس مزایا و معایب هر کدام را مطرح می‌کنیم و مشخص خواهیم کرد که کدام روش‌ها می‌توانند برای منابع جریانی داده‌های متنی به کار گرفته شوند. در نهایت نیز به جمع‌بندی کار و معرفی تحقیقات آینده، نظیر انتخاب پویای ویژگی برای داده‌های جریانی خواهیم پرداخت.

فصل ۲

مروری بر ادبیات

در فصل پیش، به معرفی فضای یادگیری در جریان‌های داده‌ای پرداختیم و ویژگی‌های این الگوریتم‌ها و محیط اجرای آن‌ها را بررسی کردیم و مقایسه‌ای از محدودیت‌های این الگوریتم‌ها با الگوریتم‌های سنتی ارائه دادیم. در این فصل به مقدمات اصلی کار می‌پردازیم و تعاریفی نظیر مفهوم، منبع داده، رانش مفهوم و پنجره‌ی زمانی را مطرح می‌کنیم. به علاوه رویکردهای کلی به الگوریتم‌های یادگیری محیط داده را نیز بررسی خواهیم کرد و در نهایت نرم‌افزارها و داده‌های مورد استفاده در فضای تحقیقاتی را معرفی خواهیم کرد.

۲-۱ تعاریف اولیه

در این بخش به معرفی چند تعریف اولیه نظیر منبع داده، مفهوم، رانش مفهوم می‌پردازیم:

۲-۱-۱ جریان داده

یک جریان داده^۱ به شکل یک دنباله از نمونه‌های داده تعریف می‌شود و آن را با نماد DS نشان می‌دهیم: $DS = \{x_1, x_2, \dots, x_t, \dots\}$ که در این رابطه x_i برابر i امین داده‌ی مشاهده شده است. هر نمونه داده‌ی x_i یک برجسب دارد و با $y_i \in Y = \{y_1, y_2, \dots, y_c\}$ نشان داده می‌شود.

^۱Data Stream

۲-۱-۲ مفهوم

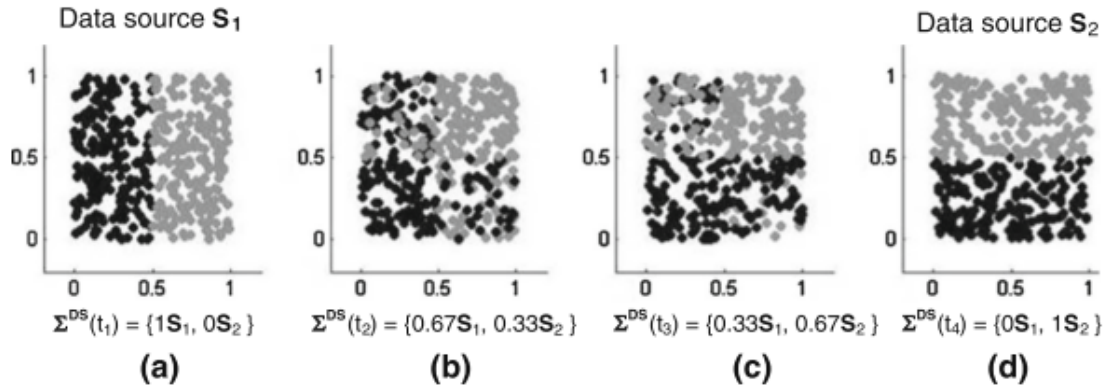
یک مفهوم^۲ یا یک منبع داده، توسط احتمال پیشین برای رده‌ها و تابع توزیع احتمال رده‌ی آن‌ها تعریف می‌شود:

$$S = \{(P(y_1), P(X|y_1)); \dots; (P(y_c), P(X|y_c))\} \quad (2-1)$$

که در این رابطه y بیانگر رده‌ها است. روش‌های سنتی یادگیری ماشین و داده‌کاوی با یک مفهوم کار می‌کند، و این بدان معنی است که تابع توزیع احتمال مجموعه آموزش و مجموعه آزمون یکسان است. این در حالی است که داده‌های جریانی پویا هستند و تعداد زیادی مفهوم دارند. یک مثال از این تغییر مفهوم را می‌توانید حمله‌ی هکری در یک شبکه‌ی کامپیوتری را در نظر بگیرید. توضیح مفهوم داده‌ی «معمولی» با داده‌ی «حمله» در طی زمان تغییر می‌کند، همیشه حمله‌ها در حال انجام هستند ولی انواع و روش‌های آن تغییر می‌کند. یک جریان داده‌ای مانند DS شامل یک مجموعه k عضوی از منبع داده را در نظر بگیرید که در آن منابع داده با s_i نشان داده می‌شود و توزیع این منابع شناخته شده است $i = 1, \dots, k$. در زمان t یک یا چند منبع داده فعال وجود دارد. فرض کنید $w_i(t) \in [0, 1]$ برابر با تاثیر منبع داده S_i در زمان t باشد. توزیع داده‌ی جریان داده‌ی DS در زمان t توسط رابطه زیر مشخص می‌شود:

$$\sum_{DS}^t(t) = \{w(1)S_1, \dots, w(k)S_k\} \quad (2-2)$$

^۲Concept



شکل ۲-۱: مثالی از تغییر تدریجی منبع S_1 به منبع S_2 . در این شکل کلاس y_1 با رنگ خاکستری و کلاس y_2 با رنگ مشکی مشخص شده است.

۲-۱-۳ رانش مفهوم

اگر فرض کنیم که یک منبع داده DS شامل k منبع جریانی مستقل S با وزن w باشند اگر در دو زمان t_1 و t_2 رابطه زیر برقرار باشد می‌گوییم رانش مفهوم^۳ رخ داده است.

$$\sum^{DS}(t_1) \neq \sum^{DS}(t_2) \quad (2-3)$$

در تصویر ۲-۱ می‌توانید یک نمونه از رانش مفهوم که بصری‌سازی شده است را طی زمان‌های مختلف مشاهده کنید. رانش‌های مفهوم به دو نوع زیر طبقه‌بندی می‌شوند:

رانش مفهوم مجازی

اگر رانش مفهوم تأثیر مستقیمی بر حدود تصمیم‌گیری نداشته باشد (احتمال پیشین داده‌ها تغییر نکند) اما رانش بر روی تابع چگالی احتمال اثر بگذارد می‌گوییم که رانش مفهوم مجازی^۴ رخ داده است.

^۳ Concept Drift

^۴ Virtual Concept Drift

رانش مفهوم حقیقی

اگر رانش مفهوم علاوه بر تغییر تابع چگالی احتمال، تأثیر مستقیم بر حدود تصمیم‌گیری بگذارد و احتمال‌های پیشین را تغییر دهد گوییم رانش مفهوم حقیقی^۵ رخ داده است. روش‌های یادگیری در محیط‌های جریان‌های داده‌ای باید به گونه‌ای باشند که رانش مفهوم حقیقی سریعاً تشخیص داده شود.

۲-۲ پنجره‌های زمانی

از آنجایی که داده‌های جریانی امکان دارد بینهایت باشند، تنها مقدور است که بخشی از تمام داده‌ی جریانی را پردازش شود. این بخش مورد توجه از داده توسط یک پنجره زمانی از نمونه‌های داده تعریف می‌شود.

$$W[i, j] = (x_i, x_{i+1}, \dots, x_j) \quad (2-4)$$

بر اساس این تعریف، انواع مختلفی از پنجره زمانی بیان می‌شود: پنجره نقطه عطفی^۶، پنجره کشویی^۷، پنجره محو شونده^۸ و پنجره یک‌بر^۹.

۲-۲-۱ پنجره نقطه عطفی

در پنجره نقطه عطفی، ما به تمام داده جریانی از زمان شروع نمونه اول تا زمان فعلی نمونه t_c توجه می‌کنیم: پنجره به شکل $W[1, c]$ تعریف می‌شود. استفاده از این پنجره نقطه عطفی بدین معناست که تمامی تراکنش‌ها در پنجره به شکل یکسانی برای ما اهمیت دارند، هیچ تفاوتی داده‌های گذشته و حال وجود ندارد. به طور مداوم که داده‌های جریانی تغییر می‌کنند، مدل با استفاده از داده قدیمی نمونه‌ها ساخته می‌شود که حتی ممکن است که با داده‌های جدید ناسازگار باشند. برای تأکید بیشتر روی داده‌های جدیدتر می‌توان از پنجره‌های کشویی، محو شونده و یک‌بر استفاده کرد.

^۵Real Concept Drift^۶Landmark Window^۷Sliding Window^۸Fading Window^۹Tilted Window

۲-۲-۲ پنجره کشویی

در پنجره کشویی که با $W[t_c - w + 1, t_c]$ نشان داده می‌شود ما تنها به w تراکنش اخیر توجه می‌کنیم و بقیه داده‌های جریانی نادیده گرفته می‌شوند. نتیجه کاوش وابسته به اندازه پنجره است که w بیانگر آن است. اگر w خیلی بزرگ باشد و رانش مفهوم رخ دهد، پنجره ممکن است شامل داده‌ها و اطلاعات منقضی شده باشد و دقت مدل کاهش پیدا می‌کند. اگر w کوچک باشد، پنجره ممکن است شامل داده‌های ناکار باشد و مدل دچار بیش‌برازش گردد. کارهای قبلی یک مقدار ثابت را برای اندازه پنجره کشویی در نظر می‌گرفتند که توسط کاربران و با آزمایش تعیین می‌شد. اخیراً چند کار برای ارایه پنجره کشویی انعطاف پذیر ارایه شده است که در آن اندازه پنجره بر اساس دقت مدل تغییر می‌کند. زمانی که دقت زیاد باشد اندازه پنجره بزرگ می‌شود و زمانی هم که دقت کم شود، پنجره کوچک می‌شود.

۲-۲-۳ پنجره محوشونده

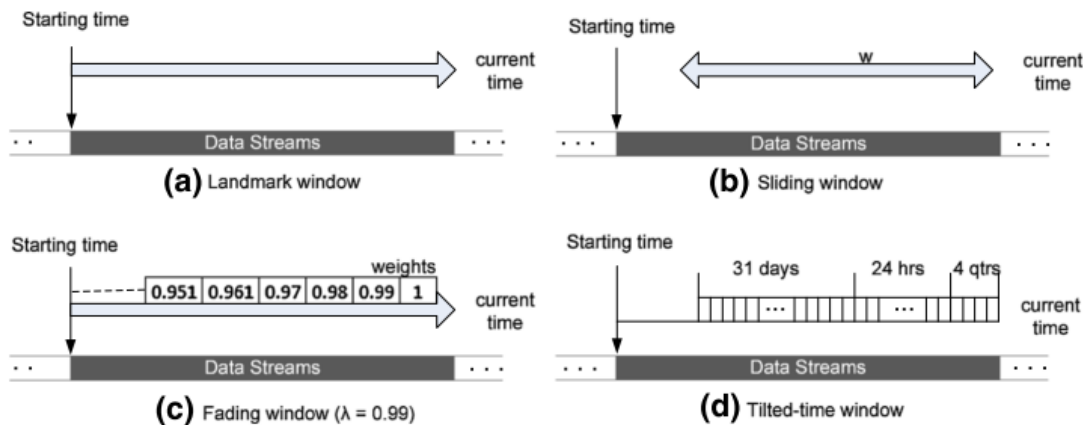
در پنجره‌های محوشونده، هر نمونه از داده‌های جریانی به یک وزن که با زمان رسیدن آن نمونه در ارتباط است شناسایی می‌شود، بنابراین داده‌های جدیدتر می‌رسند وزن بیشتری از داده‌های قدیمی‌تر دارند. با استفاده از پنجره محوشونده، می‌توان تاثیر (اهمیت) داده‌های قدیمی منقرض شده را کاهش داد تا در نتیجه کاوش تاثیری نداشته باشند. معمولاً از یک تابع نمایی نزولی مانند تابع زیر برای پنجره زمانی استفاده می‌شود:

$$f(\Delta t) = \lambda^{\Delta t} \quad (0 < \lambda < 1) \quad (2-5)$$

در این تابع Δt قدمت (سن) نمونه داده‌ی جریانی و برابر با تفاوت زمانی فعلی و زمان مشاهده داده است. پنجره محوشونده نیاز به انتخاب یک پارامتر محوشوندگی مانند λ دارد که معمولاً یک عدد در بازه $[0.99, 1]$ برای کاربردهای واقعی انتخاب می‌شود.

۲-۲-۴ پنجره یک‌بر

پنجره یک‌بر یک نوع پنجره زمانی شبیه پنجره محوشونده و کشویی است. این پنجره در سطوح مختلف ریزدانی که با توجه به تاخر داده شکل گرفته‌اند اعمال می‌شود. پنجره زمانی یک‌بر تقریباً تمام مجموعه داده



شکل ۲-۲: پنجره‌های مختلف زمانی

را ذخیره کرده و یک تعادل بین فضای مورد نیاز برای ذخیره‌سازی و دقت برقرار می‌کند. البته این مدل می‌تواند پس از اجرای طولانی غیر پایدار باشد.

تصویر ۲-۲ چهار نوع مختلف پنجره زمانی را نشان می‌دهد. برای پنجره محو شونده λ برابر با 0.99 تنظیم شده است و وزن نمونه‌ها کاهش می‌یابد. برای پنجره یک‌بر، ما چهار ربع اخیر یک ساعت را ذخیره کرده‌ایم، سپس ۲۴ ساعت گذشته و ۳۱ روز اخیر نمایش داده شده‌اند.

۲-۳ رویکردهای محاسباتی

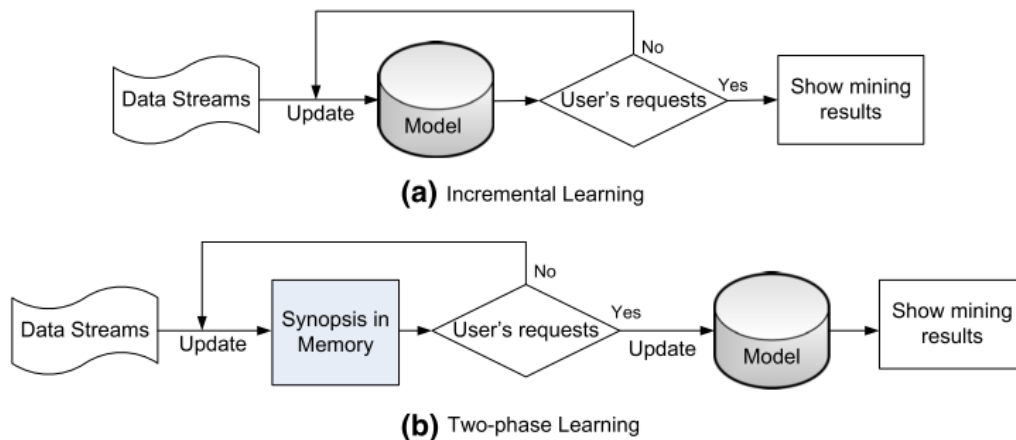
بر اساس انواع مختلف پنجره زمانی، دو رویکرد محاسباتی مختلف برای پردازش داده‌های جریانی وجود دارد.

۲-۳-۱ یادگیری افزایشی

یادگیری افزایشی^{۱۰} یک رویکرد محاسباتی برای داده‌های جریانی است. در این رویکرد، مدل به صورت افزایشی تغییری می‌کند تا با تغییرات در داده‌هایی که در حال آمدن هستند، سازگار شوند. دو طرح کلی برای بروزرسانی مدل وجود دارد: بروزرسانی با هر نمونه داده و بروزرسانی با پنجره. به عنوان نمونه ستریت^{۱۱} و همکارانش یک مجمع از رده‌بندها برای داده‌های جریانی توسعه دادند که این مجمع یک پنجره از داده‌های ورودی را

^{۱۰} Incremental Learning

^{۱۱} Street



شکل ۲-۳: الف - رویکرد افزایشی به یادگیری. ب - رویکرد دو گامی

می‌گیرد و مدل را با تنظیم وزن‌های هر رده‌بند یا جایگزین کردن رده‌بندهای قدیمی با نمونه‌های جدیدشان سازگار می‌کند. شکل ۲-۳ - الف رویکرد یادگیری افزایشی را نمایش می‌دهد. مزیت این رویکرد این است که نتیجه برای هر نمونه (یا پنجره‌ای از نمونه‌ها) در دسترس است ولی به منبع محاسباتی بیشتری نیاز است.

۲-۳-۲ یادگیری دوگامی

یادگیری دوگامی^{۱۲}، که با نام یادگیری آنالین-آفلاین نیز شناخته می‌شود یک رویکرد محاسباتی برای داده‌های جریانی است. ایده بنیادی این است که کاوش بر داده‌ها را به دو بخش تقسیم کنیم. در گام اول (گام آنالین)، چکیده‌ای از داده در لحظه ایجاد می‌شود. در گام دوم (گام آفلاین)، بر مبنای درخواست کاربر، پردازش روی چکیده‌هایی که در گام قبل ایجاد شده بود انجام می‌شود. به عنوان مثال، آگراول^{۱۳} و همکارانش، یک روش آنالین-آفلاین برای خوشه‌بندی داده‌های جریانی ارائه کرده‌اند که در بخش آنالین، خلاصه‌ای از اطلاعات آماری داده جریانی به صورت برخط گردآوری می‌شود. در طول بخش آفلاین، از این داده‌ها برای خوشه بندی سطح بالا استفاده می‌شود. رویکرد یادگیری دوگامی در شکل نشان داده شده است. این روش قابلیت پردازش داده های جریانی را با سرعت بیشتری داد منتها محدودیت آن این است که کاربر باید تا فراهم شدن نتیجه، منتظر بماند. شکل ۲-۳ - ب رویکرد یادگیری آنالین-آفلاین را نمایش می‌دهد.

^{۱۲}Two-Phase Learning

^{۱۳}Aggarwal

۴-۲ اعتبار سنجی

در روش‌های سنتی یادگیری ماشین با مقدار داده‌ی محدود، فرآیند اعتبارسنجی بر استفاده‌ی بیشینه از داده تمرکز داشت. جداسازی^{۱۴} اعتبار سنجی متقابل^{۱۵}، تک‌گذاری^{۱۶} روش‌های استاندارد اعتبار سنجی هستند. در روش جداسازی به شکل تصادفی مجموعه‌ی داده به دو زیر مجموعه، یکی برای آموزش و دیگری برای آزمایش تقسیم می‌شود. نسبت‌های رایج برای تقسیم داده به مجموعه آموزش و آزمایش نصف و یک‌سوم است. در روش اعتبار سنجی متقابل با k دسته، داده به k مجموعه‌ی مساوی و مستقل از نمونه‌ها تقسیم می‌شد و یکی از این زیر مجموعه‌ها برای آزمایش و باقی آن‌ها ($k-1$) برای آموزش ادغام می‌شد. در این روش فرآیند اعتبار سنجی k بار و هر بار با یک زیر مجموعه برای آزمایش انجام می‌شود. روش تک‌گذاری نیز یک نوع اعتبار سنجی متقابل است که در آن k با تعداد کل داده‌ها برابر است.

در محیط جریان‌های داده‌ای، از آنجایی که داده می‌تواند نامحدود باشد، اعتبار سنجی بر ارزیابی مدل در صحنه‌های مختلف متمرکز است. یک روش شناخته‌شده رسم منحنی یادگیری با ذخیره‌سازی کارایی مدل در طول زمان است. این منحنی نشان خواهد داد که چقدر مدل پس از مشاهده‌ی داده‌های بیشتر، بهتر شده است و مدل چگونه با رانش مفهوم سازگار می‌شود. یک الگوریتم برتری خود را به سایر روش‌ها زمانی نشان می‌دهد که منحنی یادگیری بیشتر مواقع از سایر الگوریتم‌ها بالاتر باشد.

روش جداسازی و روش پی‌درپی^{۱۷} دو روش پر کاربرد برای اعتبارسنجی جریان‌های داده‌ای هستند. در روش جداسازی، نمونه‌های داده به چانک‌های مختلفی تقسیم می‌شوند. از هر چانک داده ابتدا برای آزمایش و سپس برای بروزرسانی مدل استفاده می‌شود. روش جداسازی زمانی که رانش مفهوم رخ داده، ترجیح داده می‌شود چون این روش اجازه می‌دهد که مدل با آخرین تغییرات داده سازگار شود. روش پی‌درپی (یا آزمایش-سپس-آموزش^{۱۸} به شکل لایه لایه) یک روش دیگر برای ارزیابی جریان‌های داده‌ای است. هر نمونه داده ابتدا و پیش از این‌که برای آموزش به صورت افزایشی استفاده شود، برای آزمایش استفاده می‌شود. این روش می‌تواند یک حالت خاص برای روش جداسازی در نظر گرفته شود اگر اندازه چانک برابر با یک باشد. مزیت این روش این است که نیازی به تعریف از پیش اندازه چانک نیست ولی متأسفانه کارایی این الگوریتم در زمان محدود

^{۱۴} Hold-out^{۱۵} Cross-Validation^{۱۶} Leave-One-Out^{۱۷} Prequential^{۱۸} Test-Then-Train

مبهم است چراکه اشتباهات اولیه مدل به سرعت در طول زمان کاهش می یابد.

معیارهای ارزیابی

به طور عمومی معیارهای ارزیابی روش های سنتی یادگیری ماشین، می تواند برای ارزیابی یادگیرنده ها در جریان های داده نیز استفاده شود. برای رده بندی داده ها، صحت^{۱۹} و تابع ضرر^{۱ - ۰} دو معیار رایج هستند. برای ارزیابی مداوم کارایی رده بندی در داده های جریانی، گیم^{۲۰} و همکارانش یک روش ضرر پی درپی که با انواع مختلف پنجره کار می کند را ارایه کرده اند. برای از یاد بردن پی درپی خطا اثبات شده است که همگرایی خطای بیز در داده های ایستا، روش را برای تشخیص رانش کارا می سازد. این روش می تواند به سادگی روی سایر معیارهای کارایی نیز اعمال شود.

۲-۵ نرم افزارهای کاربردی

نرم افزارهای مختلف کاربردی و متن باز برای تحقیقات دانشگاهی در حوزه یادگیری و داده کاوی در محیط های پویای جریان های داده ای وجود دارد.

WEKA

شناخته شده ترین ابزار داده کاوری در محیط دانشگاهی است. این ابزار شامل یک مجموعه از الگوریتم های پردازی داده، رده بندی، رگرسیون، خوشه بندی، قواعد باهم آیی و بصری سازی داده است.

MOA

نرم افزار MOA یک فریمورک متن باز محبوب برای داده کاوی در داده های جریانی است که توسط دانشگاه وایکیتو نیوزلند^{۲۱} و بر مبنای چهارچوب WEKA توسعه داده شده است. در این نرم افزار که توسط زبان جاوا پیاده سازی شده، تعداد زیادی ابزار مناسب جهت پیاده سازی و تست الگوریتم های یادگیری و خوشه بندی و تشخیص رانش مفهوم وجود دارد. برای نمونه الگوریتم های درخت تصمیم خیلی سریع و مجمع رده بند ها

^{۱۹} Accuracy

^{۲۰} Game

^{۲۱} University of Waikato, New Zealand

که در ادامه توضیح داده می‌شوند در این چهارچوب موجود است. این ابزار تعدادی تولیدکننده^{۲۲} داده جریانی مانند مفهوم‌های SEA، ابرصفحه‌ی دوران‌کننده^{۲۳} و STAGGER را فراهم کرده است.

Rapid-Miner

یکی دیگر از ابزارهای متن‌باز برای داده‌کاوی است. RapidMiner بسیار قدرتمندتر از WEKA است و تمام الگوریتم‌های WEKA و سایر الگوریتم‌های پیشرفته را دارد. این ابزار خلاقانه‌تر است و این قابلیت را داد که پروسه داده‌کاوی را به شکل یک دنباله از عملگرها تعریف کرد. همچنین RapidMiner ابزارهای بیشتری را بصری‌سازی فراهم کرده است.

۲-۶ مخزن‌های داده

داده‌های واقعی بسیار کمی برای ارزیابی روش‌های یادگیری در داده‌های جریانی وجود دارد. یک دلیل این موضوع می‌تواند این باشد که محققان روش‌های سنتی داده‌کاوی و یادگیری ماشین معمولاً داده‌های خود را به قدری کوچک نگه می‌داشتند تا با روش‌های یادگیری دسته‌ای^{۲۴} سازگار باشد. یک دلیل دیگر می‌تواند مساله حریم شخصی در انتشار داده‌های که خیلی بزرگ هستند باشد. محققان معمولاً از داده‌های خصوصی برای اثبات سیستم‌هایشان استفاده می‌کنند که نمی‌توانند آن‌ها را منتشر کنند. برای غلبه بر این کمبود، بعضی از مجموعه داده‌های ترکیب شده با تعداد نامحدود داده ساخته شده‌اند. برای مثال تولید کننده درخت تصادفی، تولید کننده مفهوم SEA و ابرصفحه‌های چرخنده تولید شده‌اند. این تولیدکننده‌های داده در چهارچوب MOA پیاده‌سازی شده‌اند. بعضی از مخزن‌های داده با مجموعه‌های داده‌ی بزرگ نیز برای ارزیابی روش‌های یادگیری در داده‌های جریانی استفاده می‌شود:

^{۲۲}Generator

^{۲۳}Rotating Hyperplane

^{۲۴}Batch Learning

UCI Machine Learning Repository

یک مخزن آنلاین معتبر^{۲۵} و شناخته شده برای آزمایش و آنالیز الگوریتم‌های یادگیری ماشین است. سه مجموعه داده پر استفاده در چندین مقاله برای ارزیابی جریان‌ها Coverttype Forest و Poker-Hand و Elec- tricity هستند.

KDD Cup Center

رقابت‌های سالانه‌ی داده‌کاوی و کاوش دانش توسط^{۲۶} ACM Special Interest Group سازماندهی می‌شود. معمولاً داده‌هایی که برای این رقابت تولید می‌شود می‌تواند منبع خوبی برای ارزیابی الگوریتم‌های یادگیری ماشین در داده‌های جریانی باشد.

^{۲۵}<http://www.ics.uci.edu/mllearn/mlrepository.html>

^{۲۶}<http://www.kdd.org/>

فصل ۳

روش‌های یادگیری در داده‌های جریانی

رده‌بندی فرایند یافتن یک مدل عمومی از داده‌های گذشته است که به طوری که بتوان آن مدل را روی داده‌های جدید اعمال کرد. رده‌بندی از دو گام تشکیل شده است: گام یادگیری (آموزش) و گام تست. در بخش یادگیری سیستم تلاش می‌کند که یک مدل از مجموعه‌ی داده‌های نمونه‌ی مجموعه آموزشی یاد بگیرد و در گام تست این مدل برای برچسب‌زنی به داده‌هایی که برچسب زده نشده‌اند استفاده می‌شود. در ادبیات داده‌های جریانی، تعداد زیادی الگوریتم رده‌بندی مانند درخت‌های تصمیم، رده‌بند بیزین، ماشین‌های بردار پشتیبان، k نزدیک‌ترین همسایگی و روش‌هایی که مجمعی از رده‌بندها را می‌سازند وجود دارد. در این فصل به بررسی این روش‌ها می‌پردازیم و روش کار هر کدام را به طور مختصر توضیح می‌دهیم.

۳-۱ درخت هافدین

درخت هافدین یک رده‌بند بر مبنای درخت تصمیم برای داده‌های جریانی است. روش‌های سنتی درخت تصمیم که برای انتخاب خصیصه تقسیم نیاز به چندین بار پویش داده دارند که این موضوع در محیط داده‌های جریانی، عملاً نشدنی است. سایر روش‌های رده‌بندی داده‌های جریانی نیز معمولاً کاستی‌هایی مانند موارد زیر دارند:

- حساسیت زیاد به ترتیب نمونه‌ها
- کارایی کم. در بعضی موارد آن‌های از الگوریتم‌های یادگیری دسته‌ای کندترند.

درخت هافدین که یکی از روش‌های جدید یادگیری برای جریان‌هاست چالش‌های زیر را حل می‌کند:

- عدم قطعیت در زمان یادگیری. زمان یادگیری در درخت هافدین برای هر نمونه ثابت است و این بدان معنیست که درخت هافدین برای کاوش در داده‌های جریانی مناسب است.

- زمانی که تعداد نمونه کافی برای ساخت درخت پویش شود، نتیجه درخت در روش هافدین تقریباً مشابه (یا برابر) با درخت‌هایی است که با روش‌های مرسوم یادگیری دسته‌ای است.

برای تعامل با چالش پویش چندباره داده‌ها از حد هافدین برای انتخاب یک خصیصه تقسیم بهینه، پس پویش تعداد کافی نمونه استفاده می‌شود. فرض کنید N تعداد مشاهدات مستقل از یک متغیر تصادفی r باشد که این متغیر تصادفی در محدود R و با میانگین \bar{r} است. حد هافدین تضمین می‌کند که میانگین درست r حداقل از $\bar{r} - \epsilon$ با احتمال $1 - \delta$ بزرگ‌تر باشد. در این رابطه δ پارامتری است که کاربر انتخاب می‌کند.

$$P(E[r] \geq (\bar{r} - \epsilon)) \geq 1 - \delta, \epsilon = \sqrt{\frac{R^2 \ln(\frac{1}{\delta})}{2N}}$$

چیزی که حد هافدین را جالب توجه می‌کند، امکان گرفتن نتیجه‌های مشابه بدون در نظر گرفتن توزیع احتمالی است که مشاهدات را تولید می‌کند. به عنوان نمونه اگر اختلاف بهره اطلاعاتی بین دو خصیصه A و B (که می‌دانیم بهره اطلاعاتی خصیصه A بیشتر از خصیصه B است). و $e = 0.1$ باشد این معنی را می‌دهد که در آینده، حداقل تفاوت بین بهره اطلاعاتی A و B حداقل $0.2 = 0.1 - 0.3$ است. به عبارت دیگر با احتمال $1 - \delta$ ، هنگامی که تفاوت بهره اطلاعاتی مشاهده شده بزرگ‌تر از ϵ باشد یک خصیصه در مقایسه با دیگر خصیصه‌ها قالب است.

فرض کنیم $G(x_i)$ یک معیار اکتشافی برای انتخاب کردن خصیصه جداسازی باشد و پس از مشاهده N نمونه، x_a و x_b اولین و دومین خصیصه برتر برای خصیصه جداسازی باشند. با این مفروضات، در درخت هافدین، متغیر r از رابطه $r = \Delta G = G(X_a) - G(X_b)$ تعیین می‌شود. اگر $\bar{r} = \Delta \bar{G}$ باشد که در آن ϵ از رابطه بالا محاسبه شده است، می‌گوییم با اطمینان $1 - \delta$ این تفاوت بزرگ‌تر از $\bar{r} - \epsilon > 0$ است. بنابراین خصیصه x_a برای جداسازی انتخاب می‌شود.

نویسندگان الگوریتم درخت هافدین را توسط یک یادگیرنده با نام «درخت تصمیم خیلی سریع» (VFDT) پیاده‌سازی کرده‌اند. این پیاده‌سازی شامل بهبودهایی برای استفاده‌های خاص، مانند راهبرد محدود کردن

گره، راهنمایی سریع با استفاده از یادگیرنده‌های مبتنی بر RAM و قابلیت بازپوشش نمونه‌های قبلی زمانی که سرعت گذر داده کم است می‌باشد.

الگوریتم درخت هافدین، یک الگوریتم با دقت بالاست که می‌تواند بسیار خوب با مجموعه‌ی داده‌های بزرگ کار کند، ولی این الگوریتم نمی‌تواند با مساله رانش مفهوم در داده‌های جریانی تعامل کند. الگوریتم CVFDT یک تعمیم از الگوریتم درخت هافدین است برای تعامل با رانش مفهوم در داده‌های جریانی است. CVFDT، بعضی آماره‌های کارآمد را برای بررسی کردن اعتبار تصمیم‌های قبلی در کنار هر گره درخت نگهداری می‌کند. زمانی که یک داده وارد می‌شود، به طور مداوم این آماره‌های قرارگرفته در کنار گره‌های درخت بروزرسانی می‌شوند. با کمک پنجره‌های جابه‌جا شونده روی داده، این الگوریتم اثر داده‌هایی که از پنجره خارج هستند را در آماره‌های هر گره نادیده می‌گیرد. این پویای گره‌های درخت باعث تشخیص رانش مفهوم می‌شوند. اگر رانش مفهوم ظاهر شده باشد، CVFDT به صورت موازی شاخه‌های جایگزین را با انتخاب بهترین خصیصه‌های جدید و حذف شاخه‌های قدیمی، اگر دقتشان کم باشد، گسترش می‌دهد.

۲-۳ رده‌بند بیزین

سیدل^۱ و همکارانش یک روش جالب برپایه نمایه‌سازی^۲ رده‌بند ارایه داده‌اند که درخت بیز نامیده می‌شود. درخت بیز یک درخت مخلوط-گوسین سلسله مراتبی^۳ جهت نمایان کردن تمام مجموعه‌ی داده استفاده می‌کند. هر گره از درخت شامل آماره‌های از نمونه‌های داده مثل مستطیل محدودکننده کمینه^۴، تعداد نمونه‌های داده، جمع خطی و مجموع مربعات تمام داده است. برای حل مساله‌ی رده‌بندی چند کلاسه، یک درخت بیز برای هر کلاس ساخته می‌شود. برای هر داده‌ی ست مانند x ، الگوریتم سعی می‌کند که نزدیک‌ترین مجموعه گره‌ها را که مرز نامیده می‌شوند پیدا کند. احتمال تعلق x به کلاس c_i توسط رابطه زیر محاسبه می‌شود:

$$P(c_i|x) = \left(\sum_{e_s \in E_i} \frac{n_{e_s}}{n} g(x, \mu_{e_s}, \sigma_{e_s}) \right) \star P(c_i)/P(x)$$

که در این رابطه e_s یک گره درخت در مجموعه مرز E_i است. n_{e_s} و μ_{e_s} و σ_{e_s} به ترتیب، تعداد

^۱ Seidl

^۲ Index-Based

^۳ Hierarchical Gaussian-mixture Tree

^۴ Minimum Bounding Rectangle

نمونه‌های داده، مرکز و انحراف گره e_s هستند. برای مشخص کردن برچسب نمونه‌ی x ، احتمال برای تمام کلاس‌ها حساب می‌شود و کلاسی که بیشترین احتمال را داشته‌باشد به عنوان کلاس نمونه x اعلام می‌شود. درخت بیزیک رده‌بند که پس از مدت کمی از شروع، می‌تواند نتیجه و تصمیم داشته‌باشد و پس از آن دقت‌ها را با انتخاب مرزهای دقیق‌تر افزایش دهد.

۳-۳ شبکه عصبی

لیت^۵ و همکارانش روش شبکه‌های عصبی داده‌دانه تغییر پذیر^۶ (eGNN) را جهت رده‌بندی داده‌های جریانی ارایه داده‌اند. دو گام در eGNN وجود دارد. در گام اول، eGNN از نرون‌های T-S برای ساختن ریزداده‌های اطلاعات داده‌هایی که در حال آمدن هستند استفاده می‌کند. در گام دوم، شبکه عصبی روی این ریزداده‌های اطلاعاتی به جای داده‌ی اصلی ساخته می‌شود. یک ریزدانه مرتبط با برچسب یک کلاس به شکل یک تابع عضویت سه‌گوش^۷ تعریف می‌شود که بعداً جهت تطبیق با داده‌های جدید تغییر می‌کند. وزن‌های با یک ثابت زوال^۸ کاهش پیدا می‌کنند. این پردازش کمک می‌کند که درجه اهمیت ریزدانه‌های منقضی شده کاهش پیدا کند. به شکل پایه، eGNN از نمونه‌های کلاس در فرم ریزدانه‌های اطلاعاتی جهت انجام رده‌بندی استفاده می‌کند. زمانی که داده‌های تست می‌آیند یک نرون ماکسیسم گیری بهترین پیش‌بینی ریزدانه‌ی را انتخاب می‌کند و برچسب آن را به نمونه جهت پیش‌بینی کلاسش اختصاص می‌دهد. eGNN می‌تواند در مساله رده‌بندی در محیط‌های که دائماً تغییر می‌کند استفاده شود اما نیاز به زمان طولای آموزش، هنوز یک محدودیت برای شبکه‌های eGNN جهت کار با داده‌های بسیار بزرگ است. البته، در قسمت آزمایش‌های مقاله اصلی، مجموعه‌های داده‌ی کوچک برای ارزیابی این روش استفاده شده است. سپس نویسنده، این روش را جخت کارایی بهتر و کار به شکل نیمه نظارتی گسترش داده‌است تا با مجموعه‌ی داده‌هایی که کاملاً برچسب گذاری نشده‌اند هم کار کند.

^۵Leite

^۶evolving granular neural network

^۷Triangular membership function

^۸decay

۳-۴ ماشین‌های بردار پشتیبان

ماشین‌های بردار پشتیبان^۹ کارایی خود را در بسیاری از مساله‌های یادگیری ماشین با داده‌های ایستا نشان داده‌اند. البته استفاده از ماشین‌های بردار پشتیبان برای کاربردهایی با مقیاس بزرگ بسیار پرهزینه است. اگر N تعداد نمونه‌های داده باشد، این الگوریتم از پیچیدگی زمانی $O(N^3)$ و از پیچیدگی حافظه‌ای $O(N^2)$ است. برای کار کردن با داده‌های بسیار بزرگ، تی‌سنگ^{۱۰} و همکارانش، الگوریتم ماشین بردار هسته^{۱۱} (CVM) را ارائه داده‌اند تا پیچیدگی الگوریتم ماشین بردار پشتیبان را کاهش دهند. این الگوریتم از حداقل گوی نزدیک (MEB) برای این کار استفاده می‌کند. یک MEB ابر کره^{۱۲} (کره با بیش از سه بعد) است که نماینده یک مجموعه از نمونه‌های داده کنار آن است. این الگوریتم ابتدا MEB های نماینده را که تخمین خوبی از داده اصلی باشند، پیدا می‌کند. سپس مساله بهینه‌سازی پیدا کردن محدوده‌ی بیشینه بر روی این مجموعه‌های MEB اعمال می‌شود.

ری^{۱۳} و همکارانش الگوریتم دیگری با نام StreamSVM که یک بهبود از الگوریتم CVM است را برای کار با داده‌های جریانی ارائه کرده‌اند. در این الگوریتم، یک MEB شعاع انعطاف پذیری است، هر زمان نمونه‌های جدید آموزش اضافه می‌شوند، این شعاع افزایش می‌یابد. این الگوریتم زمانی که تخمین‌ها درست باشند با بهینه‌ترین روش‌ها قابل رقابت است، اما StreamSVM هنوز قابلیت تشخیص رانش مفهوم در داده‌های جریانی را ندارد.

۳-۵ رده‌بند نزدیک‌ترین همسایگی

در مورد خوشه بندی جریان‌های داده‌ای (که مورد بحث این گزارش نیست)، یک روش خوشه‌بندی با نام CluStream وجود دارد. این روش از تعریفی تحت عنوان ریز-خوشه^{۱۴} برای خوشه‌بندی داده‌ها استفاده می‌کند. یک ریز-خوشه برای تعدادی داده، شامل آماره‌هایی نظیر مجموع مربعات نقاط، مجموع نقاط، مجموع مربعات زمان رخداد نقاط، مجموع خطی زمان رخداد و تعداد نمونه‌هاست. این روش خوشه‌بندی

^۹Support Vector Machine

^{۱۰}Tsang

^{۱۱}Core Vector Machine

^{۱۲}Hypersphere

^{۱۳}Rai

^{۱۴}Micro-Cluster

به جای ذخیره کل داده‌های خوشه، در گام آنلاین ریز-خوشه‌های آن‌ها که حجم کمتری را اشغال می‌کند و پردازش آن آسان‌تر است را ساخته و ذخیره می‌کند و در گام آفلاین به جای خوشه‌بندی کل داده‌ها، این ریز-خوشه‌ها را خوشه‌بندی می‌کند. روش On-Demand-Stream یک گسترش از روش CluStream است که همانند این روش، از رویکرد آنلاین-آفلاین و پنجره یک‌بر استفاده می‌کند. تفاوت ریز-خوشه در این روش با ریز-خوشه در روش CluStream این است که برچسب کلاس نیز به ریز-خوشه اضافه می‌شود و ریز-خوشه‌ها فقط شامل نمونه‌هایی از یک کلاس مشخص و یکسان هستند. در رده‌بندی آفلاین، سعی می‌شود که بهترین پنجره از نمونه‌های داده که افق زمان^{۱۵} نامیده می‌شود انتخاب شود و ریز-خوشه‌ها از این افق زمانی استخراج می‌شوند. روش On-Demand-Stream از یک رده‌بندی ۱NN برای نسبت‌دادن داده‌ی آزمایش به نزدیک‌ترین ریز-خوشه استفاده می‌کند.

با الهام‌گرفتن از ایده M-tree^{۱۶} ژنگ و همکارانش یک روش ساختار Lazy-tree برای نمایه‌سازی ریز-خوشه‌ها پیشنهاد داده‌اند. این کمک می‌کند که پیچیدگی زمانی روش نزدیک‌ترین همسایگی اگر فرض کنیم N تعداد ریز-خوشه‌ها در Lazy-tree باشد، از $O(N)$ به $O(\log(N))$ کاهش یابد. درخت شامل سه عملگر اصلی است: جستجو، حذف و افزودن. اضافه‌کردن و حذف‌کردن برای افزودن گره‌های جدید و حذف‌کردن گره‌های منقضی شده به کار می‌رود و تضمین می‌کنند که درخت متعادل بماند. عملگر جستجو برای رده‌بندی داده‌های تست استفاده می‌شود.

۳-۶ مجمع رده‌بندها

روش‌های Bagging & Boosting برتری خود را از طریق آزمایش‌ها روی مجموعه‌ی داده‌های سنتی اثبات کرده است، نظر به این موضوع، بسیاری از محققین تلاش کرده‌اند که این روش‌ها را برای کار روی داده‌های جریانی سازگار کنند. اوزا^{۱۶} و همکارانش روشی تحت عنوان Bagging & Boosting آنلاین ارائه کرده‌اند که یکی از اولین کارها برای سازگار کردن این روش بوده است. با دید آماری، هر نمونه از داده‌های آموزشی k بار در مجموعه‌ی آموزشی اعضای رده‌بند با احتمال زیر مشاهده می‌شود:

^{۱۵}Time Horizon

^{۱۶}Oza

$$P(k) = \binom{N}{k} \left(\frac{1}{N}\right)^k \left(1 - \frac{1}{N}\right)^{N-k}$$

که در رابطه k اندازه مجموعه‌ی آموزشی و N اندازه مجموعه‌ی داده‌هاست. در داده‌های جریانی می‌توان فرض کرد که تعداد نمونه‌های داده نامحدود است، لذا $N \rightarrow \infty$. بنابراین احتمال $P(k)$ به توزیع احتمال پواسون میل می‌کند که رابطه آن مشابه زیر است:

$$Poisson(1) = \exp(1)/k!$$

با این مشاهده، اوزا و همکاران روش Online Bagging را جهت نسبت دادن هر شی از داده به یک وزن متناسب با توزیع پواسون را ارایه کردند. در Online Boosting وزن‌های داده‌هایی که در حال آمدن هستند و اعضای رده‌بندها بر اساس نسبت خطای اعضای رده‌بند به خطای کنونی تنظیم می‌شود.

۳-۶-۱ مجمع وزنی رده‌بندها

نظر به متقاضی شدن داده‌های قدیمی، باید به توزیع داده به جای زمان رسیدن داده اعتماد کرد. بر این اساس ونگ^{۱۷} و همکارانش یک روش دقت-وزن برای مجمع رده‌بندها^{۱۸} (AWE) ارایه کردند تا بتوان رانش مفهوم را در داده‌های جریانی کاوش کرد. الگوریتم با تعداد k رده‌بند که می‌توانند RIPPER یا $C4.5$ یا بیزین ساده باشند ایجاد می‌شود و با همان تعداد نیز نگهداری می‌شود. با استفاده از یک تکه از نمونه‌های داده‌ای که در حال آمدن هستند یک رده‌بند جدید آموزش داده می‌شود. سپس مجمع با انتخاب k امین رده‌بند های با دقت سازمان‌دهی می‌شود و وزن هر رده‌بند متناسب با دقت آن تعیین می‌شود. این روش بهتر از یک رده‌بند تنهای داده‌های جریانی مانند VFDT و CVFDT کار می‌کند ولی دقت آن، خیلی به اندازه تکه‌ی انتخاب شده و تعداد رده‌بندها (k) وابسته است.

در کاربردهای واقعی، در داده‌های جریانی ممکن است خطاهایی نظیر این که داده‌ها اشتباه برجسب خورده یا مقدار اشتباه داشته باشند دیده شود. ژنگ^{۱۹} و همکارانش الگوریتم مجمع متراکم را برای تعامل با داده‌های

^{۱۷}Wang

^{۱۸}Accuracy-Weighted Ensemble

^{۱۹}Zhang

دارای خطا ارایه کردند. این رویکرد ترکیبی از چهارچوب‌های افقی و عمودی مجمع‌هاست. چهارچوب افقی رده‌بندهای مخلفی بر روی هر چانک داده می‌سازد در حالی که چهارچوب عمودی رده‌بندهای مخلفی چانک داده‌ی بروز با الگوریتم‌های مخلف می‌سازد. چهارچوب افقی در برابر نویز مقاوم است و می‌تواند از اطلاعات گذشته استفاده کند ولی وقتی که رانش مفهومی رخ دهد که مفهوم در داده‌های جریانی به طور عمده تغییر کند مناسب نیست. از طرف دیگر، چهارچوب عمودی می‌تواند وقتی که رانش مفهوم رخ دهد نتیجه خوبی بدهد ولی در برابر نویز حساس است. ساختن رده‌بندها روی چانک‌های مخلف داده و با استفاده از الگوریتم‌های مخلف یادگیری، می‌تواند راه خوبی برای جمع‌آوری یک مجمع از رده‌بندها باشد. این رده‌بندها در یک ماتریس با نام ماتریس رده‌بند^{۲۰} جمع‌آوری می‌شوند. روش میانگین‌گیری وزنی روی این ماتریس، برای پیش‌بینی برچسب داده‌های آزمایش استفاده می‌شود. نویسنده به شکل تیوری اثبات کرده است که جمع‌آوری یک مجمع، به طور میانگین مجموع مربعات خطای کمتر یا حداکثر برابر با هر کدام از چهارچوب‌های افقی یا عمودی به تنهایی دارد.

نگوین^{۲۱} و همکارانش روی مساله یادگیری داده‌های جریانی با ابعاد بالا زمانی که فقط یک زیر مجموعه از ویژگی‌ها برای فرایند یادگیری با اهمیت هستند. در این موضوع، مفهوم ویژگی مرتبط یک مفهوم موقت و محدود به یک دوره زمانی خاص است. ویژگی‌های آموزنده^{۲۲}، ممکن است پس از مدتی نامربوط شوند در حالی که ویژگی‌های به‌درد نخور به ویژگی‌های مهم تبدیل شوند. نویسنده با ارایه مفهوم رانش ویژگی، به طور مثال تغییر در مجموعه‌ی ویژگی‌های باارزش و ادغام این مفهوم با وزن‌دهی در مجمع رده‌بندها، سعی کرده است که این مساله را حل کند. روش انتخاب ویژگی‌های چند متغیره، سازگار با تکنیک پنجره کشویی است و می‌توان به کمک آن رانش‌های ویژگی را تشخیص داد. وقتی یک رانش تدریجی رخ می‌دهد، رده‌بندهای مجمع بروزرسانی می‌شوند و وزن آن‌ها با توجه به نسبت خطایشان تنظیم می‌شود. زمانی که یک رانش ویژگی رخ می‌دهد، مجمع یادگیرنده‌های که قبلاً استفاده می‌شده را با رده‌بندهایی که بروز رسانی شده‌اند جایگزین می‌کند. در بروزرسانی یادگیری با یک مجموعه‌ی جدید از ویژگی‌های با ارزش انجام می‌شود. آزمایش‌ها نشان می‌دهد که برای داده‌های جریانی با ابعاد بالا، این روش موثر و کارآمد است.

^{۲۰} Classifier Matrix^{۲۱} NGUyen^{۲۲} Informative

۳-۶-۲ درخت‌های سازگار تصمیم مبتنی بر رویکرد یکی در برابر بقیه

درخت سازگار تصمیم مبتنی بر رویکرد یکی در برابر بقیه^{۲۳} یک روش مجمع جدید برای داده‌های جریانی است. در این روش، مجمع k رده‌بند باینری CVFDT را یاد می‌گیرد. هر رده‌بند آموزش داده شده است که تشخیص دهد یک نمونه به یک کلاس خاص متعلق است یا به باقی کلاس‌ها. برای رده‌بندی یک داده‌ی جدید، همه رده‌بندها اجرا می‌شوند و رده‌بندی که مطمئن‌ترین نتیجه را داد، به عنوان برچسب نمونه‌ی جدید انتخاب می‌شود. اطمینان رده‌بندی متناسب با کلاس غالب در برگ‌ها زمانی که نمونه تست وارد می‌شود تنظیم می‌شود. برای دستیابی به بیش‌ترین دقت‌ها، OVA تلاش می‌کند یک مجمعی از رده‌بندهای CVFDT بسازد که کمترین همبستگی خطا و بیشترین تنوع را داشته باشند. این روش به سهولت و سریع با رانش مفهوم سازگار می‌شود و فقط کفایت دو بخش رده‌بندها که مرتبط با کلاسی که تغییر کرده است، بروز رسانی شوند. به علاوه این روش به خوبی برای داده‌های جریانی نامتوازن کار می‌کند.

۳-۶-۳ مجمع متا-دانش

بیشترین حد ممکن پیچیدگی برای رده‌بندها یادگیرنده، محدود به محدودیت‌های زمانی جریان‌های داده‌ای در کاربردهای دنیای واقعی است. ژنگ^{۲۴} و همکارانش، یک روش مجمع متا-دانش^{۲۵} که بهترین و مناسب‌ترین رده‌بند ها را برای رده‌بندی داده‌های تست انتخاب می‌کند را ارائه دادند. یک مجمع درخت برای سازماندهی رده‌های پایه ساخته شده است و هر کدام یک وزن و یک محدوده از کل داده‌ها اختصاص داده می‌شود. مشابه RTree، درخت مجمع‌ها سه عملگر پایه، شامل جستجو، اضافه کردن و حذف کردن دارد. ارتفاع این درخت متوازن است و پیچیدگی زمانی لگاریتمی را برای پیش‌بینی تضمین می‌کند. از عملگر افزودن برای ادغام یک رده‌بند جدید به مجمع استفاده می‌شود. زمانی که تعداد رده‌بندها در یک گره، از یک حد از پیش تعیین شده تجاوز کند، گره با این ایده که محدوده تحت پوشش هر گره کمینه شود، به دو گره تقسیم می‌شود. زمانی که ظرفیت E-tree پر شده باشد، عملگر حذف رده‌بندهای منقضی شده را حذف می‌کند و درخت ممکن است که دوباره سازماندهی شود تا بالانس بودن تضمین شود. عملگر جستجو نیز برای رده‌بندی یک نمونه مانند x استفاده می‌شود. رده‌بندهای فضای نزدیک شامل x فراخوانی می‌شوند و یک روش رای‌گیری وزنی برای

^{۲۳} Adapted One-vs-All Decision Trees (OVA)^{۲۴} Zhang^{۲۵} Meta-knowledge Ensemble

تصمیم‌گیری برچسب نمونه‌ی x اجرا می‌شود.

فصل ۴

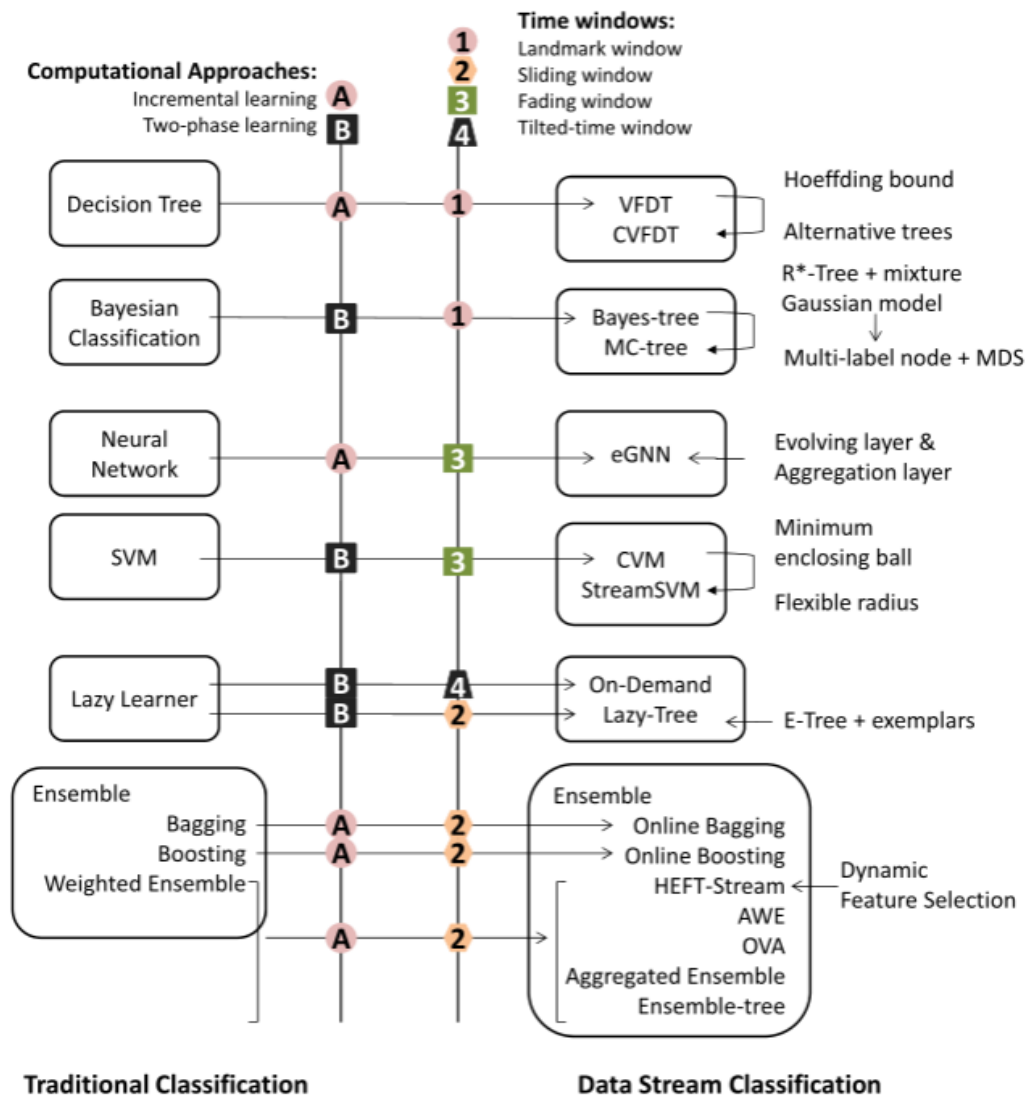
جمع‌بندی و نتیجه‌گیری

در فصل‌های گذشته به بررسی روش‌های یادگیری ماشین در فضای جریان‌های داده‌ای پرداختیم. در این فصل روش‌های معرفی شده را جمع‌بندی خواهیم کرد و مزایا و معایب هر روش را بیان می‌کنیم. به علاوه نسبت این الگوریتم‌ها را با الگوریتم‌های سنتی یادگیری ماشین بیان خواهیم کرد. سپس با توجه به ذات داده‌های متنی، مشخص خواهیم کرد که کدام یک از این روش‌ها می‌توانند در مواجهه با داده‌های متنی جریانی مورد استفاده قرار گیرند و در نهایت کارهای آینده را بیان می‌کنیم.

۴-۱ مروری بر روش‌های یادگیری

پس از معرفی تعدادی از رده‌بندی‌هایی که برای داده‌های جریانی وجود دارند، به جمع‌بندی این موضوع و تمام مسایل مرتبط خواهیم پرداخت.

تصویر ۴-۱ ارتباط بین روش‌های سنی رده‌بندی و رده‌بندی در داده‌های جریانی را نشان می‌دهد. رده‌بندی سنتی در سمت چپ نوشته شده‌اند و رده‌بندی‌های داده‌های جریانی در سمت راست. برای دسته‌بندی رده‌بندی‌های داده‌های جریانی به دو الگو، برای رویکردهای محساباتی و پنجره‌های زمانی نیز در وسط تصویر است. می‌توان دید که رده‌بندی‌های داده‌های جریانی از رده‌بندی‌های سنتی برگرفته شده‌اند، با این تفاوت که از رویکردهای محساباتی مختلف و پنجره‌های زمانی متفاوتی استفاده می‌کنند. به عنوان نمونه VFDT یک گسترش از درخت‌های تصمیم برای داده‌های جریانی است. زمانی که به حد کافی داده موجود باشد، VFDT از حد



شکل ۴-۱: مقایسه و نمایش ارتباط روش‌های سنتی یادگیری ماشین با روش‌های یادگیری در داده‌های جریانی

هافدین برای ساختن گره‌های درخت استفاده می‌کند. در واقع این دنباله‌ای از رویکرد یادگیری افزایشی و پنجره نقطه عطفی است CVFDT. یک نسخه بهبود یافته از VFDT است که می‌تواند به وسیله‌ی ساختن درخت‌های جایگزین با رانش مفهوم سازگار شود. درخت بیز یک نسخه تغییر یافته از رده‌بندهای بیزین با دو فاز مختلف یادگیری و پنجره نقطه عطفی است. MCTree نیز بهبودی از درخت بیز با استفاده از گره‌های چند برچسبه است. eGNN یک شبکه عصبی است که برای کار با داده‌های جریانی طراحی شده است. CVM نیز نشأت گرفته از رده‌بند SVM اس که از رویکرد دوفازی یادگیری و پنجره محو شونده استفاده می‌کند. StreamSVM نیز یک نسخه توسعه یافته CVM است که با ساختن حداقل گوی نزدیک به شکل پویا عمل می‌کند.

الگوریتم On-Demand، یک نسخه بهبود یافته از رده‌بندهای k-NN است که با استفاده از رویکرد یادگیری دوگامی و پنجره یک بر کار می‌کند. Lazy-Tree نیز یک نسخه بهبود یافته از رده‌بند k-NN است که با استفاده از رویکرد یادگیری دوگامی و و پنجره کشویی کار می‌کند. تعداد زیادی مجمع رده‌بندها برای کار با داده‌های جریانی طراحی شده است. Bagging و Boosting آنلاین، نسخه‌های از روش‌های سنتی Bagging و Boosting با استفاده از رویکرد یادگیری افزایشی و پنجره کشویی است.

تعداد زیادی مجمع وزنی از رده‌بندها وجود دارد که از استراتژی‌های مختلف وزن‌دهی استفاده می‌کنند. برای نمونه مجمع‌های جمع‌آوری شده که از روش میانگین وزنی (رای‌گیری) استفاده می‌کنند مانند HEFT-Stream، AWE، OVA، Ensemble-tree که در این روش‌ها وزن متناسب با دقت رده‌بند تنظیم می‌شود. این روش‌های دنباله‌ای از رویکرد یادگیری افزایشی هستند و می‌توان فرض کرد که از پنجره کشویی استفاده می‌کنند و به طور مرسوم، اعضای رده‌بندی که دقت کمی دارند را از مجمع حذف می‌کنند.

به طور خلاصه، در جدول ۴-۱ ظرفیت‌های رده‌بند های مختلف روی داده‌های جریانی، به شکل مقایسه‌ای آورده شده است. همانطور که مشهود است بسیاری از رده‌بندهای مطرح شده، عموماً نمی‌توانند در آن واحد در تمام محدودیت‌های کار با داده‌های جریانی بگنجند. در این جدول در دو سطح تعامل با رانش وجود دارد: این که بتوانند با رانش مفهوم سازگار شوند و این که رانش مفهوم را رده‌بندی کنند. به علاوه مشخص شده است که کدام یک از رده‌بندها می‌توانند با داده‌های با ابعاد بالا در جریان‌های داده‌ای کار می‌کنند.

همانطور که دور از انتظار نیست، تمام روش‌های رده‌بندی داده‌های جریانی، یکبار داده را مشاهده می‌کنند و با محدودیت حافظه مشکل چندانی ندارند. eGNN و CVM و StreamSVM نمی‌توانند بلادرنگ پاسخگو باشند و برای دادن نتیجه به زمان جهت پردازش‌هایشان احتیاج دارند. HEFT-Stream می‌تواند

جدول ۴-۱: مقایسه ظرفیت‌های الگوریتم‌های مختلف یادگیری در داده‌های جریانی

Algorithm	Bounded memory	Single-pass	Real-time response	Concept-drift adaptation	Concept-drift classification	High-dimensional data
Aggregated ensemble [106]	✓	✓	✓	✓		
AWE [98]	✓	✓	✓	✓		✓
Bayes tree [84]	✓	✓	✓	✓		
CVFDT [53]	✓	✓	✓	✓		✓
CVM [94]	✓	✓		✓		
eGNN [68]	✓	✓		✓		
Ensemble-tree [105]	✓	✓	✓	✓		✓
HEFT-stream [74]	✓	✓	✓	✓	✓	✓
Lazy-tree [104]	✓	✓	✓	✓		
MC-tree [61]	✓	✓	✓	✓		
On-Demand [5]	✓	✓	✓	✓		
Online bagging & boosting [76]	✓	✓	✓			
OVA [82]	✓	✓	✓	✓		✓
StreamSVM [79]	✓	✓		✓		
VFDT [35]	✓	✓	✓			✓

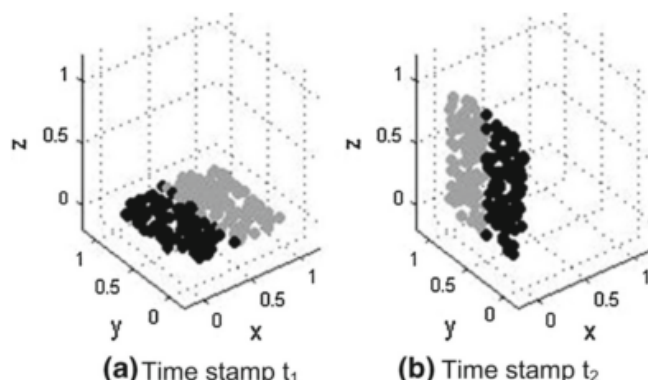
جدول ۴-۲: مزایا و محدودیت‌های روش‌های رده‌بندی برای داده‌های جریانی

روش	مزیت	محدودیت
درخت تصمیم	فهم آسان مقاوم در برابر خطا هزینه محاسباتی کم حتی با داده آموزش زیاد توانایی حذف ویژگی‌های تکراری	بیش برآزش هنگامی که عمق درخت‌ها زیاد باشد
بیزین شبکه عصبی	ساده، کارا، بهینه و مقاوم در برابر خطا به راحتی عمومی می‌شود می‌تواند مسایل پویا و غیر خطی را حل کند	پیش فرض استقلال که در دنیای واقعی معمولاً نقض می‌شود عدم توانایی تفسیر مدل یادگرفته شده پیچیدگی بالا
بودار پشتیبان	ارایه یک راه حل یگانه توانایی حل مسایل غیر خطی با گرفتن کرنل‌های متفاوت وقتی داده‌ها به یک سمت متمایل هستند هم خوب کار می‌کند.	به انتخاب کرنل وابسته است. پیچیدگی بالاست برای مسایل چند کلاسه طراحی سخت است
یادگیرنده Lazy نزدیک ترین همسایگی	پایاده سازی راحت	فضای ذخیره سازی زیاد پیچیدگی بالا
مجمع	صحت بالا پایاده سازی ساده	عملکرد بسیار بد برای داده‌های با ابعاد بالا بر پایه اکتشافات و نداشتن تئوری دقیق

دو نوع رانش مفهوم (هم رانش تدریجی و هم رانش ویژگی) را مشخص کند و به خوبی با آن‌ها سازگار شود. اگر درخت تصمیم به عنوان رده‌بند پایه برای روش‌های VFDT و CVFDT و AWE و OVA و درخت مجمع انتخاب شود، این روش‌ها می‌توانند با داده‌هایی با ابعاد بالا نیز کار کنند کارکنند.

همانطور که دور از انتظار نیست، تمام روش‌های رده‌بندی داده‌های جریانی، یکبار داده را مشاهده می‌کنند و با محدودیت حافظه مشکل چندانی ندارند. eGNN و CVM و StreamSVM نمی‌توانند بلادرنگ پاسخگو باشند و برای دادن نتیجه به زمان جهت پردازش‌هایشان احتیاج دارند. HEFT-Stream می‌تواند دو نوع رانش مفهوم (هم رانش تدریجی و هم رانش ویژگی) را مشخص کند و به خوبی با آن‌ها سازگار شود. اگر درخت تصمیم به عنوان رده‌بند پایه برای روش‌های VFDT و CVFDT و AWE و OVA و درخت مجمع انتخاب شود، این روش‌ها می‌توانند با داده‌هایی با ابعاد بالا نیز کار کنند کارکنند.

هنگامی که یک رده‌بند از رویکر محاسباتی و پنجره زمانی استفاده می‌کند بعضی مزایا و معایب که در فصل‌های قبل نیز در مورد آن‌ها بحث شد، خودشان را نشان می‌دهد. به هر روی روش‌های سنتی و رده‌بند‌های داده‌های جریانی هر دو مزایا و محدودیت‌هایی دارند که در جدول ۴-۲ آمده است. به عنوان مثال درخت تصمیم به راحتی قابل فهم است، در برابر خطا مقاوم است، کاراست و می‌تواند خصیصه‌های تکراری را حذف



شکل ۴-۲: یک مثال که نشان می‌دهد چگونه ذات ویژگی‌های بااهمیت طی زمان می‌تواند تغییر کند.

کند. اما اگر عمق درخت‌ها زیاد باشد دچار مشکل بیش‌برازش^۱ می‌شود. یادگیرنده Lazy می‌تواند به سادگی پیاده‌سازی شود اما این یادگیرنده حافظه زیادی مصرف می‌کند و در برابر داده‌هایی با ابعاد بالا مشکل دارد. مجمع‌ها صحت بالایی دارند و به سادگی پیاده‌سازی می‌شوند اما بیشترشان بر پایه اکتشافات هستند و پایه تئوری قوی ندارند.

۴-۲ کارهای آینده در جریان‌های متنی

از آنجایی که بیشتر تحقیقات در داده‌های جریانی مربوط به دهه‌ی اخیر بوده است، هنوز مسایل زیادی برای تحقیقات وجود دارد. با توجه موضوع این گزارش، در این بخش، تحقیقات آینده‌ای که مربوط به کاوش و یادگیری درباره داده‌های جریانی متنی است را مطرح می‌کنیم.

۴-۲-۱ انتخاب پویای ویژگی

در همه‌ی داده‌های با ابعاد بالا، که داده‌های متنی هم جز آن‌هاست تمام ویژگی‌ها (خصوصیه‌ها) در فرایند با اهمیت هستند. در واقع سه نوع مختلف ویژگی وجود داد: (یک) ویژگی‌های غیر مرتبط، (دو) ویژگی‌های مرتبط اما تکراری و (سه) ویژگی‌های مرتبط و غیر تکراری. وظیفه‌ی اصلی موضوع انتخاب ویژگی، استخراج مجموعه‌ی ویژگی‌های مرتبط و غیر تکراری از ویژگی‌هاست که فرایند یادگیری را با معنی‌تر و سریع‌تر کند. در

^۱Overfitting

ادبیات موضوع، روش‌های انتخاب ویژگی به سه دسته تقسیم می‌شوند: فیلتر، پوشش^۲ و مدل‌های نهفته^۳. مدل فیلتر به عنوان یک معیار مستقل برای ارزیابی مجموعه‌ی ویژگی‌ها به کار برده می‌شود، بنابراین این روش فقط به مشخصه‌های عمومی داده اتکا می‌کند. مدل پوششی، به همراه الگوریتم‌های یادگیری اجرا می‌شود از کارایی الگوریتم‌های یادگیری برای ارزیابی ویژگی‌ها استفاده می‌کند. مدل‌های ترکیبی از مزیت دو مدل گفته شده استفاده می‌کند.

اهمیت ویژگی‌ها در داده‌های جریانی تغییر می‌کند و محدود بودن به یک بازه‌ی زمان مشخص است. ویژگی‌های که پیش از این با ارزش فرض می‌شد ممکن است نامربوط شوند و برعکس ممکن است ویژگی‌های استفاده نشده در آینده با اهمیت شوند. بنابراین استفاده از روش‌های پویای انتخاب ویژگی برای رصد کردن تغییرات ویژگی‌ها ضروری است. شکل ۴-۲ ذات پویای ویژگی‌های کلیدی را نشان می‌دهد. فرض کنید داده‌های ما سه ویژگی و دو کلاس داشته باشند: نقطه‌های سیاه و قهوه‌ای نمایانگر کلاس‌های مثبت و منفی هستند. در زمان t_1 ، ویژگی‌های با ارزش x و y هستند چون داده روی صفحه xy جا گرفته است. پس از این که توزیع داده فرق می‌کند، ویژگی‌های با اهمیت y و z می‌شوند.

پس از بررسی تعداد زیادی از الگوریتم‌های یادگیری، مشاهده شد که تنها اندکی از آن‌ها می‌توانند با داده‌هایی با ابعاد بالا کار کنند و محدودیت‌های زیادی دارند. بعضی از رده‌بندی‌هایی که با داده‌های با ابعاد بالا کار می‌کنند. این الگوریتم‌ها معمولاً از درخت تصمیم به عنوان رده‌بند پایه استفاده می‌کنند و توانایی حذف ویژگی‌های تکراری را ندارند. HEFT-Stream تنها مجمع رده‌بندی است که هم ویژگی‌های نامرتب و هم ویژگی‌های تکراری را حذف می‌کند و با هر نوع رده‌بندی کار می‌کند. HEFT-Stream از مدل فیلتر استفاده می‌کند و مستقل از نوع اعضای رده‌بند است. به هر حال مساله انتخاب پویای ویژگی یک مساله باز است که نیاز به تحقیقات بیشتری دارد.

۴-۲-۲ کاوش در جریان‌های متنی

در سال‌های اخیر تعداد زیادی از نرم‌افزارهای مبتنی بر وب، حجم زیادی از جریان‌های متنی را تولید می‌کنند. به عنوان مثال، اعضای شبکه‌های اجتماعی به طور مداوم با پیام‌های متنی با یکدیگر تعامل می‌کنند. بسیاری از پرتال‌ها اخبار را بر اساس علاقه‌مندی خوانندگان، بلادرنگ به آن‌ها نشان می‌دهند و خزنده‌های وب میلیون‌ها

^۲ Wrapper

^۳ Embedded models

صفحه را برای نمایه سازی ذخیره می‌کنند. کاوش در جریان‌های متنی، که با سایر وظیفه‌هایی مانند فیلترکردن هرزنامه‌ها مرتبط است.

به طور عمومی، روش‌های داده‌های جریانی با ابعاد بالا می‌توانند برای داده‌های متنی استفاده شوند، البته این موضوع باید بعد از جمع‌آوری و عملیات پیش‌پردازش شامل حذف کلمات ایست، ریشه‌یابی، نگاشت به نمایش‌هایی از قبیل کیسه‌ای از کلمات^۴، TF-IDF و جداسازی عبارت‌ها باشد. اما داده‌های متنی پیچیده‌تر از داده‌های با ابعاد بالا هستند، چرا که معمولاً این داده‌ها غیرساخت‌یافته، شامل خطاهای سطح بالا و در فرمت‌های گوناگون هستند. به علاوه در داده‌های متنی بیشتر تغییر موضوع در زمان رخ می‌دهد. همین باعث شده است که کاوش در جریان‌های متنی یک کار دلهره‌آور باشد.

^۴ Bag of Word



**Iran University of Science and Technology
Computer Engineering Department**

A Survey on Learning Algorithms for Text Streams

**A Thesis Submitted in Partial Fulfillment of the Requirement for the Degree
of Master of Science in Computer Engineering**

By:

Vahid Kharazi

Supervisor:

Dr. Behroz Minaei Bidgoli

October 2016