انتخاب ویژگی غیرنظارتی مقیاس‌پذیر توسط یادگیری ماتریس و تئوری گراف دوقسمته

کوثر صالح‌نژاد[1] نگین دانشپور[2]

۱- دانشجوی کارشناسی ارشد- دانشکده مهندسی کامپیوتر- دانشگاه تربیت دبیر شهید رجایی- تهران- ایران
kosar.salehnezhad@yahoo.com

۲- دانشیار- دانشکده مهندسی کامپیوتر- دانشگاه تربیت دبیر شهید رجایی- تهران- ایران
ndaneshpour@sru.ac.ir

چکیده: با گسترش سریع تکنولوژی، حجم عظیمی از داده‌های بدون برچسب با ابعاد زیاد، نیاز به پردازش پیدا کردند. برای کاهش ابعاد، انتخاب ویژگی غیرنظارتی، به عنوان یک پیش‌مرحله مهم قبل از وظایف یادگیری ماشین، شناخته می‌شود. در این مقاله، یک روش انتخاب ویژگی غیرنظارتی پیشنهاد می‌شود. روش مذکور بر اساس گراف ماتریس و ماتریس وزنی، به صورت پویا و مقیاس‌پذیر عمل می‌کند. برای بهبود عمکرد این روش، به جای استفاده از تابع لاگرانژ در ساخت ماتریس وزنی، تئوری گراف دو قسمته اعمال می‌شود. انتخاب ویژگی روی گراف ماتریس انجام می‌شود. این گراف با به کارگیری k نزدیک‌ترین همسایه ساخته می‌شود، که روش را نسبت به نویز مقاوم‌تر می‌کند. همچنین ساختار سراسری داده‌ی اصلی، از طریق ساخت ماتریس وزن بازسازی‌شده با کمک محدودیت رتبه پایین، حفظ می‌شود. علاوه‌براین، نمره‌ی ویژگی، که به‌طور صریح قدرت‌مندی ویژگی‌ها را منعکس می‌کند، با کمک تابع Frobenius norm مدل می‌شود. روش پیشنهادی با روش‌های مشابه در سه معیار دقت کلاس‌بندی، حساسیت به پارامتر و پیچیدگی زمانی مقایسه شده‌است. آزمایش‌ها نشان می‌دهد که دقت کلاس‌بندی روش ارائه شده‌ی این مقاله، به طور متوسط ۲,۸۳٪ بهبود یافته‌است. همچنین پیچیدگی زمانی آن تا $\max\{O(n^2d),O(nm)\}$ کاهش یافته‌است.

واژه‌های کلیدی: داده‌کاوی، پیش‌پردازش، انتخاب ویژگی، روش غیرنظارتی، گراف

نوع مقاله: پژوهشی

DOI: 10.52547/jiaeee.20.3.135

تاریخ ارسال مقاله: ۱۴۰۰/۶/۲۳
تاریخ پذیرش مشروط مقاله: ۱۴۰۱/۰۳/۱۶
تاریخ پذیرش مقاله: ۱۴۰۱/۵/۲۴
نام نویسنده‌ی مسئول: دکتر نگین دانشپور
نشانی نویسنده‌ی مسئول: ایران – تهران – خیابان شعبانلو – دانشگاه تربیت دبیر شهید رجایی – دانشکده‌ی مهندسی کامپیوتر

۱- مقدمه

داده‌کاوی، پردازش اتوماتیک و نیمه‌اتوماتیک، برای استخراج و کشف الگوها و اطلاعات، از داده‌های بزرگ می‌باشد[۱]. با پیشرفت تکنولوژی، داده‌ها نیز به سرعت رشد پیدا کردند و بشر با حجم عظیمی از داده‌ها روبه‌رو شد. این داده‌ها گاهی دارای ابعاد بسیار زیادی هستند که برای آنالیز و تصمیم‌گیری آن‌ها ایجاد مشکل خواهند کرد. در بسیاری از برنامه‌ها مانند تشخیص الگو، داده‌کاوی و غیره، داده‌ها دارای ابعاد فراوانی هستند. این ابعاد زیاد داده‌ها، موجب افزایش زمان اجرا و فضای مصرفی برای پردازش داده‌ها، می‌شود. علاوه‌براین، برخی ویژگی‌های داده‌های با ابعاد بالا، نامرتبط و اضافی هستند، که نتایج وظایف یادگیری را ناکارآمد خواهند کرد. به طور کلی در کاربردهای علم داده به خوبی مشخص است که حدود ۸۰ درصد از زمان، صرف پیش-پردازش و آماده‌سازی داده می‌شود. همچنین تعداد بالای ویژگی‌های نامرتبط می‌تواند الگوریتم‌های طبقه‌بندی را به اشتباه بیاندازد و منجربه عملکرد پایین‌تر شود[۲]. به عبارتی اجرای داده‌کاوی روی اینچنین داده‌هایی موجب نتایج گمراه‌کننده‌ای می‌شود. بنابراین، پیش‌پردازش، یک مرحله‌ی ضروری برای تعیین و آماده‌سازی داده‌ی موردنیاز در هر مدل یادگیری می‌باشد[۳].

یکی از متداول‌ترین تکنیک‌های پیش‌پردازش، کاهش ابعاد[۱](DR) است، که تلاش می‌کند تا تعداد ویژگی‌های موجود در مجموعه‌ی داده را کمینه کند و کارایی داده‌کاوی را بهبود بخشد[۴]. الگوریتم‌های یادگیری ماشین، بر روی نمونه داده‌های آموزشی، یادگیری و مدل-سازی را انجام می‌دهند. به عبارتی فرآیند یادگیری، الگوهای موجود در داده‌ها را استخراج می‌کند و به پیش‌بینی اطلاعات داده‌ها می-پردازد[۵]. به این ترتیب کاهش ابعاد یک مسئله‌ی مهم در کاربردهای یادگیری ماشین، در مواجهه با مجموعه داده‌های با ابعاد بالا می‌باشد، که دو سودمندی دارد. اول، به کاهش پیچیدگی محاسباتی و مصرف حافظه کمک می‌کند. دوم، مجموعه داده‌های با ابعاد بالا، تعدادی ویژگی افزونه و نامرتبط دارند که بر روی کارایی مدل‌های یادگیری ماشین تأثیر منفی می‌گذارند و کاهش ابعاد، آن‌ها را از بین می‌برد[۶]. دو راه برای کاهش ابعاد معرفی شده‌است: انتخاب ویژگی[۲](FS) و استخراج ویژگی[۷]. انتخاب ویژگی، به دنبال پیداکردن زیرمجموعه‌ای از ویژگی‌ها می‌باشد؛ درحالیکه، استخراج ویژگی، داده‌های با ابعاد بالا را به داده‌های با ابعاد کوچک تبدیل می‌کند و روی داده‌های اصلی تغییراتی ایجاد می‌کند. به عبارت دیگر استخراج ویژگی شامل تبدیلی خطی یا غیرخطی از فضای ویژگی‌های اصلی به فضایی کم‌بعدتر می-شود[۸]. از آنجاییکه، انتخاب ویژگی منطق متغیرهای اصلی را تغییر نمی‌دهد، فواید بیشتری نسبت به استخراج ویژگی دارا می‌باشد[۳]. به این ترتیب استفاده از روش‌های انتخاب ویژگی موجب ساخت زیرمجموعه‌ای از مجموعه‌ی داده‌ی ورودی می‌شود. زیرمجموعه‌ی به-دست‌آمده شامل ویژگی‌های مرتبط و مهم انتخاب‌شده از مجموعه‌ی کل ویژگی‌ها می‌باشد. بنابراین مشکل ابعاد زیاد داده‌ها را حل می‌کند. بر این اساس روش‌های انتخاب ویژگی فقط ویژگی‌های مهم را به طوریکه به کیفیت مجموعه‌ی داده آسیبی وارد نشود، دریافت می-کنند[۹]. در این مقاله نیز یک روش انتخاب ویژگی برای کاهش ابعاد هرچه بهتر داده‌های حجیم معرفی می‌شود.

به طور کلی انتخاب ویژگی، زیرمجموعه‌ی کمینه‌ی ویژگی‌ها از مجموعه‌ی اصلی را با در نظر گرفتن یک سری شاخص‌های اندازه‌گیری انتخاب می‌کند. گاهی داده‌ها دارای ویژگی‌های نامرتبط و افزونه‌ای هستند، که تاثیر منفی در عملکرد می‌گذارند. ویژگی‌های نامرتبط آن دسته از ویژگی‌هایی هستند، که اطلاعات مفیدی نمی‌دهند. به عبارتی، در خوشه‌بندی و کلاس‌بندی، تأثیرگذار نیستند. ویژگی‌های افزونه نیز، آن دسته از ویژگی‌هایی هستند که اطلاعات آن‌ها از دیگر ویژگی‌ها نیز به دست می‌آید. بنابراین روش‌های انتخاب ویژگی، ویژگی‌های مفید و مرتبط را انتخاب می‌کنند [۱۰].

همانطور که گفته‌شد، انتخاب ویژگی، در بخش‌های مختلفی مثل متن‌کاوی، پردازش تصویر، بیوانفورماتیک، کاربردهای صنعتی و غیره استفاده می‌شود. در متن‌کاوی، تعداد زیادی کلمه در یک سند موجود می‌باشند و ویژگی‌ها مشخصه‌های کلمات را نشان می‌دهند. در پردازش تصویر، تعداد زیادی ویژگی برای تصاویر وجود دارد، که انتخاب مجموعه‌ای از این ویژگی‌ها کار ساده‌ای نیست. از دیگر کاربردهای انتخاب ویژگی، در کشفیات زیستی داده‌های ژنی می‌باشد. وقتی ویژگی‌هایی با بیشترین ارتباط انتخاب می‌شوند، اطلاعات مهمی درباره‌ی داده‌های ژنی به‌دست می‌آید. روش‌های انتخاب ویژگی در بیوانفورماتیک برای پیداکردن ساختار ذاتی موروثی ژن‌ها، کاهش ابعاد، ارتباط بین ژن‌ها و کاربردهای دیگر نیز به‌کارگرفته می‌شوند [۱۱].

به طورکلی، رویکرد روش‌های انتخاب ویژگی براساس سه دیدگاه تقسیم می‌شوند. اولین دیدگاه براساس برچسب داده‌ها می‌باشد. بسته به نوع مسئله‌ی مورد بررسی، برچسب‌ها می‌توانند متفاوت باشند. به طور کلی برچسب‌ها به ما نشان می‌دهند که هر نمونه از مجموعه‌ی داده به چه کلاس و دسته‌ای تعلق دارد. ما به کمک ویژگی‌ها تلاش می‌کنیم تا برچسب هر نمونه را پیش‌بینی کنیم. همانطور که گفته‌شد، برچسب‌ها می‌توانند انواع مختلفی باشند. برای مثال در حوزه‌ی پردازش تصویر، برچسب می‌تواند یکی از ویژگی‌های مجموعه‌ی داده مانند رنگ، باشد. همچنین می‌توان برای هر تصویر یک نوع در نظر گرفت و برچسب‌ها، نوع تصاویر را نشان می‌دهند. برای نمونه، گربه و سگ می‌توانند دو برچسب باشند. به این ترتیب به هر نمونه از مجموعه‌ی داده می‌توان برچسب گربه یا سگ را اختصاص داد. بنابراین زمانیکه ما می‌خواهیم پیش‌بینی کنیم که تصویر مورد پردازش، گربه است یا سگ، برچسب گربه و سگ را می‌توانیم در نظر بگیریم، تا به هر نمونه یا تصویر اختصاص دهیم[۱۲].

روش‌های انتخاب ویژگی ، براساس میزان دسترسی به برچسب به سه دسته‌ی نظارتی، غیرنظارتی و نیمه‌نظارتی تقسیم می‌شوند. مجموعه‌ی

داده‌ها در روش‌های نظارتی دارای برچسب، در روش‌های غیرنظارتی فاقد برچسب و در نیمه‌نظارتی‌ها، فقط تعدادی از آن‌ها دارای برچسب می‌باشند. انتخاب ویژگی نظارتی، اغلب به مسائل کلاس‌بندی تمایل دارد و از ارتباط بین ویژگی‌ها و برچسب کلاس آن‌ها استفاده می‌کند. اهمیت یک ویژگی، با استفاده از معیارهای اندازه‌گیری ارتباط، می‌تواند ارزیابی شود. مدل نظارتی تلاش می‌کند تا بهترین زیرمجموعه‌ای از ویژگی‌ها را پیدا کند که با آن‌ها دقت کلاس‌بندی به بیشترین مقدار ممکن برسد [۷].

همانطور که گفته‌شد برچسب‌ها می‌توانند مشخص کنند که هر مجموعه‌ای از ویژگی‌ها، چه حیوانی را نشان می‌دهد، موقعیت یک بازیکن بولینگ در چه وضعیتی است و یا اینکه چقدر یک فرد خوشحال است. چنین برچسب‌هایی اغلب در دسترس نیستند؛ به ویژه در مواردی مانند متن‌کاوی، بیوانفورماتیک و رسانه‌های اجتماعی که داده‌ها دارای ابعاد بالا هستند. علاوه‌براین برچسب‌گذاری داده‌ها هم از نظر زمان هم از نظر هزینه، گران است. زیرا برچسب باید دقیق باشد و به نیروی واجد شرایط نیاز دارد. به همین دلیل روش‌های انتخاب ویژگی غیرنظارتی به دلیل اینکه بدون داشتن دانش قبلی به خوبی عمل می‌کنند از محبوبیت خاصی برخوردارند[۱۳].

روش‌های انتخاب ویژگی غیرنظارتی به دلیل پرهزینه بودن داده‌های برچسب‌دار از نظر زمانی، به‌کارگرفته شده‌اند [۳]. آن‌ها به این‌گونه عمل می‌کنند که به شناسایی ویژگی‌های مرتبط بدون نیاز به برچسب‌شان می‌پردازند. روش‌های انتخاب ویژگی غیرنظارتی دو فایده‌ی اساسی دارند: ۱) به داشتن اطلاعاتی از قبل نیاز ندارند و به همان خوبی کار می‌کنند. ۲) می‌توانند وقتی انتخاب ویژگی نظارتی نمی‌داند با کلاس جدیدی که وارد مجموعه‌ی داده می‌شود چه کند، خودشان را با شرایط جدید مطابقت دهند [۱۰]. براین‌اساس در این مقاله از روش انتخاب ویژگی غیرنظارتی برای ارائه‌ی یک روش انتخاب ویژگی استفاده شده‌است. انتخاب ویژگی غیرنظارتی با کمک خود داده‌ها و توسط یک شاخص نمره تلاش می‌کند تا خصیصه‌های خاصی از داده‌ها را حفظ کند [۱۴]. بنابراین بدون اطلاعات برچسب، ارتباط بین نمونه‌های داده را می‌توان استخراج کرد و با کمک آن‌ها ویژگی‌های مرتبط را انتخاب کرد. برخی از آن‌ها، به عنوان یکی از روش‌های کاهش ابعاد، زیرمجموعه‌ای از ویژگی‌های مرتبط را که شامل مهم‌ترین اطلاعات از داده‌ی اصلی هستند، انتخاب می‌کنند. در این حین نیز، ساختار هندسی ذاتی داده‌ها، بدون استفاده از برچسب‌ها حفظ می‌شود. روش‌های قبلی پیشنهادشده، به طور کلی، کارایی خوبی تا کنون داشته‌اند، اما به دلیل اینکه متعلق به مدل براساس بردار می‌باشند، ممکن است که در تبدیل ماتریس داده به بردارها، ناکارآمد باشند. دلیل آن این است که بردارسازی، اطلاعات مکان مقادیر ماتریس اصلی را نادیده می‌گیرد، که برای حل این مسئله، ماتریس پراکندگی پیشنهادشده‌است. اما ماتریس پراکندگی متعلق به یادگیری نظارتی می‌باشد، که نیازمند اطلاعات برچسب است [۳].

انتخاب ویژگی نیمه‌نظارتی[۳](SSFS) نیز تنها نیاز دارد که تعدادی از اشیاء برچسب داشته باشند. مجموعه‌ی داده در این مدل به صورت $D=\{D_L,D_U\}$ می‌باشد، که در آن D_L، نمونه‌های دارای برچسب و D_U، نمونه‌های بدون برچسب می‌باشند. مدل نیمه‌نظارتی با استفاده از آموزش به دست آمده از D_L، یادگیری را روی D_U بهبود می‌بخشد [۷]. علاوه‌براین، نوعی روش انتخاب ویژگیی نیمه‌نظارتی نیز مطرح می‌شود، که در آن برچسب‌ها به دسته‌ای از نمونه‌ها نسبت داده می‌شوند. به عبارت دیگر برچسب‌ها، به هر نمونه به صورت جدا اختصاص داده نمی‌شوند. لازم به ذکر است که امکان وجود نمونه‌های نامرتبط در این دسته‌ها نیز وجود دارد[۱۵].

دومین دیدگاه که دیدگاه نویسندگان این مقاله است، بر پایه‌ی انجام انتخاب ویژگی می‌باشد، که به روش‌های براساس گراف، ماتریس قسمت‌بندی، جست‌وجو و رگرسیون تقسیم می‌شوند. الگوریتم‌های یادگیری روش‌های براساس گراف، به دلیل هزینه‌ی محاسباتی کمتر، دقت بالاتر و قابلیت پیش‌بینی بیشتر، مورد توجه خاصی قرار گرفته‌اند. به همین دلیل در این مقاله نیز یک روش انتخاب ویژگی براساس گراف پیشنهاد شده‌است. الگوریتم یادگیری براساس گراف، همه‌ی داده‌ها را به صورت یک گراف وزنی $\tau(X,E)$ مدل می‌کند؛ که $X=[x_1,\ldots,x_l,x_{l+1},\ldots,x_n]^T$، نشان‌دهنده‌ی نمونه‌هاست، که n طول بردار X و برابر با تعداد نمونه‌ها می‌باشد. E نیز مجموعه‌ای از یال‌هایی که بین دو زوج گره ارتباط برقرار می‌کند و بیانگر شباهت بین نمونه‌ها است [۱۶]. از چالش‌های روش‌های این دسته، توانایی ثبت تغییرات گراف به‌طور مؤثر است. همچنین در اکثر این روش‌ها با وجود اینکه نتایج خوبی را در بعضی وظایف کلاس‌بندی، به‌دست می‌آورند، اما همچنان محدودیت‌هایی را در کاربرد واقعی دارا می‌باشند. اول اینکه، گراف ماتریس را براساس داده‌ی اصلی می‌سازند، که اغلب ویژگی‌های پرت و افزونه دارند. بنابراین موجب می‌شوند که مدل‌های انتخاب ویژگی تخریب شوند. دوم، حفظ ساختار سراسری داده‌ها و ویژگی‌ها، هنگام ساخت گراف ماتریس در نظر گرفته نمی‌شود، که ساختار سراسری داده‌ها می‌تواند اطلاعاتی که در ساختار محلی داده‌ها از دست رفته‌است را نشان دهد [۱۴]. روش پیشنهادی این مقاله، این نواقص را رفع کرده‌است و یک روش انتخاب ویژگی پویا و مقیاس‌پذیر را که هر دو ساختار محلی و سراسری داده‌ها را حفظ می‌کند، معرفی می‌کند. دسته‌ی دوم، روش‌های براساس ماتریس قسمت‌بندی هستند. انتخاب ویژگی قسمت‌بندی ماتریس، به یادگیری زیرفضاها می‌پردازد. در یادگیری ماشین و داده‌کاوی، تکنیک یادگیری زیرفضا، بسیار مطالعه‌شده‌است و در کاربردهای زیادی استفاده شده‌است. این روش‌ها معمولاً یک نماینده‌ی کم‌بعد را از فضای با ابعاد بالا یاد می‌گیرند تا ساختار سراسری داده را استخراج کنند [۱۷].

دسته‌ی سوم، روش‌های براساس جست‌وجو هستند. روش‌های انتخاب ویژگی براساس جست‌وجو به دنبال پیداکردن بهترین زیرمجموعه از ویژگی‌های ورودی می‌باشند. زیرمجموعه‌ی انتخاب‌شده

را می‌توان با استفاده از یک سری اندازه‌گیری‌های آماری (مانند ارتباط بین ویژگی‌ها و برچسب کلاس)، و اندازه‌گیری کلاس‌بندی(مانند دقت)، ارزیابی کرد. سپس پروسه تا زمان رسیدن به محدودیت‌ها ادامه می‌یابد و تکرار می‌شود. در آخر نیز اغلب بعد از پروسه‌ی انتخاب ویژگی برای بررسی کیفیت نتایج زیرمجموعه‌ی انتخاب ویژگی، اعتبارسنجی انجام می‌شود [۱]. دسته‌ی آخر روش‌های براساس رگرسیون هستند، که در روش‌های نظارتی، ارتباط بین ویژگی‌ها و برچسب‌ها را ارزیابی می‌کنند. روش‌های غیرنظارتی نیز با رویکردهای مختلف، عمل می‌کنند، برخی از آن‌ها ویژگی‌ها را به صورت ترکیب خطی از ویژگی‌های مرتبط به آن، نمایش می‌دهند [۳].

آخرین دیدگاه روش‌های انتخاب ویژگی، بر اساس نوع خروجی هستند، که به رتبه‌بندی ویژگی و نمره‌دهی ویژگی تقسیم می‌شوند. اولی ویژگی‌ها را براساس یک سری معیارهای اندازه‌گیری بررسی کرده و مرتب می‌کند؛ و تعدادی از ویژگی‌های بالای این لیست را انتخاب می‌کند. دومی با استفاده از مقادیر تناسب[۴]، ویژگی‌های قطعی را مشخص می‌کند [۱۸]. از مزایای رتبه‌بندی ویژگی این است که می‌توان با مشخص‌کردن تعداد ویژگی‌های مورد نظر(k)، k ویژگی بالای لیست را انتخاب کرد. هر چند مشخص‌کردن مقدار k مناسب، که ویژگی‌های سودمند واقعی بدون افزونگی و نامرتبطی را به ما دهد، خود یک مسئله‌ی چالش‌انگیز است. از طرفی نمره‌دهی ویژگی به دلیل نامشخص‌بودن مقدار k، ممکن است که ویژگی‌هایی کم‌تر یا بیشتر از ویژگی‌های مورد نیاز را مشخص کند، که این امر کارایی این دسته از الگوریتم‌ها را کاهش می‌دهد [۱۹]. بنابراین در این مقاله از روش‌های رتبه‌بندی ویژگی برای ارائه‌ی یک روش انتخاب ویژگی استفاده شده‌است.

با توجه به مطالب گفته‌شده، در این مقاله یک روش انتخاب ویژگی براساس گراف پیشنهاد می‌شود. زیرا روش‌های براساس گراف، از دقت و کارایی بالایی برخوردار می‌باشند. روش پیشنهادی با ارائه‌ی یک روش انتخاب ویژگی و به کارگیری تئوری گراف دوقسمته، روش انتخاب ویژگی پیشنهادشده در[۱۴] را بهبود می‌بخشد. از چالش‌های مهم روش‌های انتخاب ویژگی براساس گراف، مقیاس‌پذیری و پویایی آن‌ها می‌باشد. چالش پویایی، چالشی است که در [۱۴] مورد بررسی قرار گرفته‌است و در آن ساخت گراف ماتریس و ماتریس وزنی به صورت پویا و در یک حلقه‌ی تکرار، انجام می‌شود. اما، چالش مقیاس پذیری در روش ارائه‌شده در این مقاله، مورد توجه قرار نگرفته است. بنابراین، در روش پیشنهادی با افزودن ویژگی مقیاس‌پذیری و اعمال آن در روش ارائه‌شده در [۱۴]، عملکرد آن را بهبود می‌بخشیم. مقیاس‌پذیری، پیچیدگی زمانی الگوریتم ارائه‌شده را تا حد قابل توجهی کاهش می‌دهد که موجب کارآمدی و هرچه بهتر شدن این روش در داده‌های با مقیاس بالا می‌شود.

به این ترتیب، روش پیشنهاد شده، هم به حفظ ساختار سراسری و محلی داده‌ها می‌پردازد، هم دو ویژگی پویایی و مقیاس‌پذیری را در نظر گرفته است. حفظ ساختار سراسری و محلی داده‌ها، هر کدام به خودی خود دارای اطلاعات به خصوصی از داده‌ها می‌باشند و در نظر نگرفتن هر کدام از این دو، بخشی از اطلاعات به دست آمده از داده‌ها را از بین می‌برد. دو ویژگی پویایی و مقیاس‌پذیری نیز، سبب می‌شوند تا علاوه‌بر ساخت ماتریس وزنی به صورت پویا، پیچیدگی زمانی و محاسباتی روش پیشنهادی را بهبود بخشد و رویکرد آن را در مواجهه با داده‌های با مقیاس بالا تقویت کند. بنابراین در روش پیشنهادی این مقاله، گراف ماتریس در یک حلقه‌ی تکرار به صورت پویا همراه با ماتریس وزنی ساخته می‌شود. در ساخت ماتریس وزنی با بهره‌گیری از نتایج به دست‌آمده از k نزدیک‌ترین همسایه، تأثیر داده‌های دورافتاده کاهش می‌یابد تا ساختار محلی داده‌ها حفظ شود. علاوه‌براین برای ایجاد ویژگی مقیاس‌پذیری در روش پیشنهادشده، گراف ماتریس با کمک تئوری گراف دو قسمته ساخته می‌شود؛ و پس از محاسبه‌ی ماتریس وزن بازسازی با کمک محدودیت رتبه پایین[۵]، به نمره‌دهی ویژگی‌ها پرداخته می‌شود، تا یک لیست مرتب شده از ویژگی‌ها را به دست آورد.

ادامه‌ی مقاله به این شرح است: در بخش ۲، به مرور روش‌های انتخاب ویژگی در دسته‌بندی‌های مختلف پرداخته می‌شود. در بخش ۳، یک بررسی کوتاه در مورد چارچوب تئوری گراف دو قسمته و محدودیت رتبه پایین ارائه می‌شود. پس از آن، در بخش ۴ روش انتخاب ویژگی غیرنظارتی پیشنهادی معرفی می‌شود. بخش ۵ نتایج آزمایشات را نشان می‌دهد. سرانجام، نتیجه گیری در بخش ۶ ارائه می‌شود.

۲- تحقیقات پیشین

انتخاب ویژگی، از مراحل پیش‌پردازش داده‌ها، برای کاهش ابعاد ویژگی‌ها، با حذف ویژگی‌های نامرتبط و افزونه می‌باشد؛ که به این ترتیب به ویژگی‌های سودمند و مرتبطی دست پیدا خواهد کرد. این ویژگی‌ها، زیرمجموعه‌ای از ویژگی‌های اصلی می‌باشند، که به عنوان نماینده‌ی ویژگی‌های اصلی مورد استفاده قرار می‌گیرند، و موجب بهبود کارایی الگوریتم‌های یادگیری و کاهش پیچیدگی زمانی محاسبات و مصرف حافظه می‌شوند. روش‌های انتخاب ویژگی براساس دسترسی به برچسب داده‌ها به سه دسته‌ی نظارتی، نیمه‌نظارتی و غیرنظارتی تقسیم می‌شوند. روش‌های نظارتی، آن دسته از روش‌هایی هستند که به برچسب داده‌ها، دسترسی دارند. این روش‌ها به دلیل نیازمندی به برچسب داده‌ها، کمتر مورد استفاده قرار می‌گیرند. زیرا در عمل، داده‌های موجود در دنیای واقعی بدون برچسب هستند و برچسب‌دار کردن آن‌ها نیز بسیار پرهزینه است. روش ارائه شده در [۱۸] یک روش نظارتی است که در آن، از مدل رگرسیون خطی استفاده شده‌است. در این روش، با کمک ستون برچسب‌ها، زیرمجموعه‌ای از ویژگی‌ها را انتخاب می‌کند که ترکیب خطی از آن‌ها، ستون برچسب را نتیجه دهد. همچنین در آن از زاویه‌ی ویژگی نیز، که

شامل زاویه‌ی بین هر ویژگی و برچسب است، استفاده می‌شود. در [۲۰] نیز از روش‌های نظارتی استفاده شده‌است. نویسندگان در [۲۰] با استفاده از مقدار تجزیه‌ی مفرد به کمینه‌کردن ویژگی‌های افزونه می‌پردازند و با استفاده از اطلاعات Gain، نرخ ویژگی‌های مرتبط را بیشینه می‌کنند. در RFS [۲۱][۶] نیز، از $L_{2,1}$-norm براساس عملکرد ضرر[۷] و تنظیم $L_{2,1}$-norm به صورت مشترک برای اجرای انتخاب ویژگی با کمک اطلاعات برچسب استفاده می‌شود و یادگیری زیرفضا را در نظر نمی‌گیرد.

دسته‌ی دوم، روش‌های غیرنظارتی هستند. این روش‌ها دارای داده‌های بدون‌برچسب می‌باشند و فارغ از نیاز به برچسب داده‌ها می‌باشند و با بررسی خصیصه‌های ذاتی داده‌ها، به انتخاب ویژگی می‌پردازند. معیارهای ارزیابی خصیصه‌های ذاتی بین داده‌ها و پیدا کردن ویژگی‌های مرتبط و غیرافزونه، بدون راهنمایی برچسب‌ها از مهم‌ترین چالش‌های رویکردهای موجود در این دسته می‌باشند. از رویکردهای ارائه شده در این دسته می‌توان به رویکرد ارائه‌شده در [۲۲] اشاره کرد که با حفظ ساختار محلی و استفاده از یک فضای نماینده‌ی نهفته در داده‌ها به جای فضای اصلی داده‌ها، به انتخاب ویژگی می‌پردازد. رویکرد ارائه شده در [۲۳] نیز، با به‌کارگیری گراف، به حفظ ساختار محلی در انتخاب ویژگی می‌پردازد. رویکرد ارائه شده در [۲۴]، با کمک یادگیری زیرفضا، به حفظ ساختارهای محلی می‌پردازد و از خصیصه‌های ذاتی داده‌ها برای دستیابی به این ساختارها بدون راهنمایی برچسب، استفاده می‌کند. در [۲۵]، ایده‌ی اصلی، حفظ توزیع متعادل داده‌ها میان خوشه‌ها می‌باشد و خوشه‌بندی متعادل را با انتخاب ویژگی ادغام می‌کند. در [۲۶]، از یادگیری زیرفضا استفاده می‌شود و چندین فضا از داده‌ها استخراج می‌شود. در این روش، ویژگی‌ها در زیرفضاهای مختلف، و نه تنها یک زیرفضای خاص، به رقابت با یکدیگر می‌پردازند. در LS[۸] [۲۷] نیز، اهمیت ویژگی‌ها با کمک نمره‌ی لاپلاس آن‌ها ارزیابی می‌شود. هدف LS حفظ ساختار محلی داده‌ها است. این روش نیز از یادگیری زیرفضا بهره‌ای نمی‌برد. در روش ارائه‌شده در [۲۸] نیز یک روش انتخاب ویژگی غیرنظارتی مبتنی بر رمزگذاری خودکار ارائه می‌شود. این روش از دو تابع Loss و فعالسازی منعطف بهره می‌برد. به این ترتیب که از طریق تابع اول گستردگی وظایف یادگیری را متناسب می‌کند و با کمک تابع دوم نیز، منظم‌سازی[۹] را در مدل پیشنهادی اعمال می‌کند.

آخرین دسته، نیمه‌نظارتی‌ها هستند که در آن‌ها برخی از دادها دارای برچسب و برخی بدون‌برچسب می‌باشند. از رویکردهای این دسته می‌توان به روش ارائه شده در [۱۶] اشاره کرد که در آن با ترکیب دو روش با نام‌های SSFS[۱۰] و SPL[۱۱]، روش MASFS معرفی می‌شود. در این روش گراف ساخته‌شده در SSFS که ثابت است و تغییر نمی‌کند، با کمک SPL، به طور انطباقی تغییر می‌کند. این روش از یادگیری چندنما برای استفاده‌ی کامل اطلاعات نماهای مختلف استفاده می‌کند. در [۲۹] نیز، به حفظ هر دو ساختار سراسری و محلی پرداخته شده‌است، و با کمک یادگیری گراف و یادگیری پراکندگی به ساختار محلی و سراسری دست پیدا می‌کند. در روش پیشنهاد شده‌ی این مقاله از روش انتخاب ویژگی غیرنظارتی که به اطلاعات قبلی نیاز ندارد و وفق‌پذیری آن نیز بیشتر است استفاده می‌شود.

علاوه‌بر این، روش‌های انتخاب ویژگی برمبنای اساس کار خود نیز به ۴ دسته‌ی گراف، قسمت‌بندی، جست‌وجو و رگرسیون تقسیم می‌شوند. روش‌های براساس گراف، با استفاده از ماتریس شباهت، به دستیابی به ساختار داده‌ها می‌پردازند و تلاش می‌کنند تا ویژگی‌هایی را انتخاب کنند، که این ساختارها را حفظ کنند. از رویکردهای این دسته، می‌توان به روش ارائه شده در [۱۴] اشاره کرد، که در آن از یادگیری زیرفضا برای حفظ ساختار سراسری و بهبود روش LLE[۱۲] استفاده می‌شود. LLES[۱۳] [۳۰] نیز، ایده‌ی LLE را در چارچوب انتخاب ویژگی حفظ گراف، اجرا می‌کند و با بررسی اختلاف بین ساختار محلی هر ویژگی و داده‌ی اصلی، به انتخاب ویژگی‌های مرتبط می‌پردازد. این روش ماتریس گراف را براساس داده‌ی اصلی می‌سازد و فقط ساختار محلی نمونه‌ها و ویژگی‌ها را در نظر می‌گیرد. روش ارائه شده در [۳۱] نیز با نام DGSPSFS معرفی می‌شود، که این روش از ساختار سراسری دوگانه استفاده می‌کند که قادر به حفظ دو ساختار سراسری به طور هم‌زمان است. همچنین این روش ماتریس پاسخ را از طریق یادگیری پراکندگی می‌سازد که موجب می‌شود تا این ماتریس وابسته به برچسب داده‌ها نباشد. در[۳۲] و [۴] نیز از روش‌های براساس گراف استفاده شده‌است. [۳۲]، به حفظ ساختار محلی می‌پردازد و ایده‌ی اصلی آن در نظر گرفتن ارتباط ویژگی‌ها هم با داده‌ها و هم با خود ویژگی‌ها می‌باشد؛ اما [۴]، از گراف دوقسمته استفاده می‌کند و با نگاشت داده‌ها به یک گراف کوچک‌تر، به انتخاب ویژگی می‌پردازد. در MCFS[۱۴] [۳۳] نیز، از گراف نماینده استفاده می‌شود و رگرسیون حداقل مربع را برای حفظ ساختار خوشه چندگانه‌ی مجموعه‌های داده به کار می‌گیرد و از ساختار محلی داده‌ها استفاده می‌کند.

دسته‌ی دوم روش‌های براساس ماتریس قسمت‌بندی می‌باشند، که با به‌کارگیری یادگیری زیرفضا، به تجزیه‌ی ماتریس می‌پردازند، مانند روش ارائه شده در [۱۷] که به کمک یادگیری زیرفضا، ساختار سراسری داده‌ها دریافت می‌شود و به کمک آن ماتریس وزن ویژگی‌ها به روزرسانی می‌شود. علاوه‌براین با به کارگیری تنظیم norm مناسب، حساسیت الگوریتم به نویز و دورافتاده کاهش می‌یابد. در [۳۴]، نویسندگان ابتدا با استفاده از روش‌های لاپلاسی به ساخت شبه‌برچسب‌ها می‌پردازند. سپس ارتباط ویژگی‌ها با شبه‌برچسب‌ها را در یک ماتریس شباهت مشخص می‌کنند.

دسته‌ی سوم، روش‌های براساس جست‌وجو می‌باشند، که به دنبال پیداکردن بهترین زیرمجموعه‌ی ویژگی‌ها، از میان ویژگی‌های اصلی می‌باشند. برای نمونه در [۱] از استراتژی جست‌وجوی گرانشی با نام GSA[۱۵] به همراه عامل‌های ژنتیک برای انتخاب بهترین زیرمجموعه‌ی

ویژگی‌ها استفاده می‌شود. در [۱۱] از ترکیب خوشه‌بندی و سیستم دودویی مورچه با نام BAS[۱۶] استفاده می‌شود و در داخل و خارج خوشه، جست‌وجوی محلی و سراسری انجام می‌شود. این مقاله از عامل جهش نیز برای فرار از بهینه‌ی محلی استفاده می‌کند. در [۳۵]، از روش‌های ACO استفاده می‌شود. اما در این روش، برای ارزیابی زیرمجموعه‌ی ویژگی‌ها و به روزرسانی مقدار فرمون، از خصیصه‌های ذاتی داده‌ها استفاده می‌شود تا میزان مطلوبیت هر ویژگی به همراه تراکم فرمون برای آن ویژگی، در نظر گرفته شود.

در آخر، روش‌های براساس رگرسیون، نوعی ارتباط بین متغیرها پیدا می‌کنند، تا با راهنمایی آن، ویژگی‌های مرتبط را انتخاب کنند. از روش‌های این دسته، نویسندگان [۳] با به کارگیری تکنیک‌های خوشه‌بندی، به ساخت شبه‌برچسب‌ها می‌پردازند و در نهایت با استفاده از رگرسیون بین ماتریس داده و شبه‌برچسب‌ها، انتخاب ویژگی را انجام می‌دهند. در [۳۶] با به کارگیری یادگیری چندنما، و اختصاص برچسب به هر نما و استفاده از رگرسیون میان هر نما و برچسب، وزن ویژگی‌ها محاسبه می‌شود. همچنین در RSR[۱۷] [۳۷]، ویژگی خودنمایندگی را برای توصیف همه‌ی ویژگی‌ها به کار می‌گیرد و از $L_{2,1}$-norm روی ماتریس ضریب خودنمایندگی استفاده می‌کند تا ویژگی‌های نماینده را انتخاب کند و ویژگی‌های نامرتبط و افزونه را حذف کند. همچنین ویژگی‌های محلی را با خاصیت خودنمایندگی در نظر می‌گیرد. در روش پیشنهاد شده‌ی این مقاله از روش‌های انتخاب ویژگی براساس گراف به دلیل هزینه محاسباتی کم، دقت و کارایی بالا و همچنین قابلیت پیش‌بینی بیشتر، استفاده می‌شود.

در جدول (۱)، الگوریتم‌های انتخاب ویژگی ، براساس نقاط قوت و ضعف باهم مقایسه می‌شوند.

۳- مفاهیم اولیه

از آنجاییکه در روش پیشنهادی این مقاله از تئوری گراف دوقسمته و محدودیت رتبه پایین استفاده شده‌است، در این بخش به شرح مختصر آن‌ها می‌پردازیم. در ۳-۱ نمادهای استفاده‌شده در این مقاله مرور می‌شوند. در ۳-۲ و ۳-۳ نیز به ترتیب تئوری گراف دو قسمته و محدودیت رتبه پایین توضیح داده می‌شوند.

۳-۱- نمادها

در این مقاله، ماتریس‌ها با حروف بزرگ و بردارها با حروف کوچک، نمایش داده می‌شوند. همچنین ترانهاده‌ی ماتریس X را با X^T، و وارون آن را با X^{-1}، 2-norm با $\|.\|_2$ و Frobenius norm با $\|.\|_F$، نشان داده می‌شوند. همچنین $X \in \mathbb{R}^{n*m}$ ، نشان می‌دهد که X یک ماتریس n در m است که مقادیر المان‌های آن اعداد حقیقی است.

۳-۲- تئوری گراف دو قسمته

اخیراً، برای کاهش پیچیدگی و تسریع زمان اجرای الگوریتم‌ها، در داده‌های با مقیاس بالا، از تئوری گراف دو قسمته استفاده می‌شود. داده‌ی اصلی دارای تعداد زیادی نقاط داده‌ی نمونه (n) می‌باشد. بنابراین به وسیله‌ی این گراف، میزان نزدیکی نقاط زیاد داده‌ی اصلی به نقاط لنگر[۱۸]، که تعداد این نقاط لنگر(m) بسیار کم‌تر از نقاط داده‌ی اصلی می‌باشد ($m \leq n$)، نشان داده می‌شود. با کمک نقاط لنگر، گراف بدون جهت دو قسمته‌ی $\beta(X,P,\varepsilon,B)$ ساخته می‌شود، که در آن $X=\{x_1,x_2,..,x_n\}$ نقاط داده، $P=\{p_1,p_2,\dots p_m\}$ نقاط لنگر، ε شامل یال‌های بین X و P و $B=[b_{ij}]_{n*m}$ ماتریس وابستگی را نشان می‌دهد. بنابراین اگر بین x_i و p_j یال وجود داشته باشد، آنگاه b_{ij}، وزن مثبتی می‌گیرد. در غیر این صورت مقدار آن صفر خواهد بود. هر چقدر مقدار b_{ij} بزرگتر باشد، شباهت بین x_i و p_j نیز بیشتر خواهد بود [۴]

جدول (۱): مقایسه‌ی روش‌های ارائه‌شده

روش	سال	نوع الگوریتم	الگوریتم پایه	اساس کار	نقاط قوت	مشکلات و چالش‌ها
[28]	۲۰۲۱	غیرنظارتی	-	رگرسیون و ماتریس	• مشخص کردن ویژگی‌ها با رابطه‌ی خطی یا غیرخطی	• در نظر نگرفتن ارتباط محلی و سراسری داده‌ها
FSBC [25]	۲۰۱۹	غیرنظارتی	-	خوشه-بندی	• بهبود دقت الگوریتم یادگیری	• در نظر نگرفتن ساختار سراسری • در نظر نگرفتن ارتباط بین ویژگی‌ها • وابسته به کارایی و دقت الگوریتم‌های یادگیری
FSCBAS [11]	۲۰۱۹	غیرنظارتی	-	جست و جو	• سرعت بالا • جست‌وجوی محلی و سراسری	• وابسته به دقت الگوریتم‌های خوشه-بندی
HGSA [1]	۲۰۱۹	غیرنظارتی	GSA	جست و جو	• سادگی پیاده‌سازی • فرار از بهینه‌ی محلی	• در نظر نگرفتن ساختارها
JSMRNS [3]	۲۰۱۹	غیرنظارتی	-	رگرسیون	• بهره گرفتن از راهنمایی شبه برچسب‌ها	• پیچیدگی زمانی برای تخصیص شبه برچسب

						• از دست رفتن برخی اطلاعات با به کارگیری رگرسیون خطی • وابسته به دقت الگوریتم خوشه‌بندی
LDSSL [24]	۲۰۱۹	غیرنظارتی	-	قسمت-بندی	• حفظ ساختار محلی	• ارتباط بین ویژگی‌ها را در نظر نمی-گیرد.
LLES [30]	۲۰۱۷	غیرنظارتی	LLE	گراف	• در نظرگرفتن ساختار محلی	• نادیده گرفتن ساختار سراسری • عدم مقیاس‌پذیری
LRLMR [22]	۲۰۱۹	غیرنظارتی	-	قسمت-بندی و گراف	• مقاوم بودن نسبت به نویز • حفظ ساختار محلی	• عدم مقیاس‌پذیری • درنظر نگرفتن ساختار سراسری
LS [27]	۲۰۰۶	غیرنظارتی	-	گراف و نمره‌ی لاپلاس	• حفظ ساختار محلی	• به کارنبردن یادگیری زیرفضا • نادیده گرفتن ساختار سراسری • عدم مقیاس‌پذیری
MCFS [33]	۲۰۱۰	غیرنظارتی	-	گراف وخوشه-بندی	• حفظ ساختار سراسری یا محلی	• پیچیدگی زمانی بالا • عدم مقیاس‌پذیری
NLE-SLFS [34]	۲۰۱۹	غیرنظارتی	-	قسمت-بندی	• سادگی بیشتر به دلیل بهره‌مندی از شبه‌برچسب‌ها • در نظر گرفتن ارتباط ویژگی‌ها با شبه‌برچسب	• پیچیدگی بالای محاسباتی و زمانی • در نظر نگرفتن ارتباط ویژگی‌ها باهم
NSSLFS [17]	۲۰۱۹	غیرنظارتی	-	قسمت-بندی	• مقاوم دربرابر نویز و دور افتاده • حفظ ساختار سراسری داده‌ها	• در نظر نگرفتن ساختار محلی و ارتبط داده‌ها با همسایه‌های خود
RSR [37]	۲۰۱۵	غیرنظارتی	-	رگرسیون	• ایجاد توانایی بازسازی ویژگی‌های از دست رفته	• نادیده گرفتن ساختار سراسری و محلی • عدم مقیاس‌پذیری
RUFS [14]	۲۰۱۹	غیرنظارتی	LLE	گراف	• حفظ ساختار محلی و سراسری • مقاوم نسبت به نویز	• عدم مقیاس‌پذیری
SLSDR [32]	۲۰۱۹	غیرنظارتی	SGFS	گراف	• حفظ ساختار هندسی هم داده و هم ویژگی • مقاوم در برابر نویز و داده‌ی پرت	• عدم مقیاس‌پذیری • دقت کم‌تر به دلیل نداشتن راهنمایی برچسب‌ها
WFAACOFS [35]	۲۰۱۹	غیرنظارتی	ACO	جست و جو	• سادگی پیاده‌سازی • بهبود سنجش سودمندی یک ویژگی با راهنمایی میزان مطلوبیت آن	• گرفتارشدن در بهینه‌ی محلی
[23]	۲۰۱۹	غیرنظارتی	-	گراف	• حفظ ساختار محلی • آزاد از پارامتر	• عدم مقیاس‌پذیری • در نظر نگرفتن ساختار سراسری
[26]	۲۰۱۹	غیرنظارتی	-	گراف	• حفظ چند ساختار به طور هم‌زمان • دقت بالا	• عدم مقیاس‌پذیری
[4]	۲۰۱۹	غیرنظارتی	-	گراف	• پیچیدگی محاسباتی کم‌تر به دلیل ایجاد گراف کوچک‌تر از گراف اصلی	• عدم مقیاس‌پذیری
DGSPSFS [31]	۲۰۲۰	نظارتی	-	گراف	• بهبود دقت انتخاب ویژگی با حفظ دو ساختار سراسری به طور هم‌زمان • ماتریس پاسخ دارای اطلاعات بیشتری است.	• عدم مقیاس‌پذیری • مصرف حافظه‌ی بالا
RFS [21]	۲۰۱۰	نظارتی	-	رگرسیون و ماتریس	• از بین بردن داده‌ی نویزی	• به کارنبردن یادگیری زیرفضا • نادیده گرفتن اطلاعات به دلیل استفاده از بردار • عدم مقیاس‌پذیری
[18]	۲۰۱۹	نظارتی	-	رگرسیون	• در نظر گرفتن ارتباط بین ویژگی‌ها و داده‌ها • بهبود کارایی روش‌های یادگیری	• از دست رفتن برخی اطلاعات به دلیل محاسبات خطی و برداری • وابستگی به برچسب داده‌ها

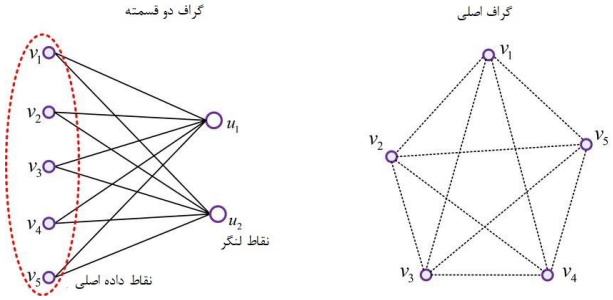
[20]	۲۰۱۹	نظارتی	-	قسمت-بندی	• در نظر گرفتن ارتباط بین ویژگی‌ها • بهبود کارایی الگوریتم‌های یادگیری با در نظر گرفتن همسایه‌ی داده‌ها	• وابستگی به برچسب داده‌ها • مصرف حافظه‌ی بالا
[36]	۲۰۱۹	نظارتی	-	رگرسیون	• حفظ اطلاعات داده‌ها	• وابستگی به برچسب داده‌ها
MASFS [16]	۲۰۱۹	نیمه-نظارتی	SSFS	گراف	• انطباقی بودن • استفاده از اطلاعات نماهای مختلف	• عدم کارایی • وابستگی به برچسب داده‌ها
[29]	۲۰۱۹	نیمه-نظارتی	-	گراف	• حفظ ساختار محلی • حفظ ساختار سراسری	• کارایی کم به دلیل نیاز به تعدادی برچسب • عدم مقیاس‌پذیری

به عبارتی منظور از گراف دوقسمته، گرافی است که ارتباط بین نمونه-ها و نقاط لنگر را نشان می‌دهد. این گراف که شامل n نقطه‌ی داده و m نقطه‌ی لنگر است، در محاسبات به جای گراف مجموعه‌ی داده‌ی اصلی، به کار گرفته می‌شود و هزینه‌ی محاسبات را کاهش می‌دهد. به این ترتیب ماتریس B نیز وزن یال‌های گراف دوقسمته را نشان می-دهد. در شکل(۱) مثالی از گراف دو قسمته و گراف مجموعه‌ی داده‌ی اصلی نشان داده شده‌است.

تئوری گراف دو قسمته را می‌توان برای وظایف مختلفی چون خوشه-بندی، انتخاب ویژگی و غیره، به کار گرفت. Hu و همکاران [۳۸]، مدلی ارائه کرده‌اند، که در آن با کمک k-means یا به صورت تصادفی، نقاط لنگر به دست آورده می‌شوند، و سپس با کمک ماتریس داده‌ی $X \in \mathbb{R}^{n*d}$ و نقاط لنگر $P \in \mathbb{R}^{m*d}$، گراف دو قسمته به صورتی که در رابطه‌ی(۱) آمده است، محاسبه می‌شود.

$$(۱)\quad b_{ij} = \begin{cases} \dfrac{dis(i,r+1) - dis(i,j)}{r.dis(i,r+1) - \sum_{j=1}^{r} dis(i,j)} & j \le r \\ 0 & j > r \end{cases}$$

که تنها rتا از نزدیک‌ترین نقاط لنگر به هر نقطه‌ی داده‌ی اصلی متصل هستند و dis(i,j)، فاصله‌ی اقلیدسی بین iامین نمونه‌ی x_i و jامین نقطه‌ی لنگر p_j را نشان می‌دهد. علاوه‌براین، این فاصله‌ها برای هر نقطه‌ی داده، از کوچک به بزرگ مرتب شده‌اند. به عبارتی برای هر i، $dis(i,1) \le dis(i,2) \le \ldots\ldots \le dis(i,m)$ برقرار است.



شکل(۱): مثالی از گراف دو قسمته و گراف اصلی [۳۹]

۳-۳- محدودیت رتبه پایین

ماتریس وزنی M تنها با در نظر گرفتن اطلاعات به دست آمده از داده-ها ساختار محلی آن‌ها را مشخص می‌کند. این امر سبب می‌شود تا در ساخت فضای محلی داده‌ها دچار خطا شود. به همین دلیل ماتریس وزن بازسازی M^{ra} برای هر ویژگی محاسبه می‌شود، تا با محاسبه‌ی اختلاف این دو ماتریس، قابلیت حفظ گراف توسط هر ویژگی به دست آید. بر این اساس، ساختار محلی داده‌ها با کمک اطلاعات نمونه‌ها و ویژگی‌ها به دست می‌آید[۴۰]. به این ترتیب در اکثر روش‌های براساس گراف که تاکنون ارائه شده، هنگام به دست آوردن ماتریس وزن بازسازی[۱۹] M^{ra}، ساختار محلی داده‌ها، با در نظر گرفتن همسایه-های هر نقطه‌ی داده، حفظ می‌شود. درحالیکه ساختار سراسری داده-ها، می‌تواند اطلاعات تکمیل کننده‌ای ارائه کند، که در ارتباط محلی داده‌ها از دست می‌رود. همچنین، استفاده از این ساختار می‌تواند به شکل قابل توجهی کارکرد انتخاب ویژگی را بهبود بخشد. علاوه‌براین، توجه به ساختار سراسری داده‌ها، می‌تواند تأثیر داده‌های نویزی و دورافتاده را در انتخاب ویژگی کاهش دهد. بنابراین با اضافه کردن محدودیت رتبه پایین، نه تنها ارتباط محلی داده‌ها بلکه ارتباط سراسری آن‌ها نیز در نظر گرفته می‌شود [۴۱].

به این ترتیب، برای حفظ ساختار سراسری داده‌ها، وزن بازسازی M^{ra} به صورت رابطه‌ی(۲) محاسبه می‌شود.

$$\min_{M^{ra}} \| X - XM^{ra} \|_2 + \| M^{ra} \|_2 \qquad (۲)$$

که در آن X، ماتریس داده‌ی اصلی می‌باشد. سپس، محدودیت رتبه پایین روی M^{ra} را می‌توان به صورت دو ماتریس درجه ra، به صورت رابطه‌ی (۳) به دست آورد.

$$M^{ra}=AB \qquad (۳)$$

که $A \in \mathbb{R}^{d*ra}$ و $B \in \mathbb{R}^{ra*r}$ می‌باشند. سپس محاسبه‌ی وزن بازسازی به صورت رابطه‌ی(۴) تغییر می‌کند.

$$\min_{A,B} \| X - XAB \|_2 + \| AB \|_2 \qquad (۴)$$

در رابطه‌ی(۴)، ماتریس کاهش‌یافته‌ی $XA \in \mathbb{R}^{n*r}$ در B ضرب می‌شود تا ماتریس ویژگی‌های X را ارائه کند. از نظر هندسی وقتی A (یا B) ضرب در X (یا XA) می‌شوند، X (یا XA) را به یک فضای

جدید تبدیل می‌کنند. به عبارتی یادگیری زیرفضا را با در نظر گرفتن همبستگی بین d ویژگی (یعنی همه‌ی ویژگی‌ها به عنوان یک گروه) انجام می‌دهد، که همبستگی سراسری ویژگی‌ها نامیده می‌شود. بنابراین محدودیت رتبه پایین روی A (یا B) با در نظر گرفتن همبستگی سراسری ویژگی‌ها، به صورت یادگیری زیرفضا عمل می‌کند. همچنین چنین یادگیری زیرفضایی، از طریق در نظر گرفتن همبستگی سراسری ویژگی‌ها، منجر به حفظ ساختار سراسری نمونه‌ها می شود [۴۲].

به این ترتیب در رابطه‌ی(۴)، با اجرای یادگیری زیرفضا، ارتباط سراسری داده‌ها و ویژگی‌ها در نظر گرفته می‌شود [۱۴]. بنابراین نه تنها ساختار محلی داده‌ها و ویژگی‌ها، که ساختار سراسری آن‌ها نیز حفظ می‌شود و در نتیجه کارایی انتخاب ویژگی بهبودیافته و تأثیر نویز و دورافتاده کاهش می‌یابد.

۴- روش پیشنهادشده

انتخاب ویژگی، از مراحل پیش‌پردازش داده‌ها می‌باشد، که به کمک آن، ویژگی‌های نامرتبط و اضافی حذف می‌شوند و به این ترتیب پردازش داده‌ها، با سرعت و دقت بالاتری همراه خواهد بود. روش پیشنهادشده‌ی این مقاله، انتخاب ویژگی را براساس گراف انجام می‌دهد. روش‌های براساس گراف، به دلیل دقت و کارایی بالایی که دارند، از محبوبیت خاصی برخوردارند. اما از چالش‌های مهم روش‌های انتخاب ویژگی براساس گراف، مقیاس‌پذیری و پویایی آن‌ها می‌باشد. چالش پویایی، چالشی است که در [۱۴] مورد بررسی قرار گرفته‌است و در آن ساخت گراف ماتریس و ماتریس وزنی به صورت پویا و در یک حلقه‌ی تکرار، انجام می‌شود. همچنین این روش، ساختار سراسری داده‌ها را نیز، که در بردارنده‌ی اطلاعات مکملی از داده‌ها هستند، در نظر می‌گیرد. اما، چالش مقیاس‌پذیری در روش ارائه‌شده در این مقاله، مورد توجه قرار نگرفته است. بنابراین، در روش پیشنهادی قصد داریم، تا با افزودن ویژگی مقیاس‌پذیری در [۱۴] عملکرد آن را بهبود بخشیم. به این ترتیب با کمک تئوری گراف دوقسمته و اعمال آن در روش ارائه‌شده در [۱۴]، مقیاس‌پذیری این روش افزایش می‌یابد. مقیاس‌پذیری موجب می‌شود تا پیچیدگی زمانی الگوریتم ارائه‌شده، تا حد قابل توجهی کاهش یابد. این مسئله، روش را برای داده‌های با مقیاس بالا بسیار کارآمدتر خواهد کرد. علاوه‌براین در روش پیشنهادی، برای حفظ ساختار سراسری داده‌ها، ماتریس وزن بازسازی، به کار گرفته می‌شود. در روش پیشنهادی، ماتریس وزن بازسازی از طریق محدودیت رتبه پایین که تحت نظارت ماتریس وزنی بدست آمده، ساخته می‌شود. به این ترتیب موجب حفظ ساختار سراسری داده‌ها می‌شود. این امر سبب می‌شود تا اطلاعات سراسری و محلی داده‌ها به طور هم‌زمان در نظر گرفته‌شود.

به این ترتیب، به طور کلی در روش پیشنهاد شده، ۴ هدف دنبال می‌شود. حفظ ساختار سراسری و محلی داده‌ها، که هر کدام به خودی خود اطلاعات به خصوصی از داده‌ها را دارا می‌باشند و در نظر نگرفتن هر کدام از این دو، باعث از دست رفتن بخشی از اطلاعات به دست آمده از داده‌ها می‌شود دو هدف دنبال شده در روش پیشنهادی است. علاوه‌براین دو، در روش پیشنهاد شده، پویایی و مقیاس‌پذیری نیز در نظر گرفته شده‌است. در اکثر روش‌های براساس گراف، گراف ماتریس به صورت ایستا از قبل ساخته می‌شود، درحالیکه در این مقاله، گراف ماتریس به همراه ماتریس وزنی به صورت پویا ساخته می‌شوند. همچنین مقیاس‌پذیری این روش باعث می‌شود تا از لحاظ پیچیدگی زمانی و محاسبات، عملکرد بهتری نسبت به سایر روش‌های انتخاب ویژگی داشته‌باشد. در ادامه به شرح نحوه‌ی اجرای این روش می‌پردازیم.

در کاربرد واقعی، داده‌ی اصلی اغلب دارای داده‌های نویزی و دورافتاده است. بنابراین، گراف ماتریس ساخته‌شده، ممکن است نادرست باشد. برای حل این مسئله، گراف ماتریس از داده‌های تمیز با ابعاد کم، که داده‌ی نویزی و دورافتاده‌ی کمتری دارد، به دست می‌آید. با این حال، نه ماتریس گراف و نه داده‌ی کم‌بعد، از قبل مشخص نیستند. به همین خاطر ماتریس گراف به همراه ماتریس وزنی به صورت مکرر بهینه‌سازی می‌شوند، تا بهترین نتیجه برای هر کدام به دست آید [۱۴]. به این ترتیب، در یک حلقه‌ی تکرار و به صورت پویا، گراف ماتریس S و ماتریس وزنی M محاسبه می‌شوند. این حلقه تا زمانی که بهترین مقدار S و M از رابطه‌ی(۵) به دست آید، تکرار می‌شود.

$$\min_{S,M} \frac{1}{2}\sum_{i,j}^{n} \| x_i - x_j M_i \|_2 \;\; s_{ij} + a\sum_{i,j}^{n} s_{(i,j)}^2 \qquad (۵)$$

که x_i، نمونه‌ی i از مجموعه‌ی داده‌ی اصلی $X \in \mathbb{R}^{n*d}$ ، M_i، بردار سطر i از ماتریس وزنی $M \in \mathbb{R}^{d*d}$، $s_{i,j}$، مقدار شباهت بین نمونه‌ی i و j از گراف ماتریس $S \in \mathbb{R}^{n*n}$ و aپارامتر تنظیم است. به این ترتیب S و M در این حلقه به روز رسانی می‌شوند، که برای به روز رسانی M از رابطه‌ی(۶) استفاده می شود.

$$M_i = (X^{i^T} X^i)^{-1} X^{i^T} x_i \qquad (۶)$$

که $X^i \in \mathbb{R}^{k*d}$ ، مجموعه‌ی همسایه‌های x_i را نشان می‌دهد. برای مشخص کردن X^i، معیار شباهت بین زوجین نقاط داده در مجموعه‌ی داده‌ی اصلی محاسبه می‌شود. سپس k نقطه‌ای که بیشترین شباهت را به داده‌ی x_i دارند، به عنوان همسایگان x_i انتخاب می‌شوند. این معیار شباهت می‌تواند فاصله‌ی اقلیدسی بین زوج نمونه‌ها باشد. به این ترتیب، با در نظر گرفتن همسایه‌های هر نقطه‌ی داده، از تأثیر داده‌های دور افتاده، ممانعت می‌شود. زیرا داده‌های دورافتاده، اغلب بسیار دورتر از k نزدیک‌ترین همسایه‌ی هر نقطه‌ی داده می‌باشند و علاوه‌بر این، ساختار محلی داده‌ها نیز حفظ می‌شود.

پس از این، S به جای روش ارائه‌شده در [۱۴]، با استفاده از تئوری گراف دو قسمته [۴]، به روز رسانی می‌شود و در نتیجه، مقیاس‌پذیری این روش فراهم می‌شود. ماتریس گراف S، ارتباط نمونه‌ها در مجموعه‌ی داده را نشان می‌دهد. به عبارتی داده‌ها در ماتریس گراف S نگاشت

می‌شوند. بر این اساس، ماتریس گراف S، با رابطه‌ی(۷) محاسبه می‌شود.

$$S = B\Delta^{-1}B^{T} \quad (۷)$$

که گراف دو قسمته‌ی B با رابطه(۱)، به دست می‌آید و $\Delta \in \mathbb{R}^{m*m}$ یک ماتریس قطری است و با $\Delta_{jj} = \sum_{i=1}^{n} b_{ij}$ محاسبه می‌شود. بنابراین رابطه‌ی(۵) به رابطه‌ی(۸) تغییر می‌کند. تا بهترین مقدار برای گراف دو قسمته و در نهایت ماتریس گراف به دست آید.

$$\min_{B,M} \frac{1}{2}\sum_{i,j}^{n} \| x_i - x_j M_i \|_2 \ (B\Delta^{-1}B^{T})_{i,j} + a\sum_{i,j}^{n} (B\Delta^{-1}B^{T})_{i,j}^{2} \quad (۸)$$

رابطه‌ی(۸) برعکس رابطه‌ی(۵) برای حل معادله‌ی کمینه، نیازی به ماتریس گراف S ندارد و با کمک گراف دو قسمته‌ی B که محاسبه‌ی آن بسیار کم‌هزینه‌تر از ماتریس گراف S است، این کار را انجام می‌دهد. همانطور که در رابطه‌ی(۱) نشان داده شده‌است، ماتریس B با کمک نقاط لنگر محاسبه می‌شود. بنابراین در تابع بهینه‌سازی در هر مرحله، انتخاب این نقاط لنگر حائز اهمیت است و در محاسبه‌ی B تأثیرگذار می‌باشند.

همانطور که گفته شد، ارتباط سراسری داده‌ها می‌تواند اطلاعات تکمیل کننده‌ای ارائه کند، که در ارتباط محلی داده‌ها از دست می‌رود. بنابراین با رابطه‌ی(۲)، محدودیت رتبه پایین روی ماتریس وزن بازسازی M^{ra} اعمال می‌شود و ارتباط سراسری داده‌ها در نظر گرفته می‌شود. پس از محاسبه‌ی ماتریس وزن بازسازی M^{ra}، نمره‌ی هر ویژگی با رابطه‌ی(۹)، محاسبه می‌شود. لازم به ذکر است که ویژگی‌ها در اصل ستون‌های ماتریس داده‌ی X می‌باشند و تعداد آن‌ها برابر با d است. به این ترتیب اندازه‌ی ماتریس‌های M و M^{ra} برابر با (d×d) می‌باشد.

$$score_i = \|M_i - M^{ra}_i\|_F \quad (۹)$$

نمره‌ی هر ویژگی، قابلیت حفظ گراف را برای هر ویژگی نشان می‌دهد، که ویژگی با کم‌ترین نمره، بهترین ویژگی می‌باشد. به این ترتیب، ویژگی‌ها بر اساس نمره، در یک لیست به صورت صعودی مرتب می‌شوند و در نهایت این روش یک لیست مرتب شده از ویژگی‌ها را برمی‌گرداند. در الگوریتم(۱) مراحل اجرای روش پیشنهاد شده، نشان داده می‌شود.

استفاده از این تئوری موجب می‌شود تا هزینه‌ی محاسباتی گراف ماتریس تا O(nm) ، که m≤n است، کاهش یابد. به این ترتیب با کمک گراف دو قسمته‌ی $B \in \mathbb{R}^{n*m}$، زمان محاسبه‌ی S، تا حد قابل توجهی کاهش می‌یابد، که این مسئله، روش پیشنهادشده را برای داده‌های با حجم بالا و پرهزینه، بسیار مفید و کارآمد می‌کند. زیرا اغلب، روش‌های انتخاب ویژگی برای داده‌هایی استفاده می‌شوند که تعداد نمونه‌ها بسیار زیاد است. بنابراین، هزینه‌های محاسباتی گراف ماتریس برای این چنین داده‌هایی، کارایی روش‌های انتخاب ویژگی را بسیار کاهش می‌دهد. به همین دلیل این روش به علت داشتن ویژگی مقیاس‌پذیری، عملکرد بسیار کارآمدتری دارا می‌باشد.

ورودی: ماتریس داده‌ی $X \in \mathbb{R}^{n*d}$
خروجی: لیست مرتب‌شده از ویژگی‌ها

۱-محاسبه‌ی S و M:
۱-۱-محاسبه‌ی Kنزدیک‌ترین همسایه برای هر نمونه
۱-۲-t=0
۱-۳-تکرار:
۱-۳-۱-به‌روز رسانی M با رابطه‌ی(۶)
۱-۳-۲-به‌روزرسانی S :
۱-۲-۳-۱-محاسبه‌ی B با رابطه(۱)
۱-۲-۳-۲-محاسبه‌ی S با رابطه‌ی(۷) با استفاده از گراف دوقسمته‌یB
۱-۳-۳-t=t+1
تا زمانیکه مقدار به دست آمده در رابطه‌ی(۸)، در دو تکرار مکرر، کمتر از 10^{-5} باشد و یا حلقه تا یک حد آستانه‌ای تکرار شود.
۲-محاسبه‌ی ماتریس وزن بازسازی M^{ra} با رابطه‌ی(۲)
۳-محاسبه‌ی نمره‌ی هر ویژگی با رابطه‌ی (۹)
۴-مرتب‌سازی صعودی ویژگی‌ها براساس نمره‌ی هر ویژگی
۵-بازگرداندن لیست مرتب‌شده‌ی ویژگی‌ها

الگوریتم (۱): روش پیشنهادشده

۵- آزمایشات

در این بخش، روش پیشنهادشده با ۷ روش انتخاب ویژگی LLES [۳۰] ، RSR [۳۷] ، MCFS [۳۳] ، RFS [۲۱] ، LS [۲۷] ، RUFS [۱۴] و [۲۸] از نظر کارایی کلاس‌بندی و پیچیدگی زمانی مقایسه می‌شود. همچنین میزان حساسیت به پارامتر این روش، مورد ارزیابی و بررسی قرار می‌گیرد.

۵-۱- مجموعه‌های داده

آزمایشات این مقاله روی ۶ مجموعه‌ی داده، که در دسترس عموم می‌باشند، انجام شده‌است. یک مجموعه‌ی داده با نام Wine از مخزن یادگیری ماشین UCI[۲۰] دانلود شده‌است. ۵ مجموعه داده‌ی دیگر نیز از سایت مجموعه‌های داده‌ی انتخاب ویژگی[۲۱] دانلود شده‌اند، که شامل Yale، Colon، WrapAR10p، ORL و GLOMA می‌باشند. جزئیات این مجموعه‌های داده در جدول(۲) نشان داده شده‌است.

جدول (۲): معرفی مجموعه‌های داده

مجموعه داده	تعداد نمونه‌ها	تعداد ویژگی‌ها	تعداد کلاس‌ها
Wine	۱۷۸	۱۳	۳
Yale	۱۶۵	۱۰۲۴	۱۵
Colon	۶۲	۲۰۰۰	۲
WrapAR10p	۱۳۰	۲۴۰۰	۱۰
ORL	۴۰۰	۱۰۲۴	۴۰
GLOMA	۵۰	۴۴۳۴	۴

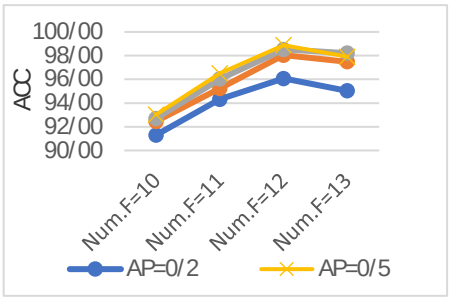
مجموعه‌های داده‌ی استفاده‌شده در آزمایشات، انواع مختلفی از اطلاعات را در بردارند. برای مثال Colon و GLOMA شامل اطلاعات

بیولوژیکی هستند. مجموعه‌های داده‌ی Yale ، WrapAR10p و ORL داده‌های مربوط به تصاویر صورت را نگه می‌دارند و Wine داده‌های علمی فیزیکی را دارا می‌باشد. به طور کلی ویژگی‌ها، خصوصیات و اطلاعات مربوط به هر نمونه را نشان می‌دهند. برای مثال در مجموعه‌ی داده‌ی Wine برخی ویژگی‌ها منیزیم، شدت رنگ، اسید مالیک و غیره می‌باشند، که خصوصیات فیزیکی نمونه‌های نوشیدنی را مشخص می‌کنند. همچنین هر یک از نمونه‌ها در این مجموعه‌های داده در یک کلاس یا دسته طبقه‌بندی می‌شوند، که با برچسب کلاس خود مشخص می‌شوند. از بین مجموعه‌های داده‌ی معرفی‌شده، Colon یک مجموعه‌ی داده‌ی دو کلاسه است و باقی، مجموعه داده‌های چند کلاسه می‌باشند. برای مثال در مجموعه‌ی داده‌ی Colon دو برچسب سرطان داشتن یا نداشتن وجود دارد، که نشان می‌دهد هر نمونه در کدام کلاس قرار دارد. تعداد ویژگی‌های مجموعه‌های داده‌ی به کار گرفته‌شده در آزمایشات، از ۱۳ تا ۴۴۳۴ متغیر است و تعداد نمونه‌ها بین ۶۲-۴۰۰ می‌باشند. در اکثر مجموعه‌های داده‌ی استفاده‌شده، تعداد ویژگی‌ها بسیار بیشتر از تعداد نمونه‌هاست، که این امر انتخاب ویژگی را دشوارتر می‌کند.

۵-۲- تنظیمات آزمایشگاهی

در این مقاله، آزمایشات با استفاده از MATLAB R2018a روی کامپیوتر با Corei7، 2.8 GHz و حافظه‌ی رم 8GB اجرا شده‌است. با انجام آزمایشات مختلف، بهترین مقدار برای پارامتر تعداد همسایگان به دست می‌آید. به همین ترتیب بهترین مقدار برای پارامتر r در رابطه‌ی (۱)، که یک عدد صحیح است، به دست می‌آید، که در این آزمایشات برابر با ۴ تنظیم می‌شود. لازم به ذکر است که روش پیشنهادی این مقاله، براساس نقاط لنگر عمل می‌کند. بنابراین برای دستیابی به کارایی بهتر روش انتخاب ویژگی پیشنهادی، لازم است تا تعداد مناسب نقاط لنگر مشخص شود. تعداد نقاط لنگر براساس درصدی از تعداد نقاط داده‌ی اصلی می‌باشد. به عبارتی m=n*AP است، که m تعداد نقاط لنگر، n تعداد نقاط داده اصلی و AP نسبت تعداد نقاط لنگر به نقاط داده را نشان می‌دهد [۴]. برای به دست آوردن بهترین مقدار AP، آن را در بازه‌ی {۰,۲،۰,۳،۰,۴،۰,۵} تغییر می‌دهیم و دقت کلاس‌بندی ACC را برای آن، با تغییر تعداد ویژگی‌های انتخاب‌شده در بازه‌ی {۱۰،۱۱،۱۲} برای مجموعه‌ی داده‌ی Wine به دست می‌آوریم. به طور واضح هر چه تعداد نقاط لنگر بیشتر باشد، نتایج بهتر است. اما شکل(۲) نشان می‌دهد که منحنی AP=0.3 و AP=0.4 بسیار بهم نزدیک هستند. به این ترتیب با رعایت انصاف، برای تمام روش‌های مورد مطالعه و روش پیشنهادی AP=0.2 تنظیم می‌شود.

برای انجام وظیفه‌ی کلاس‌بندی نیز از ماشین بردار پشتیبان[۲۲](SVM) استفاده شده‌است. علاوه‌براین برای به دست‌آوردن نتایج با ثبات بیشتر، آزمایشات روی همه‌ی مجموعه‌های داده برای همه‌ی الگوریتم‌ها، ۱۰ بار تکرار شده‌اند. برای مشخص‌کردن داده‌های تربیت‌شده[۲۳] از k-fold-cross-validation استفاده‌شده‌است. روش پیشنهادشده از طریق دقت کلاس‌بندی برای ارزیابی کارایی کلاس‌بندی، با روش‌های دیگر مقایسه می‌شود.



شکل(۲): تعداد نقاط لنگر در مجموعه داده Wine

دقت کلاس‌بندی(ACC) به صورت رابطه‌ی(۱۰) تعریف‌شده‌است [۱۴].

$$ACC=N_{correct}/N \quad (۱۰)$$

که N تعداد نمونه‌های مجموعه‌ی داده و $N_{correct}$ تعداد نمونه‌هایی که به درستی کلاس‌بندی شده‌اند را نشان می‌دهد. ACC بیشتر، کارایی بالاتر را نشان می‌دهد.

۵-۳- نتایج آزمایشات بررسی دقت کلاس‌بندی

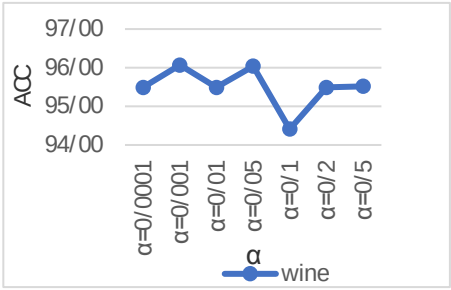
میانگین دقت کلاس‌بندی(mACC) برای ۱۰ بار اجرای الگوریتم پیشنهادشده و روش‌های معرفی‌شده، با به کارگیری SVM، در جدول(۳) نشان داده شده‌است.

همانطور که در جدول(۳) نشان داده‌شده‌است، روش پیشنهادی در mACC که بعد از ۱۰ بار اجرا برای هر مجموعه‌ی داده، محاسبه شده‌است، با سایر روش‌ها مقایسه شده‌است. روش پیشنهادی، در دو مجموعه‌ی داده‌ی WarpAR10p و ORL، که از مجموعه داده‌های پردازش تصویر هستند، به دلیل شباهت خیلی زیاد بین نقاط داده و توزیع یکنواخت آن‌ها، در مقایسه با اکثر روش‌های مورد بررسی بهبود یافته‌است. همچنین این روش، در چهار مجموعه‌ی داده‌ی Wine، Yale، Colon و GLOMA از همه‌ی روش‌های مورد مقایسه بهتر عمل کرده‌است. برای مثال میانگین دقت کلاس‌بندی روش پیشنهادی در مقایسه با روش LLES به دلیل در نظر گرفتن ساختار سراسری داده‌ها و همچنین افزایش مقیاس‌پذیری، به طور متوسط ۲,۶٪ بیشتر است. همچنین در مقایسه با RSR که از ویژگی خودنمایندگی استفاده می‌کند، به طور میانگین ۳,۸۴٪ بهبود پیدا کرده‌است. در مقایسه با MCFS نیز، از آنجاییکه این روش یا ساختار سراسری یا ساختار محلی را حفظ می‌کند، ۴,۷۴٪ بهتر عمل کرده‌است. در مقایسه با RFS و LS نیز که یادگیری زیرفضا را در مدل انتخاب ویژگی خود به کار نبرده‌اند، به ترتیب ۳,۷۷٪ و ۱,۷۶٪ بیشتر است. علاوه‌براین، دقت کلاس‌بندی روش پیشنهادی در مقایسه با روش RUFS، به طور متوسط ۰,۳۱٪ بهبود یافته‌است. زیرا علاوه‌بر حفظ ساختار محلی و سراسری، با اعمال تئوری گراف دوقسمته در روش پیشنهادی، خاصیت مقیاس‌پذیری این

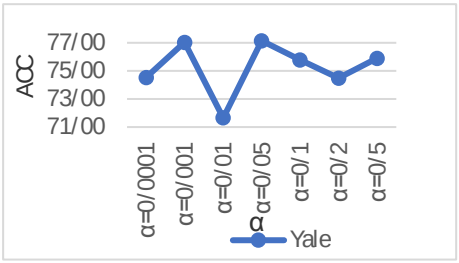
روش تقویت یافته‌است. علاوه‌بر این در مقایسه با روش [۲۸] نیز، به طور متوسط ۰,۲۸٪ به دلیل در نظر گرفتن ارتباط سراسری و محلی بین نمونه‌ها ارتقا یافته‌است.

۵-۴- آنالیز حساسیت به پارامتر

در روش پیشنهادشده پارامتر مهم α در رابطه‌ی(۵) وجود دارد. این پارامتر، چگونگی حفظ ساختار محلی نمونه‌ها را توسط تابع بازسازی، کنترل می‌کند. برای سنجش میزان حساسیت روش پیشنهادشده به پارامتر α در تابع هدف و بررسی اینکه این پارامتر تا چه اندازه روی دقت کلاس‌بندی ACC تأثیر می‌گذارد، مقدار آن در بازه‌ی $\{10^{-4},10^{-3},10^{-2},0.05,0.1,0.2,0.5\}$ تغییر داده می‌شود. شکل(۳) و (۴) مقدار ACC را برای مقادیر مختلف پارامتر α از روش پیشنهادی، روی دو مجموعه‌ی داده‌ی نمونه‌ی Wine و Yale نشان می‌دهد. از این دو شکل می‌توان دریافت که تابع هدف پیشنهادی به تنظیم پارامتر حساس است، یعنی با دادن مقادیر مختلف به پارامتر α، دقت کلاس‌بندی متفاوتی نتیجه می‌شود. همانطور که در این دو شکل دیده می‌شود، بهترین کارایی مجموعه‌ی داده‌ی Wine و Yale وقتی حاصل می‌شود که α به ترتیب 10^{-3} و 0.05 باشد.



شکل(۳): حساسیت دقت کلاس‌بندی ACC مجموعه داده Wine به پارامتر α



شکل(۴): حساسیت دقت کلاس‌بندی ACC مجموعه داده Yale به پارامتر α

برای دستیابی به بهترین مقدار پارامتر، در روش پیشنهادی و روش‌های مورد آزمایش، از یک روش جست‌وجوی شبکه‌ای[۲۴] در بازه‌ی $\{10^{-4},10^{-3},10^{-2},0.05,0.1,0.2,0.5\}$ استفاده می‌شود. به این صورت که با توجه به بازه‌ی تعریف‌شده، پارامترها به صورت تصادفی مقداردهی می شوند. علاوه‌براین برای متدهای با بیش از یک پارامتر، مجموعه‌ی ترکیبات مقادیر پارامترها مشخص می شود. سپس تابع نمره، که در اینجا دقت کلاس‌بندی ACC می‌باشد، محاسبه می شود. به این ترتیب مقدار پارامتر مناسب برای مجموعه‌های داده مورد آزمایش، با رعایت انصاف به دست می‌آید.

۵-۵- آنالیز پیچیدگی زمانی

درRUFS در هر تکرار، هزینه زمان اجرای الگوریتم روی محاسبه‌ی $(X^{i^T}X^i)^{-1}X^{i^T}x_i$ در رابطه‌ی(۶) و $f_{i,j}$ برای محاسبه‌ی S در رابطه‌ی $S_{i,j}=(-\frac{1}{4\alpha}f_{i,j}+\tau)$ و $f_{i,j}=\|x_i-x_jM\|_2^2$ تمرکز دارد، که پیچیدگی زمانی متناظر آن‌ها $O(n^2d)$ و $O(nd^2)$ می‌باشد، که n تعداد نمونه‌ها و d تعداد ویژگی‌ها می‌باشند. اما در روش پیشنهادی این مقاله به جای محاسبه‌ی $f_{i,j}$ از گراف دو قسمته(B) که از رابطه‌ی(۱) محاسبه می‌شود، استفاده می‌شود که پیچیدگی زمانی و محاسباتی بسیار کمتری نسبت به گراف S دارد. بنابراین به جای $O(nd^2)$، پیچیدگی به $O(nm)$ کاهش می‌یابد، که در آن m تعداد نقاط لنگر است و تعداد نقاط لنگر بسیار کمتر از نقاط اصلی مجموعه‌ی داده می‌باشد ($m\leq n$). به این ترتیب پیچیدگی زمانی الگوریتم پیشنهادی $\max\{O(n^2d),O(nm)\}$ خواهد بود. این امر سبب می‌شود تا روش پیشنهادی برای داده‌های بزرگ با صرف هزینه‌ی محاسباتی کمتری عمل کند و کارایی آن را بهبود بخشد. پیچیدگی زمانی روش LLES نیز، $O(n^2d+ndk^3)$ است، که k در آن تعداد همسایه‌های نقاط داده می‌باشد. در روش RSR نیز پیچیدگی $O(T(d^3+d^2n)$ است که T مقدار پارامتر تکرار را نشان می‌دهد. پیچیدگی زمانی روش MCFS نیز $O(n^2d+km^3+nkm^2+mlogm)$ تعیین شده‌است، که k تعداد خوشه‌ها و m تعداد ویژگی‌های انتخاب شده‌است. پیچیدگی RFS برابر با $O(T(n^2+nd+nc+d^2)$ است و T تعداد تکرار، c نیز تعداد کلاس‌ها را نشان می‌دهد. پیچیدگی زمانی [۲۸] نیز $O(n^2k^2d^2+k^2d^2)$ می‌باشد که در آن k تعداد ویژگی‌های انتخاب‌شده می‌باشد. در نهایت پیچیدگی زمانی روش LS، $O(n^2d+nd+d)$ می‌باشد. بنابراین مطالب گفته‌شده نشان می‌دهد که روش پیشنهادی این مقاله در بهبود پیچیدگی زمانی روش انتخاب ویژگی، کارآمد عمل کرده‌است.

بر این اساس، زمان اجرای الگوریتم پیشنهادی و الگوریتم RUFS که مرجع اصلی مورد مقایسه می‌باشد، محاسبه شده‌است. زمان اجرای الگوریتم پیشنهادی برای نمونه، بر روی مجموعه داده‌های Wine، Yale و Colon به ترتیب برابر با ۱۴(s)، ۹۴۴۹۰(s) و ۹۶۷۹۵(s) می‌باشد. در مقایسه با آن، زمان اجرای الگوریتم RUFS نیز بر روی این مجموعه‌های داده به ترتیب ۲۰(s)، ۱۰۳۲۲۷(s) و ۱۰۷۳۴۶(s) می‌باشد. نتایج نشان می‌دهد که زمان اجرای الگوریتم پیشنهادی، در مقایسه با الگوریتم RUFS بر روی مجموعه داده‌های اشاره‌شده، به ترتیب ۶(s)، ۸۷۳۶(s) و ۱۰۵۵۱(s) کاهش یافته‌است؛ که به این ترتیب، کارآمدی روش پیشنهادی در بهبود زمان اجرا را نشان می‌دهد.

۶- نتیجه‌گیری

در این مقاله یک روش جدید انتخاب ویژگی غیرنظارتی براساس گراف ارائه شده است. این رویکرد به عنوان یک رویکرد بهبودیافته از روش ارائه‌شده در [۱۴]، با حفظ ساختار محلی و ویژگی پویایی، به جای بدست آوردن ماتریس وزنی با کمک تابع لاگرانژ مانند بسیاری از روش‌های موجود، از تئوری گراف دو قسمته بهره می‌برد. این امر سبب می‌شود تا روش پیشنهادی در مواجهه با داده‌های با مقیاس بالا نیز کارآمدتر عمل کند و خصوصیت مقیاس‌پذیری این روش را فراهم کند. همچنین ماتریس وزن بازسازی برای انتخاب ویژگی، از طریق محدودیت رتبه پایین که تحت نظارت ماتریس وزنی بدست آمده ساخته می‌شود، که سبب حفظ ساختار و ارتباط سراسری داده‌ها می‌شود. با استفاده از مدل پیشنهادی این مقاله، ویژگی‌های انتخاب شده توسط مدل جدید، با صرف پیچیدگی محاسباتی پایین‌تر، به یک عملکرد طبقه‌بندی مناسب دست می‌یابند که تجربیات گسترده در مجموعه داده‌های استاندارد مختلف، اثربخشی روش پیشنهادی را تأیید می‌کند. برای ادامه‌ی کار این مقاله، در نظرگرفتن سودمندی مجموعه‌ی چندتایی ویژگی‌ها، در حین انتخاب ویژگی، می‌تواند سودمند باشد.

جدول (۳): میانگین دقت کلاس‌بندی(mACC)برای ۱۰ بار اجرای SVM

مجموعه‌ی داده	LLES	RSR	[28]	MCFS	RFS	LS	RUFS	روش پیشنهادشده	تعداد ویژگی انتخاب‌شده
Wine	۹۵,۸۸	۹۵,۶۶	۹۴,۸۳	۹۳,۳۶	۹۲,۴۴	۹۳,۸۰	۹۳,۸۲	۹۶,۰۴	۱۲
Yale	۷۱,۳۴	۷۲,۵۹	۷۶,۰۳	۶۹,۴۲	۷۶,۱۰	۷۴,۲۹	۷۱,۱۲	۷۶,۲۸	۳۰۰
Colon	۶۳,۹۵	۶۹,۰۴	۷۰,۱۵	۶۳,۱۶	۶۲,۴۵	۷۰,۳۷	۷۰,۴۵	۷۱,۶۶	۳۰۰
WarpAR10p	۸۰,۶۴	۶۱,۷۸	۷۷,۵۹	۷۸,۷۹	۶۹,۸۹	۷۲,۱۴	۸۱,۲۳	۷۳,۰۷	۳۰۰
ORL	۹۰,۱۷	۹۴,۴۷	۸۹,۸۰	۹۲,۴۰	۹۱,۸۴	۹۵,۴۰	۹۶,۴۱	۹۲,۰۰	۳۰۰
GLIOMA	۷۳,۸۴	۷۴,۴۳	۸۰,۹۶	۶۵,۴۹	۷۵,۷۶	۷۴,۴۶	۷۶,۱۴	۸۲,۰۱	۳۰۰

مراجع

[1] Taradeh M, Mafarja M, Heidari AA, Faris H, Aljarah I, Mirjalili S, Fujita H. An evolutionary gravitational search-based feature selection. Information Sciences. 2019 Sep 1;497:219-39.

[2] Effrosynidis D, Arampatzis A. An evaluation of feature selection methods for environmental data. Ecological Informatics. 2021 Mar 1;61:101224.

[3] Yuan H, Li J, Lai LL, Tang YY. Joint sparse matrix regression and nonnegative spectral analysis for two-dimensional unsupervised feature selection. Pattern Recognition. 2019 May 1;89:119-33.

[4] Zhang H, Zhang R, Nie F, Li X. An efficient framework for unsupervised feature selection. Neurocomputing. 2019 Nov 13;366:194-207.

[5] Tabatabaei M S, Yazdian-Dehkordi M, Jahangard Rafsanjani A. Predicting Dimensional Deviation of Ceramic Tiles using Machine Learning Methods. Journal of Iranian Association of Electrical and Electronics Engineers. 2022; 19 (2) :199-206.

[6] Saberi-Movahed F, Mohammadifard M, Mehrpooya A, Rezaei-Ravari M, Berahmand K, Rostami M, Karami S, Najafzadeh M, Hajinezhad D, Jamshidi M, Abedi F. Decoding clinical biomarker space of covid-19: Exploring matrix factorization-based feature selection methods. Computers in biology and medicine. 2022 Jul 1;146:105426.

[7] Cai J, Luo J, Wang S, Yang S. Feature selection in machine learning: A new perspective. Neurocomputing. 2018 Jul 26;300:70-9.

[8] Rostami M, Berahmand K, Nasiri E, Forouzandeh S. Review of swarm intelligence-based feature selection methods. Engineering Applications of Artificial Intelligence. 2021 Apr 1;100:104210.

[9] Balogun AO, Basri S, Mahamad S, Abdulkadir SJ, Capretz LF, Imam AA, Almomani MA, Adeyemo VE, Kumar G. Empirical analysis of rank aggregation-based multi-filter feature selection methods in software defect prediction. Electronics. 2021 Jan 15;10(2):179.

[10] Solorio-Fernández S, Carrasco-Ochoa JA, Martínez-Trinidad JF. A review of unsupervised feature selection methods. Artificial Intelligence Review. 2020 Feb;53(2):907-48.

[11] Manbari Z, AkhlaghianTab F, Salavati C. Hybrid fast unsupervised feature selection for high-dimensional data. Expert Systems with Applications. 2019 Jun 15;124:97-118.

[12] Morales M, Grokking Deep Reinforcement Learning. MANING SHELTER ISLAND. 2020 Nov 10; 3-54.

[13] van der Weij T, Aguilar VS, Solorio-Fernández S. Runtime Prediction of Filter Unsupervised Feature Selection Methods. Research in Computing Science. 2022;150(8):138-50.

[14] Han X, Liu P, Wang L, Li D. Unsupervised feature selection via graph matrix learning and the low-dimensional space learning for classification. Engineering Applications of Artificial Intelligence. 2020 Jan 1;87:103283.

[15] Keyvanpour M R, Moghadam Charkari N. Interactive Retrieval of Natural Images Using Multiple Instance Learning. Journal of Iranian Association of Electrical and Electronics Engineers. 2009; 6 (1) :19-35.

[16] Shi C, Gu Z, Duan C, Tian Q. Multi-view adaptive semi-supervised feature selection with the self-paced learning. Signal Processing. 2020 Mar 1;168:107332.

[17] Zheng W, Yan H, Yang J. Robust unsupervised feature selection by nonnegative sparse subspace learning. Neurocomputing. 2019 Mar 21;334:156-71.

[18] Anaraki JR, Usefi H. A feature selection based on perturbation theory. Expert Systems with Applications. 2019 Aug 1;127:1-8.

[19] Bolón-Canedo V, Alonso-Betanzos A. Ensembles for feature selection: A review and future trends. Information Fusion. 2019 Dec 1;52:1-2.
[20] Zare M, Eftekhari M, Aghamollaei G. Supervised feature selection via matrix factorization based on singular value decomposition. Chemometrics and Intelligent Laboratory Systems. 2019 Feb 15;185:105-13.
[21] Nie F, Huang H, Cai X, Ding C. Efficient and robust feature selection via joint ℓ2, 1-norms minimization. Advances in neural information processing systems. 2010;23.
[22] Tang C, Bian M, Liu X, Li M, Zhou H, Wang P, Yin H. Unsupervised feature selection via latent representation learning and manifold regularization. Neural Networks. 2019 Sep 1;117:163-78.
[23] Du L, Ren C, Lv X, Chen Y, Zhou P, Hu Z. Local graph reconstruction for parameter free unsupervised feature selection. IEEE Access. 2019 Jul 23;7:102921-30.
[24] Shang R, Meng Y, Wang W, Shang F, Jiao L. Local discriminative based sparse subspace learning for feature selection. Pattern Recognition. 2019 Aug 1;92:219-30.
[25] Zhou P, Chen J, Fan M, Du L, Shen YD, Li X. Unsupervised feature selection for balanced clustering. Knowledge-Based Systems. 2020 Apr 6;193:105417.
[26] Huang D, Cai X, Wang CD. Unsupervised feature selection with multi-subspace randomization and collaboration. Knowledge-Based Systems. 2019 Oct 15;182:104856.
[27] He X, Cai D, Niyogi P. Laplacian score for feature selection. Advances in neural information processing systems. 2005;18.
[28] Zhang Y, Lu Z, Wang S. Unsupervised feature selection via transformed auto-encoder. Knowledge-Based Systems. 2021 Mar 5;215:106748.
[29] Zeng Z, Wang X, Yan F, Chen Y. Local adaptive learning for semi-supervised feature selection with group sparsity. Knowledge-Based Systems. 2019 Oct 1;181:104787.
[30] Yao C, Liu YF, Jiang B, Han J, Han J. LLE score: A new filter-based unsupervised feature selection method based on nonlinear manifold embedding and its application to image recognition. IEEE Transactions on Image Processing. 2017 Jul 28;26(11):5257-69.
[31] Ye Q, Zhang X, Sun Y. Dual Global Structure Preservation Based Supervised Feature Selection. Neural Processing Letters. 2020 Mar 14:1-23.
[32] Shang R, Xu K, Shang F, Jiao L. Sparse and low-redundant subspace learning-based dual-graph regularized robust feature selection. Knowledge-Based Systems. 2020 Jan 1;187:104830.
[33] Cai D, Zhang C, He X. Unsupervised feature selection for multi-cluster data. InProceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining 2010 Jul 25 (pp. 333-342).
[34] Zhang Y, Wang Q, Gong DW, Song XF. Nonnegative Laplacian embedding guided subspace learning for unsupervised feature selection. Pattern Recognition. 2019 Sep 1;93:337-52.
[35] Manosij G, Ritam G, Sarkar R, Abraham A. A wrapper-filter feature selection technique based on ant colony optimization. Neural Computing & Applications. 2020 Jun 1;32(12):7839-57.
[36] Zhong J, Wang N, Lin Q, Zhong P. Weighted feature selection via discriminative sparse multi-view learning. Knowledge-Based Systems. 2019 Aug 15;178:132-48.
[37] Zhu P, Zuo W, Zhang L, Hu Q, Shiu SC. Unsupervised feature selection by regularized self-representation. Pattern Recognition. 2015 Feb 1;48(2):438-46.
[38] Hu H, Wang R, Nie F, Yang X, Yu W. Fast unsupervised feature selection with anchor graph and ℓ 2, 1-norm regularization. Multimedia Tools and Applications. 2018 Sep;77(17):22099-113.
[39] Song H, Yang W, Bai Y, Xu X. Unsupervised classification of polarimetric SAR imagery using large-scale spectral clustering with spatial constraints. International Journal of Remote Sensing. 2015 Jun 3;36(11):2816-30.
[40] Yao C, Liu YF, Jiang B, Han J, Han J. LLE score: A new filter-based unsupervised feature selection method based on nonlinear manifold embedding and its application to image recognition. IEEE Transactions on Image Processing. 2017 Jul 28;26(11):5257-69.
[41] Chen L, Huang JZ. Sparse reduced-rank regression for simultaneous dimension reduction and variable selection. Journal of the American Statistical Association. 2012 Dec 1;107(500):1533-45.
[42] Zhu X, Zhang S, Hu R, Zhu Y. Local and global structure preservation for robust unsupervised spectral feature selection. IEEE Transactions on Knowledge and Data Engineering. 2017 Oct 25;30(3):517-29.

زیرنویس‌ها

[1] Dimensionality reduction
[2] Feature Selection
[3] Semi-Supervised Feature Selection
[4] Fitness value
[5] low-rank constraint
[6] Robust Feature Selection
[7] Loss Function
[8] Laplacian Score
[9] Regularization
[10] Semi-Supervised Feature Selection
[11] Self-Paced Learning
[12] Locally Linear Embedding
[13] Locally Linear Embedding Score
[14] Multi-Cluster Feature Selection
[15] Gravitational Search algorithm
[16] Binary Ant System
[17] Regularized Self-Representation
[18] Anchor
[19] Reconstruction Weight Matrix
[20] UCI Machine Learning Repository: Data Sets
[21] Datasets | Feature Selection @ ASU (jundongl.github.io)
[22] Support Vector Machine
[23] Train
[24] Grid-Search Strategy