بهار ۱۴۰۰، دوره ۶، شماره ۱، پیاپی ۲۰
DOR :20.1001.1.2538161.1400.6.1.3.1

تاریخ دریافت مقاله: ۹۸/۱۱/۲۵
تاریخ پذیرش مقاله: ۹۹/۰۵/۱۱

نوع مقاله: پژوهشی

تشخیص نشانگرهای زیستی سرطان: رویکرد سریع بیوانفورماتیک

مریم رزمجویی*
کارشناسی ارشد گروه مهندسی کامپیوتر، دانشکده مهندسی، دانشگاه بین‌المللی امام خمینی، قزوین، ایران
پست الکترونیکی: maryam.raz21@gmail.com

حمیدرضا حمیدی
استادیار گروه مهندسی کامپیوتر، دانشکده مهندسی، دانشگاه بین‌المللی امام خمینی، قزوین، ایران
پست الکترونیکی: hamidreza.hamidi@eng.ikiu.ac.ir

اسماعیل ابراهیمی
استادیار گروه بیوتکنولوژی، دانشکده کشاورزی، دانشگاه شیراز، شیراز، ایران
پست الکترونیکی: ebrahimiet@gmail.com

چکیده

نشـــانگرهای زیســـتی در تشـــخیص زودهنگام انوع بیماری‌ها از جمله ســـرطان نقش عمده‌ای دارند. بر اساس تعریف سازمان بهداشـــت جهانی، هر ساختار یا فرآیندی در بدن که قابـــل اندازه‌گیری بوده و بر پیش‌بینی یا نتیجه بیماری اثرگذار باشـــد، به‌عنوان نشانگر زیستی[1] شناخته می‌شود. امروزه شناسایی نشانگرهای زیستی با استفاده از ابزارهـــای بیوانفورماتیـــک امکان‌پذیر اســـت. مســـئله تشخیص نشـــانگرهای زیســـتی در حوزه بیوانفورماتیک بیشـــتر به‌عنوان یک مســـئله انتخاب ویژگی مطرح است. الگوریتم‌های انتخاب ویژگی متعددی در زمینه شناســـایی نشـــانگرهای زیستی مورد اســـتفاده قرار می‌گیرد. امّا این الگوریتم‌ها یا از دقت کافی برخوردار نیستند و یا دقت لازم را داشته امّا از پیچیدگی محاســـباتی بالایی برخوردارند. به همین دلیل الگوریتم‌هایی که دقت بیشـــتری داشـــته‌اند، تنهـــا به‌دلیل زمانبر بودن کنار گذاشـــته می‌شـــوند. هدف اصلـــی در این پژوهش انتخـــاب الگوریتمی با دقت بالا در شناسایی نشانگرهای زیستی است که در نهایت با استفاده از شیوه‌های موازی‌سازی، ســـرعت آن را بهبود بخشیم. الگوریتم انتخابی با اســـتفاده از روش‌های موازی‌ســـازی باز طراحی شده است و با اســـتفاده از داده‌ای که مربوط به سرطان ســـینه می‌باشـــد، به ارزیابی روش پیشنهادی پرداخته‌ایم. در نهایت الگوریتم توانسته است با همان دقت الگوریتم اصلی، ولی با افزایش سرعتی در حدود یک و نیم برابر، خروجی مورد نظر را تولید می‌کند.

واژه‌هـــای کلیدی: نشـــانگر زیســـتی، بیوانفورماتیک، سرطان سینه، الگوریتم موازی، الگوریتم ام.اس.وی.ام–ای. اف.ای

۱- مقدمه

اســـتفاده از نشانگر زیستی در تشخیص به‌هنگام انواع بیماری‌ها از جمله سرطان نقش عمده‌ای دارد [۱]. هرماده،

* نویسندهٔ مسئول

1- Biomarker

ساختار یا فرآیندی که در بدن افراد قابل اندازه‌گیری بوده و بر پیش‌بینی یا روند بیماری اثرگذار باشد به‌عنوان نشانگر زیستی شناخته می‌شود [۲]. نشانگرهای زیستی شاخصی هستند که هم در زمان بیماری و هم در زمانی که فرد سالم است در بدن وجود دارند و در مایعات موجود در بدن و بافت‌ها یافت می‌شوند. نشانگرهای زیستی قادرند بیماری را قبل از ظاهرشدن علائم بالینی تشخیص دهند بنابراین استفاده از آن‌ها در تشخیص زود هنگام بیماری‌ها بسیار حائز اهمیت است [۳]. نشانگرهای زیستی از تحلیل بیومولکول‌هایی مانند دی.ان.ای[۲]، آر.ان.ای[۳]. و پروتئین به‌دست می‌آیند و خود می‌توانند پروتیئن، ژن، هورمون و آنزیم باشند[٤].

کشف نشانگرهای زیستی سرطان، تشخیص زودهنگام سرطان را در پی داشته که خود تاثیر بسزایی در کاهش مرگ و میر ناشی از این بیماری خواهد داشت. همچنین نظارت بر روند درمانی، تشخیص و به‌کارگیری درمان مناسب، ارزیابی وضعیت بیماری، ساخت دارو و تجویز داروی مناسب به کمک نشانگرهای زیستی قابل انجام است [۲]. امروزه تشخیص نشانگرهای زیستی سرطان به کمک علم بیوانفورماتیک امکان‌پذیر است. این علم با استفاده از مجموعه داده عظیم و ابزارهای محاسباتی، امکان ذخیره، بازیابی، تجزیه و تحلیل داده را به وجود می‌آورد.

در بیوانفورماتیک، کشف نشانگرهای زیستی سرطان بیشتر به‌عنوان یک مسئله انتخاب ویژگی مطرح است، بخصوص زمانی که تمایز بین ویژگیها از اهمیت بالایی برخوردار است [٥]. انتخاب ویژگی به معنای یافتن یک زیرمجموعه با حداقل اندازه ممکن از ویژگی‌ها است که برای هدف مورد نظر اطلاعات لازم و کافی را در بر داشته باشد. در مسئله تشخیص نشانگرهای زیستی نیز با حجم زیادی از ویژگی‌ها و نمونه‌ها روبرو هستیم. هدف انتخاب یک زیر مجموعه حداقل از ویژگی‌هاست که به نمونه‌ها بسیار نزدیک و کارآمد باشند. این مجموعه در بردارنده ویژگی‌هایی خواهد بود که بیشترین تاثیر در تفکیک و تمایز نمونه‌ها از یکدیگر دارند.

الگوریتم‌های انتخاب ویژگی بسیار متنوع بوده و بسیاری از آن‌ها در تشخیص نشانگرهای زیستی مورد استفاده قرار گرفته است. این دسته از الگوریتم‌ها علاوه بر برگرداندن مجموعه‌ای از ویژگی‌ها به‌عنوان خروجی، کاهش ابعاد داده، کاهش افزونگی و افزایش دقت را در پی دارند [٦]. به کمک این الگوریتم‌ها، شناسایی و تایید نشانگرهای زیستی در چند گام امکان‌پذیر است. ابتدا داده ژنومی و پروتئومی مورد نیاز جمع‌آوری شده و توسط پایگاه های داده سازماندهی می‌شود. همچنین داده غیرضروری حذف می‌شوند. مرحله دوم شامل انتخاب ویژگی‌ها و در بعضی موارد استفاده از روش‌های دسته‌بندی است. نشانگرهای زیستی نامزد در مرحله بعد با ابزارهای مناسب اعتبار سنجی می‌شوند [۷].

۲- الگوریتم‌های انتخاب ویژگی

الگوریتم‌های انتخاب ویژگی در یک دسته‌بندی کلی در دو دسته پالایه[٤] و بسته‌بند[٥] قرار می‌گیرند. الگوریتم‌های پالایه، براساس ویژگی ذاتی داده کار می‌کنند نه بر اساس دسته‌بندی. این دسته از الگوریتم‌ها معمولا ساده بوده و از پیچیدگی محاسباتی بالایی برخوردار نیستند. به همین دلیل سرعت اجرایی بالا و قابل قبولی دارند. امّا این الگوریتم‌ها از دقت بالایی برخوردار نبوده و معمولا پایدار نیستند. بدین معنا که در هر بار تکرار معمولا ویژگی‌های متفاوتی را برمی‌گردانند [۸]. از نمونه این الگوریتم‌ها می‌توان به الگوریتم رلیف اشاره کرد که فاقد پایداری است و در تشخیص نشانگرهای زیستی مورد استفاده قرار گرفته است [۹]. الگوریتم‌های بسته‌بند به جای استفاده از یک تابع معیار، از الگوریتم‌های یادگیری ماشین برای امتیاز دهی به ویژگی‌ها استفاده می‌کنند که معمولا الگوریتم‌های دسته‌بندی هستند. الگوریتم‌های رَپِر پیچیدگی محاسباتی

2- DNA
3- RNA
4- Fitter
5- Wrapper

بالایی داشــته و از ســرعت کافی برخوردار نیستند امّا به دلیل اســتفاده از الگوریتم‌هــای دســته‌بندی، روابط بین ویژگی‌ها در نظر گرفته می‌شود و الگوریتم‌ها، دقت بالایی دارند [۸].

الگوریتم‌هــای انتخاب ویژگی متعددی در تشــخیص نشــانگرهای زیستی سرطان و همچنین تشخیص ژن‌های دسته‌بندی‌کننده ســرطان مورد استفاده قرار گرفته است. یکی از پرکاربردترین الگوریتم‌های اســتفاده شده در این حوزه، الگوریتم ماشــین بردار پشتیبان است که به‌عنوان الگوریتــم بســته‌بند مورد اســتفاده قرار گرفــت [۱۰]. به دلیل این‌که این الگوریتم یک الگوریتم دســته‌بندی بوده و محدودیت‌هایی به‌عنوان الگوریتم انتخاب ویژگی داشــت، گویُــن[6] ترکیبی از این الگوریتــم و الگوریتم حذف ویژگی بازگشتی را در سال ۲۰۰۲ به نام الگوریتم (اس.وی.ام.-آر. اف.ای)[7] ارائه کرد [۱۱]. پس از آن الگوریتم ماشــین بردار پشــتیبان به‌عنوان یکی از بهترین الگوریتم‌های دسته‌بندی در کنار الگوریتم‌های انتخاب ویژگی مورد اســتفاده قرار گرفت و الگوریتم بسته‌بند ارائه شده توسط گوین به‌عنوان یک الگوریتم معیار برای دیگر الگوریتم‌ها شد [۱۲].

همفیــل[8] و همکارنــش مقایســه جامــع و کاملــی از الگوریتم‌های بسته‌بند برای شناسایی نشانگرهای زیستی ارائه کردند. از الگوریتم‌های حذف ویژگی بازگشــتی (آر. اف.ای.)، جنگل تصادفی[9]، اکســترا تــری[10] و آنووا[11] برای انتخاب ویژگی استفاده شد و از الگوریتم‌های ماشین بردار پشــتیبان، جنگل تصادفی، درخت تصمیم و چند الگوریتم دیگر به‌عنوان الگوریتم‌های دسته‌بندی استفاده شد. تمامی ترکیبات ممکــن از این الگوریتم‌ها (یــک الگوریتم انتخاب ویژگی به همراه یک الگویتم دســته‌بندی) در این مقایســه با یکدیگر ســنجیده شــدند [۱۲]. این مقایسه بر روی دادهٔ اطلس ژنوم ســرطان انجام شد که چندین نوع داده شامل میکــروآر.ان.ای[12]، ام.آر.ان.ای[13]، پروتیئن و اس.ان.پی[14] و ســی.ان.وی[15] (تغییــرات ژنومی در مقیاس بــزرگ) بود. مقایسه‌این محققان از این نظر حائز اهمیت بود که عملکرد و کارایــی تمامی الگوریتم‌ها بــر روی نوع داده مختلف با اندازه‌های مختلفی از نشانگرهای زیستی به تصویر کشیده شــد. در این مقایســه در نهایت الگوریتم اس.وی.ام.-آر. اف.ای بهترین عملکرد را نشــان داد و عملکرد آن بر روی داده ژنومی بهتر از داده پروتئینی بود [۱۲].

۱،۲. چالش انتخاب الگوریتم

تعــداد زیادی الگوریتمهای انتخاب ویژگی وجود دارد که عموما بر روی انواع مختلف داده از نظر زیست‌شناسی، عملکرد متفاوتی دارند. بنابراین تصمیم‌گیری در این رابطه که کدام الگوریتم برای کدام مجموعه داده مناســب اســت و موفقیــت حداکثری دارد، آســان نیســت [۸]. علاوه بر ایــن به دلیل حجم رو به رشــد داده در این حوزه، انتخاب الگوریتم‌های ســریع از اولویت برخوردار اســت. امّا تنها الگوریتم‌های دسته پالایه با این حجم داده می‌توانند خروجی را در زمان قابل قبولی برگردانند. همان‌طور که بیان شــد ایــن الگوریتم‌ها دقت لازم را در تشــخیص نشــانگرهای زیســتی ســرطانی ندارند، در حالی که تشخیص نادرست نشانگرهای زیســتی می‌تواند مشکلات متعددی ایجاد کند [۸]. به همین دلیل امروزه استفاده از الگوریتم‌های بسته‌بند به‌دلیل دقت بالا در تشخیص صحیح نشانگرهای زیستی، به ســرعت در حال افزایش است. امّا پیچیدگی محاسباتی این الگوریتم‌ها و زمانبر بودن آن‌ها باعث شــده که اجرای یک الگوریتم چندین روز به طول انجامد [۱۲].

یکــی از اصلی‌تریــن راهکارها بــرای کاهش پیچیدگی محاســباتی الگوریتم‌ها، روش موازی‌سازی است. رویکرد موازی‌سازی الگوریتم‌ها به معنای افزایش تعداد پردازنده و تقســیم کار یا داده بر روی آن‌هاســت تا بار محاسباتی بر روی پردازنده‌ها تقسیم شده و زمان پاسخ کاهش یابد [۱۳].

6- Guyon
7- Support vector machines- Recursive Feature Elimination (SVM-RFE)
8- Hemphil
9- Random Forest
10- Extra tree
11- Anova
12- MicroRNA
13- mRNA
14- SNP
15- CNV

۳- معرفی الگوریتم اس.وی.ام-آر.اف.ای.

الگوریتم اس.وی.ام-آر.اف.ای به‌عنوان یکی از بهترین الگوریتم‌های انتخاب ویژگی جهت تشخیص نشانگرهای زیستی مورد توجه بسیاری از پژوهشگران این حوزه است. این الگوریتم یک الگوریتم پایدار است بدین معنا که در هر بار تکرار، خروجی الگوریتم ثابت است. همچنین این الگوریتم از دقت و صحت خوبی در تشخیص نشانگرهای زیستی برخوردار است و تنها به‌دلیل استفاده از الگوریتم ماشین بردار پشتیبان، بسیار زمانبر است به‌طوری که استفاده از این الگوریتم را محدود می‌کند. این الگوریتم بر روی داده ژنومی عملکرد بسیار خوبی از خود نشان می‌دهد [۱۲].

۱,۳. دادهٔ پژوهش

اطلس ژنوم سرطان[16] مجموعه داده‌ای عظیم از تغییرات ژنومی در بیش از ۳۳ نوع سرطان است که مجموعه داده ارزشمندی برای استفاده در ابزارهای محاسباتی می‌باشد. این مجموعه داده، با همکاری موسسه ملی سرطان و موسسه ملی تحقیقات ژنوم انسانی ایجاد گردید تا نقشه‌ای جامع و چند بُعدی از تغییرات ژنومی را فراهم نماید. در این مجموعه داده تغییرات ژنومی، توالی دی.ان.ای. و بیان ژن‌ها در یک تومور سلولی نسبت به یک سلول سالم بررسی می‌شود. دادهٔ حاصل از این نتایج، به‌صورت هزاران نمونه ادغام شده و در دسترس عموم کاربران قرار می‌گیرد [۱٤].

داده‌ای که در این پژوهش مورد استفاده قرار گرفته است شامل داده بیان ژن مربوط به بیش از هزار بیمار مبتلا به سرطان سینه است. این داده شامل بیان ژن‌ها، سوخت‌وساز و دادهٔ کلینیکی است که مربوط به سال‌های ۲۰۱۲ تا ۲۰۱۵ است. با ادغام این مجموعه ها، تعداد نمونه‌ها نزدیک به ۲۰۰۰ بیمار بوده و تعداد ویژگی‌ها به ۱۹۰۰ ویژگی می‌رسد. داده از نوع فایلِ اکسل است که سطرها را نمونه‌ها (بیماران) تشکیل داده و ستون‌ها شامل ویژگی‌ها می‌باشند. ستون اول جدول مربوط به وضعیت بیمار است که شامل دو حالت زنده یا مرده است و در تحلیل، این ستون نقش برچسب را برعهده خواهد داشت. در واقع در این داده‌ها به دنبال آن نشانگرهای زیستی هستیم که بیشترین نقش را در زنده ماندن یا مردن بیمار ایفا می‌کنند.

۲,۳. الگوریتم ام. اس.وی.ام-آر.اف.ای

دوآن[17] و همکارن در سال ۲۰۰۵ پیاده‌سازی دیگری از الگوریتم اس.وی.ام-آر.اف.ای ارائه کردند که در آن به جای استفاده از یک ماشین بردار پشتیبان در هر مرحله، از چندین ماشین بردار پشتیبان استفاده شد و ویژگی‌ها توسط ماشین‌های مختلف مورد سنجش قرار گرفتند [۱۵]. این الگوریتم برای انتخاب ژن‌های موثر در دسته‌بندی سرطان مورد استفاده قرار گرفت که نسبت به الگوریتم اصلی ویژگی‌های بهتری را برمی‌گرداند و کیفیت دسته‌بندی ایجاد شده نیز بهتر است. به دلیل استفاده از چندین رده در مرحله دسته‌بندی، این الگوریتم اس.وی.ام-آر.اف.ای-چندگانه (ام. اس.وی.ام-آر.اف.ای)[18] نامیده شد [۱۵].

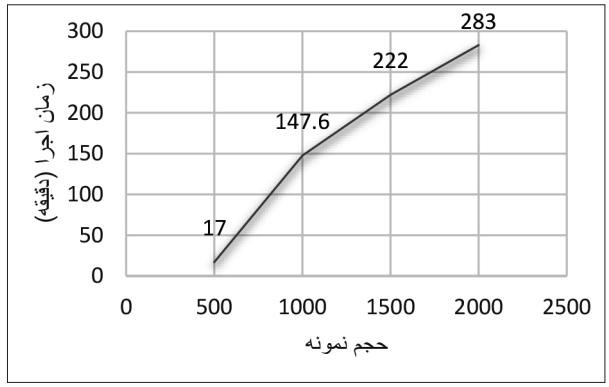
روش الگوریتم ام.اس.وی.ام-آر.اف.ای به این صورت است که داده را از کاربر دریافت کرده و سپس با استفاده از روش کا-فلد، زیرنمونه‌های مختلف از نمونه اصلی را ایجاد می‌کند. تعداد این زیرنمونه‌ها(فُلدها) با متغیر K و توسط کاربر معرفی می‌شود. بنابراین اگر K=10 باشد، ۱۰ زیرنمونه خواهیم داشت که اندازه هر نمونه به داده اولیه بستگی دارد. در گام دوم ماشین‌های بردار پشتیبان (که تعداد آن‌ها توسط کاربر مشخص شده است) بر روی هر زیرنمونه آموزش داده می‌شوند و ارزش هر ویژگی با استفاده از تمامی ماشین‌ها تخمین زده می‌شود. هر ماشین، بردار وزن مخصوص به خود دارد و برای هر ویژگی مانند i، ارزش آن از فرمول زیر محاسبه می‌شود:

$$\text{ارزش } i = \frac{\text{میانگین ارزش های به‌دست آمده از ماشین ها}}{\text{انحراف معیار}}$$

16- The Cancer Genome Atlas Program

17- Duan

18- MSVM-RFE: Multiple SVM-RFE



نمودار ۱: تاثیر حجم داده ورودی بر زمان اجرا

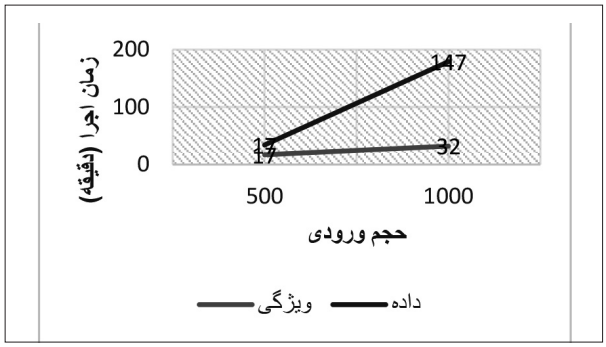
در گام سوم، آن ویژگی که کمترین ارزش را داشته باشد،حذف می‌شود. همچنین این ویژگی به همراه مقدار ارزش آن در لیست R قرار داده می‌شوند. گام دوم و سوم مرتبا تکرار می‌شوند تا زمانی که برای تمامی ویژگی‌ها، ارزش آن‌ها به‌دست آمده و در لیست R قرار گیرند. پس از پایان این روند بازگشتی، ٥ ویژگی با بالاترین ارزش به‌عنوان خروجی برگردانده می‌شوند [۱۵]. همچنین این الگوریتم، خطای حاصل از هر زیرنمونه را برای تعدادی از ویژگی‌هایی که ارزش بیشتری کسب کرده‌اند، برآورد می‌کند. این خطا برای سنجش کارایی دسته‌بندی انجام می‌شود. برآورد خطا از میانگین تمامی خطاهای ماشین‌های بردار پشتیبان روی زیرنمونه‌های مختلف به‌دست می‌آید.

در این الگوریتم مانند الگوریتم اس.وی.ام-آر.اف.ای درصورتی که تعداد ویژگی‌های اولیه بسیار زیاد باشد، در هر تکرار می‌توان به جای یک ویژگی، چند ویژگی را حذف کرد. الگوریتم ام.اس.وی.ام-آر.اف.ای در مقایسه با الگوریتم اس.وی.ام-آر.اف.ای پرهزینه‌تر است به‌دلیل این‌که به جای استفاده از یک ماشین از چندین ماشین بردار پشتیبان استفاده می‌کند. امّا مقایسه این دو نشان می‌دهد که این هزینه در نهایت به انتخاب ویژگی‌های بهتر و دسته‌بندی با دقت بیشتر منجر خواهد شد. علاوه بر این، یکی از راه‌های افزایش پایداری الگوریتم‌ها این است که عمل انتخاب بر روی زیرمجموعه های مختلف از نمونه‌ها انجام شود و نه فقط یک نمونه که این روش در این الگوریتم پیاده‌سازی شده است [۱۵]. هر دو الگوریتم بر روی مجموعه دادهٔ سرطان سینه، ریه، روده بزرگ و خون انجام شد که بهترین ویژگی‌ها برای سرطان سینه (داده پژوهشی ما از این نوع می‌باشد)، روده و خون با استفاده از الگوریتم ام. اس.وی.ام-آر.اف.ای به‌دست آمد. و تنها در سرطان روده دقت ردهٔ ایجاد شده توسط اس.وی.ام-آر. اف.ای از ام. اس.وی.ام-آر.اف.ای بیشتر بود [۱۵]. بنابر توضیحات ذکر شده الگوریتم ام. اس.وی.ام-آر.اف.ای به دلیل دقت و کارایی بالا در تشخیص نشانگرهای زیستی و از طرفی به دلیل داشتن محاسبات پیچیده به‌عنوان الگوریتم مورد نظر جهت موازی‌سازی در نظر گرفته شد.

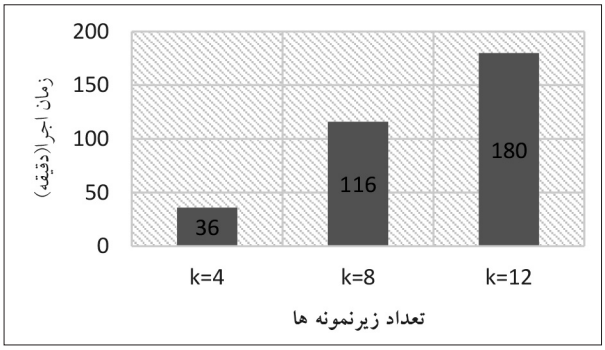
۳,۳ تاثیر متغیرهای مختلف بر زمان اجرای الگوریتم

در این الگوریتم اگر حجم داده اولیه را n در نظر بگیریم، با افزایش حجم داده، و ثابت بودن تعداد زیرنمونه‌ها (فُلدها)، اندازه هر زیرنمونه نیز افزایش می‌یابد. برای مثال اگر حجم داده اولیه ٦۰۰ باشد و تعداد زیرنمونه‌ها را ٤ در نظر بگیریم، اندازه هر زیرنمونه ۱۵۰ خواهد بود و در صورتی که حجم داده اولیه را دو برابر کنیم و به ۱۲۰۰ افزایش یابد، اندازه هر زیرنمونه به ۳۰۰ افزایش پیدا می‌کند. افزایش حجم داده در افزایش زمان اجرای الگوریتم نقش اساسی دارد. این مسئله در نمودار ۱ به تصویر کشیده شده است. در این مقایسه که بر روی یک سیستم با ۸ گیگابایت حافظه انجام گرفته، تعداد فلدها ثابت و برابر با ٤ در نظر گرفته است. در این نمودار با دو برابر کردن داده اولیه که منجر به دو برابر شدن اندازه زیرنمونه‌ها می‌شود، زمان اجرای الگوریتم چندین برابر شده است.

شایان ذکر است که در این مقایسه منظور از افزایش داده اولیه، افزایش نمونه‌ها و ویژگی‌ها متناسب با یکدیگر می‌باشد امّا این نمونه‌ها هستند که بیشترین نقش را در افزایش زمان اجرا ایفا می‌کنند. افزایش تعداد ویژگی‌ها، تاثیر بسیار کمتری در افزایش زمان اجرای الگوریتم خواهد داشت به این دلیل که در این الگوریتم زمانی که تعداد ویژگی‌ها زیاد باشد در هر بار تکرار الگوریتم، تعداد ویژگی‌ها نصف شده تا به کمتر از ۱۰۰ ویژگی برسند[۱۵].



نمودار ۲: تاثیر افزایش تعداد ویژگی‌ها بر زمان اجرا



نمودار ۳: تاثیر تعداد زیرنمونه‌ها بر زمان اجرا

بنابراین تعداد ویژگی‌ها سریعاً کاهش یافته و نمی‌تواند تاثیر چندانی در سرعت الگوریتم داشته باشند. برای واضح‌تر شدن موضوع، در یک آزمایش، تعداد نمونه‌ها را ۵۰۰ در نظر گرفته و زمان اجرای الگوریتم را در حالتی که تعداد ویژگی‌ها برابر با ۵۰۰ و ۱۰۰۰ بود مقایسه کردیم. در نمودار ۲، رنگ سبز بیانگر میزان افزایش زمان اجرا در این حالت است. همان‌طور که مشاهده می‌شود با افزایش تعداد ویژگی‌ها از ۵۰۰ به ۱۰۰ زمان اجرا دو برابر شده است. حال آنکه اگر حجم نمونه را نیز به ۱۰۰۰ افزایش دهیم، زمان اجرای الگوریتم تقریبا هشت برابر شده است(نمودار آبی رنگ). بنابراین افزایش حجم نمونه، زمان اجرای الگوریتم را به شدت افزایش می‌دهد.

الگوریتم ام. اس.وی.ام-آر.اف.ای به دلیل این‌که از الگوریتم‌های دسته‌بندی استفاده می‌کند، دارای پیچیدگی محاسباتی بالایی است. از طرفی به دلیلی استفاده از چندین ماشین بردار پشتیبان در هر مرحله، پیچیدگی آن افزایش یافته است. در این الگوریتم در صورتی که در تابع کا-فلد تعداد متغیر k را مثلا ۱۰ بگیریم، بدین معناست که برنامه باید از داده اولیه بر اساس نمونه‌ها (و نه ویژگی‌ها) ۱۰ زیرنمونه ایجاد کند. تعداد اعضای هر نمونه از تقسیم تعداد ستون‌های داده اولیه بر متغیر k به‌دست می‌آید. تعداد ماشین‌های بردار پشتیبان نیز توسط کاربر تعیین می‌شود. اگر این تعداد را برابر با متغیر t در نظر بگیریم، در صورتی که t=1 باشد، الگوریتم اس.وی.ام-آر.اف.ای اجرا خواهد شد چون تنها یک ماشین بردار پشتیبان داریم و اگر این تعداد بیشتر از یک باشد، الگوریتم ام. اس.وی.ام-آر.اف.ای اجرا می‌شود. هر چه این عدد بزرگ‌تر باشد، دقت افزایش خواهد یافت امّا به موازات آن پیچیدگی نیز افزایش قابل توجهی خواهد داشت. با وجود k زیرنمونه و t ماشین بردار پشتیبان برای هر نمونه، در هر تکرار از الگوریتم تعداد k*t ماشین بردار پشتیبان اجرا خواهد شد[۱۵].

تاثیر افزایش تعداد زیرنمونه‌ها بر زمان اجرا، در نمودار ۳ به تصویر کشیده شده است. حجم داده در این بررسی شامل ۵۰۰ نمونه و ۶۰۰ ویژگی است. همان‌طور که در نمودار مشاهده می‌شود، با افزایش تعداد زیرنمونه‌ها، به دلیل این‌که در هر بار اجرا، تعداد ماشین بردار پشتیبان بیشتری ایجاد و در حال اجرا می‌باشند، پیچیدگی الگوریتم افزایش یافته و زمان اجرا افزایش قابل ملاحظه‌ای دارد. همان‌طور که مشاهده می‌شود، افزایش دقت، هزینه محاسباتی زیادی در پی خواهد داشت.

۴،۳. روش موازی‌سازی

بستر نرم‌افزاری مورد استفاده در این پژوهش نرم‌افزار آر[19] است. نرم‌افزار آر یک زبان و محیط برنامه‌نویسی است که محاسبات آماری (از جمله دسته‌بندی، خوشه‌بندی و ...) و گرافیکی را فراهم می‌کند. الگوریتم ام. اس.وی.ام-آر.اف.ای توسط جان کُلبی[20] در سال ۲۰۱۱ در زبان آر پیاده‌سازی شده است [۱۶]. علاوه بر این کتابخانه‌های مختلفی برای موازی‌سازی در این زبان وجود دارد که مهمترین آن کتابخانه آر.ام.پی.آی[21] است. این کتابخانه

19- R
20- John Colby
21- Rmpi

امکان اســتفاده از فناوری موازی‌ســازی ام.پی.آی را در محیط آر فراهم می سازد [۱۷].

در طراحــی این الگوریتم، به نوعی تقســیم داده انجام شده است. در گام اول الگوریتم تابع کا-فلد داده را تقسیم کرده و زیرنمونه‌های مختلف ایجاد می‌کند. بنابراین نیازی نیست که ما از ابتدا داده را تقسیم کنیم. زیرنمونه‌های ایجاد شده توسط کا-فلد می‌توانند بین پردازنده‌ها تقسیم شوند و ســپس مراحل دیگر الگوریتم توسط هر فرآیند به اجرا درآیــد. در واقع با این عمل یکــی از اصلی‌ترین متغیرهای تاثیرگذار در زمان اجــرا یعنی حجم زیرنمونه‌ها، بر روی پردازنده‌های مختلف تقسیم می‌شود.

تعداد زیرنمونه‌ها در حالت موازی بهتر است بر اساس تعداد فرآیندها تنظیم شــود. زیــرا در این حالت تعادل بار بهتری خواهیم داشــت. اگر رایانه‌ای که در اختیار داریم از ٤ پردازندهٔ همزمان استفاده کند، بنابراین تعداد زیرنمونه‌ها مضربــی از ٤ خواهد بود. اگر در نهایت ۸ زیرمجموعه در اختیار داشــته باشــیم به هر فرآیند، دو زیرنمونه خواهد رســید. در صورتی که تعداد ماشین بردار پشتیبان را ۱۰ در نظر بگیریم هر فرآیند در هر بار تکرار، ۲۰ ماشــینِ در حال اجرا دارد. اگر تعداد زیرنمونه‌ها را با متغیر k نشــان دهیم و تعداد ماشین بردار پشتیبان را با متغیر t، تعداد کل ماشین‌های بردار پشتیبان در هر لحظه از زمان اجرا، برابر با t*k خواهد بود:

$$Number\ of\ SVM = t * k$$

این قســمت از الگوریتــم یکی از عوامــل تعیین کننده در مرتبــه زمانی الگوریتم خواهد بــود. این عمل به تعداد ویژگی‌ها، تکرار می‌شــود. بنابراین اگر تعداد ویژگی‌ها را m و حجــم داده اولیــه را n در نظر بگیریــم مرتبه زمانی الگوریتم در حالت ترتیبی به‌صورت زیر محاسبه می‌شود:

$$(\sum_{i=0}^{m}(t*k))*n = O((m*t*k)*n)$$

با تقســیم تعداد فلدهــا به تعــداد پردازنده‌ها (p) عملا تعداد ماشــین‌های بردار پشــتیبان نیز بر روی فرآیندها تقسیم می‌شــود. در بخش قبل مشــاهده کردیم که تعداد زیرنمونه‌ها و همچنین تعداد ماشین‌های بردار پشتیبان نیز یکی دیگر از متغیرهای تاثیرگذار در زمان اجرا هستند که با موازی‌سازی عملا تعداد ماشین‌ها نیز بر روی پردازنده‌ها تقســیم می‌شوند. اگر تعداد ماشینهای پشتیبان ایجاد شده در مرحله بازگشتی را به‌صورت زیر در نظر بگیریم:

$(t * k)$

بنابراین با در نظر گرفتن حجم نمونه اولیه (n)، مرتبه زمانــی الگوریتم در حالت موازی در قســمت بازگشــتی به‌صورت زیر خواهد بود:

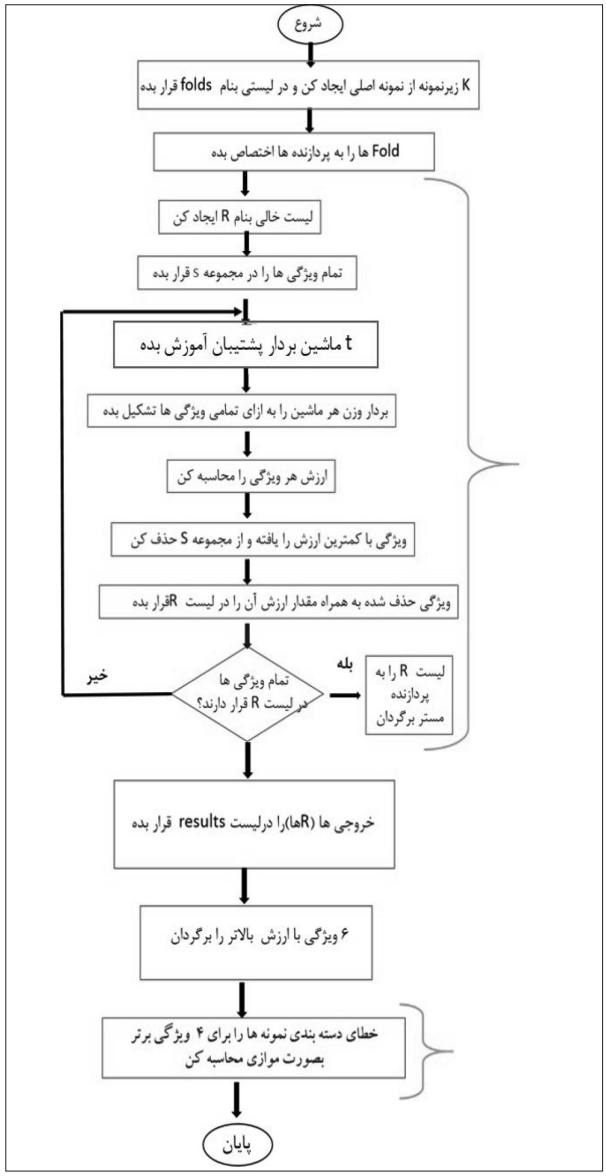
$$[\sum_{i=0}^{m}(\frac{t*k}{p})]*\frac{n}{p} = O\left(\left(m*\frac{t*k}{p}\right)*(\frac{n}{p})\right)$$

در صورتــی که تعــداد پردازنده‌هــا برابر بــا تعداد زیرنمونه‌ها باشــد یعنــی k=p، در این صــورت در حالت موازی، در هر لحظه، تعداد ماشــین‌های بردار پشتیبانِ در حال اجرا، همان t خواهد بود یعنی ماشــین‌ها تنها به ازای یک زیرنمونه در محاســبه زمان اجرا نقش خواهند داشت. در این حالت مرتبه زمانی الگوریتم به‌صورت زیر محاسبه می‌شود:

$$O\left((m*t)*(\frac{n}{p})\right)$$

پس از توزیع زیرنمونه‌ها، داده اصلی به تمام فرآیندها فرستاده می‌شــود تا ویژگی‌ها در اختیار آن‌ها قرار گیرد. پس از اجرای ماشــین‌ها و ارزیابی ویژگی‌ها توســط هر فرآیند، پردازش‌هــای موازی پایان می‌یابند و خروجی هر فرآیند توسط فرآیند اصلی جمع‌آوری می‌شود.

روال الگوریتم در شــکل ۱ ترسیم شده است. همچنین بخش بعدی الگوریتم که مربوط به تخمین خطای طبقه‌بندی می‌باشد نیز به همین روش موازی شده است. تخمین خطای طبقه‌بنــدی در یک الگوریتم طبقه‌بنــدی و همچنین انتخاب ویژگــی از اهمیت بالایی برخوردار اســت. اگــر از میزان خطای ایجاد شده توســط طبقه‌بندی‌کننده اطلاعی نداشته باشیم نمی‌توانیم عملکرد طبقه‌بندی کننده را بررسی کنیم. در ایــن الگوریتم، خطای طبقه‌بندی به ازای هر فلد برای ٤ ویژگی با امتیاز بالاتر محاســبه می‌شــود. یعنی ابتدا خطا به ازای ویژگی برتر، ســپس دو ویژگی برتر، سه ویژگی



شکل ۱: روندنمای الگوریتم موازی شده

برتر و در نهایت هر چهار ویژگی برتر محاسبه می‌شود. بنابراین با این خطا می‌توانیم زیرنمونه‌های مختلف را مورد سنجش قرار دهیم که کدام نمونه کمترین خطا را در طبقه‌بندی ایجاد کرده است. بخش‌های که به‌صورت موازی اجرا می‌شوند در تصویر با رنگ نارنجی مشخص شده‌اند. در پیاده‌سازی ما به‌دلیل داشتن چهار پردازنده، عمل تخمین خطای طبقه‌بندی نیز به‌صورت موازی انجام می‌شود. به این صورت که هر فلد به یک پردازنده اختصاص یافته و تخمین خطا به ازای هرتعداد ویژگی محاسبه شده و برگردانده می‌شود. بنابراین بار محاسباتی این بخش از الگوریتم نیز در حالت ایده‌آل به تعداد پردازنده‌ها تقسیم می‌شود.

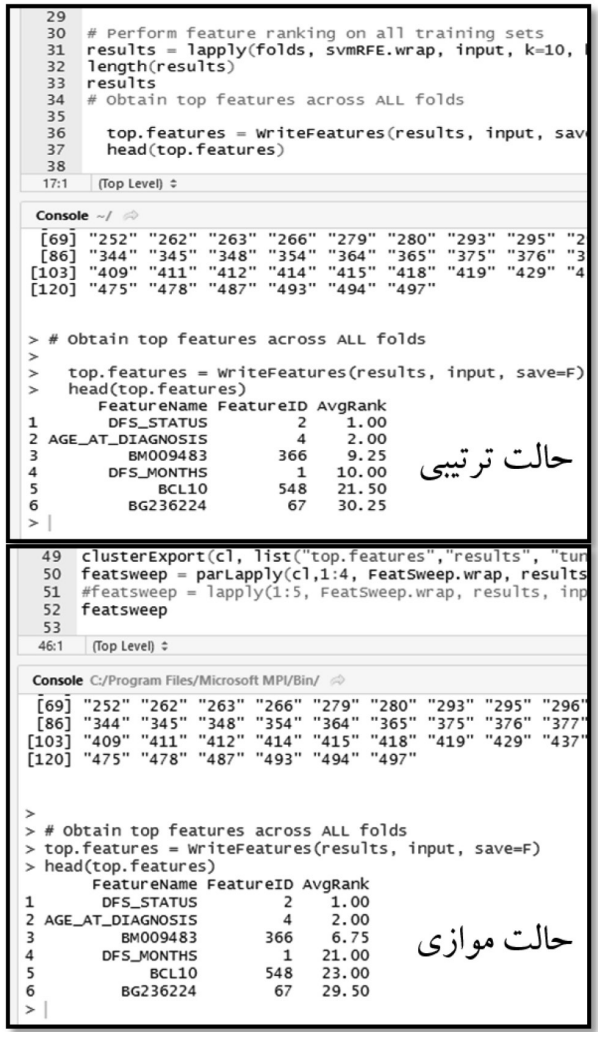
۴- ارزیابی نتایج عملی

پیاده‌سازی الگوریتم موازی پیشنهادی بر روی رایانه‌ای با قابلیت انجام ٤ پردازش همزمان و با هشت گیگا بایت حافظه اصلی بر روی سیستم عامل ویندوز هشت (٦٤ بیت) انجام گرفته است. ابتدا صحت طراحی موازی در قیاس با حالت ترتیبی سنجیده شده است و سپس تاثیر موازی‌سازی در زمان پاسخ مورد ارزیابی قرار گرفته است.

۱،٤. صحت الگوریتم موازی

الگوریتم را با داده شامل ۵۰۰ نمونه و ٦۰۰ ویژگی آزمایش کردیم و تعداد زیرنمونه‌های تولید شده توسط تابع کا-فلد را برابر با تعداد پردازنده‌ها در نظر گرفتیم که در نهایت ٤ زیرنمونه ایجاد شد. خروجی حاصل از حالت ترتیبی و موازی در شکل ۲ قابل مشاهده است. ویژگی‌هایی که در نهایت به‌عنوان نشانگرهای زیستی احتمالی برگردانده شده‌اند در هر دو حالت یکسان بوده و حتی اولویت‌بندی ویژگی‌ها نیز یکسان است و تنها ارزش به‌دست آمده برای ویژگی‌ها در چند مورد متفاوت است.

قابل ذکر است که در بعضی اجراها ممکن است ویژگی‌ها در یک مورد اختلاف داشته باشند که به‌دلیل تغییر در زیرنمونه‌ها به وجود می‌آید. در واقع عدم تطابق بعضی موارد اندک در کد ترتیبی نسبت به کدهای موازی نتیجه دنبالهٔ عدد تصادفی است. زیرا هر چند که مقدار تابع تولید عدد تصادفی یکسان است ولی دنبالهٔ تصادفی در برنامه ترتیبی یک دنبالهٔ پیوسته بوده که در حالت موازی، این دنباله بین پردازش‌ها پخش شده است. بنابراین، الزاماً همان ترتیب دنبالهٔ تصادفی اصلی، بر روی پردازش‌ها رعایت نشده است. به دلیل تایید صحت نتایج بخشی از الگوریتم نیز در هر دو حالت ترتیبی و موازی در تصویر قابل مشاهده است.



شکل ۲: بررسی صحت الگوریتم موازی شده

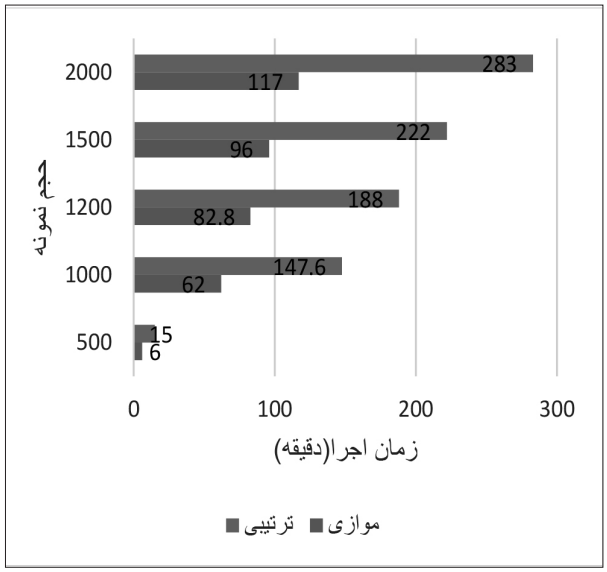
۲،٤. ارزیابی زمان اجرا

زمان اجرا در حالت موازی نسبت به حالت ترتیبی نهایتاً می‌تواند به تعداد پردازنده‌ها کاهش داشته باشدکه البته حالت ایده‌آل است. زیرا زمان اجرا به موارد دیگر از جمله تبادل اطلاعات بین فرآیندها و جمع‌آوری اطلاعات از فرآیندها نیز بستگی دارد. در تمامی اجراهای ترتیبی و موازی، تعداد زیرنمونه‌ها (متغیر kfolds) را برابر با ٤ قرار دادیم زیرا در حالت موازی تنها قادر به انجام ٤ پردازش همزمان است و بدین صورت تقسیم فلدها به‌صورت مناسبت تری صورت می‌گیرد هر چند که برابر نبودن این دو مقدار با یکدیگر مشکلی در روند اجرای الگوریتم ایجاد نخواهد کرد.

نمودار ٤ مقایسه‌ای از اجرای ترتیبی و موازی الگوریتم برای داده باحجم ۵۰۰، ۱۰۰۰، ۱۲۰۰، ۱۵۰۰ و ۲۰۰۰ نشان می‌دهد. رنگ قرمز زمان اجرا در حالت ترتیبی و رنگ آبی، زمان اجرا در حالت موازی را نمایش میدهد. زمان اجرا در حالت موازی نسبت به حالت ترتیبی کاهش چشم‌گیری داشته است. این کاهش، تقریبا ٦۰ درصدِ زمان اجرای ترتیبی است. با وجود این‌که شاهد کاهش زمان اجرای قابل قبولی هستیم امّا علت این‌که شاهد کاهش زمان اجرای بیشتر نبوده‌ایم این است که در این موازی‌سازی، بخش‌هایی از الگوریتم موازی نشده است. در این الگوریتم در صورتی که تعداد ویژگی‌ها بیشتر از ۱۰۰ باشد، ابتدا کاهش حجم ویژگی‌ها برای هر فلد به‌صورت جداگانه انجام می‌شود. به این صورت که الگوریتم ماشین بردار پشتیبان اجرا شده و در هر مرحله تعداد ویژگی‌ها نصف می‌شوند تا تعداد ویژگی‌ها به ۱۰۰ یا کمتر برسد. ویژگی‌هایی در این مرحله حذف می‌شوند که از اولویت کمتری برخوردار هستند. بنابراین این بخش از الگوریتم که خود با توجه به حجم بالای نمونهٔ ما، زمان زیادی را صرف میکند موازی‌سازی نشده است.

در آزمایش بعدی تعداد زیرنمونه‌ها را متفاوت گرفتیم. ابتدا برای ٤ زیرنمونه و سپس ۸ زیرنمونه آزمایش کردیم. در این مقایسه حجم نمونه برابر با ۵۰۰ و تعداد ویژگی‌ها ٦۰۰ مورد است.

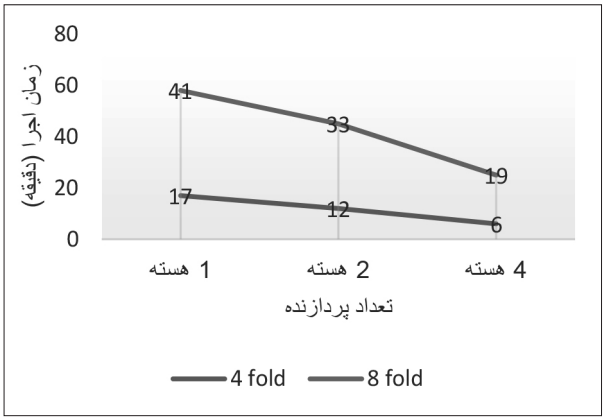
نمودار ۵ مربوط به اجرای برنامه در حالتی است که تعداد زیرنمونه‌ها را ٤ و ۸ در نظرگرفته‌ایم. همان‌طور که مشاهده می‌شود در حالتی که ٤ پردازنده داریم، بهترین زمان اجرا به‌دست آمده است. زیرا به هر پردازنده، تنها یک زیرنمونه اختصاص یافته است و در نتیجه موازی‌سازی بیشتر است. زمانی که به هر پردازنده بیش از یک زیرنمونه اختصاص یابد، هر پردازنده برای هر زیرنمونه، از ابتدا شروع به کاهش ویژگی‌ها تا کمتر از ۱۰۰ می‌کند که خود زمانبر است. علاوه بر این، هر پردازنده باید ماشین‌های بیشتر را به دلیل داشتن زیرنمونه‌های بیشتر اجرا کند.



نمودار ۴: ارزیابی زمان اجرا

با مقایسه زمان اجرا در حالت یک هسته‌ای با حالتی که ٤ هسته در اختیار داریم، درمی‌یابیم که سرعت اجرای الگوریتم بیش از دو برابر افزایش یافته است. هر چند که با وجود ٤ هسته، انتظار داریم که زمان اجرا نزدیک به ٤ برابر افزایش یابد امّا در صورتی که تعداد زیرنمونه‌ها از تعداد پردازنده‌ها بیشتر باشد، نمی‌توانیم در یک مرحله، زیرنمونه‌ها را به پردازنده‌ها اختصاص دهیم. پس در صورتی که ۸ زیرنمونه داشته باشیم و ۲ پردازنده، باید در طی ٤ مرحله زیرنمونه‌ها را به پردازنده‌ها اختصاص دهیم که زمانبر است. علاوه بر این پس از اتمام کار پردازنده‌ها، ارزش ویژگی‌ها باید از پردازنده‌های مختلف جمع‌آوری شده و سپس در پردازنده اصلی این نتایج برای هر ویژگی ادغام و ارزیابی شود.

همان‌طور که در نمودار ٥ مشاهده می‌شود، با افزایش تعداد زیرنمونه‌ها، زمان اجرا به‌طور قابل ملاحظه‌ای افزایش می‌یابد. امّا نباید فراموش کنیم که علت ایجاد زیرنمونهٔ بیشتر، درواقع افزایش دقت الگوریتم در شناسایی صحیح نشانگرهای زیستی است. بنابراین باید میان افزایش تعداد زیرنمونه‌ها و زمان اجرا تعادلی برقرار کنیم. همان‌طور که از نمودارها برمی‌آید زمانی که تعداد زیرنمونه‌ها برابر با تعداد پردازنده باشد، بهترین تعادل برقرار می‌شود. زیرا هم



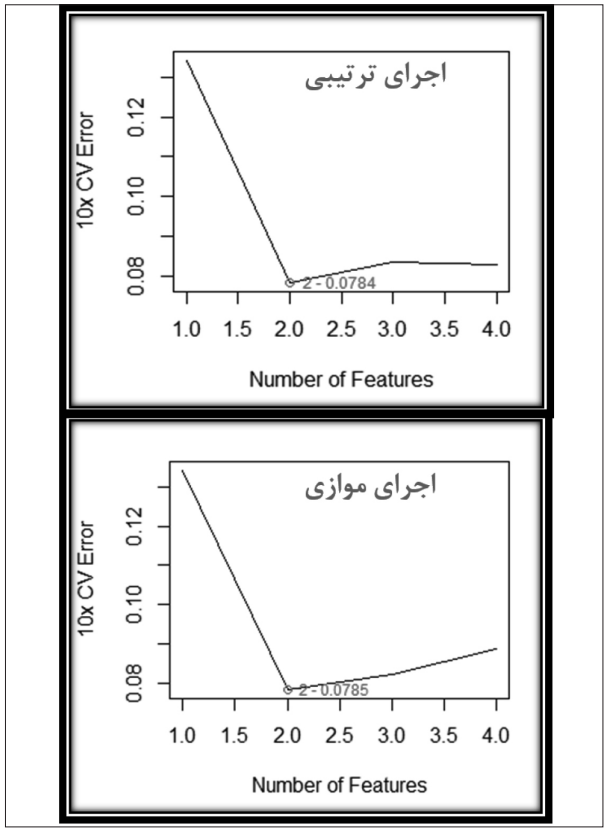
نمودار ۵: مقایسه زمان اجرا در تعداد زیرنمونه‌های مختلف

به افزایش دقت الگوریتم توجه داشته‌ایم و هم موازی‌سازی به‌طورکامل انجام گرفته و در نتیجه زمان اجرا کاهش یافته است.

۳،٤. ارزیابی دقت الگوریتم

الگوریتم در نهایت، خطای دسته‌بندی را برای ویژگی‌های برتر محاسبه می‌کند. برای تخمین خطای طبقه‌بندی از تخمین‌گر کراس‌ولیدیشن[۲۲] استفاده شده است. این خطا به ازای هر فُلد برای هر ویژگی محاسبه می‌شود. تخمین خطای طبقه‌بندی در نهایت می‌تواند نشان دهد که کدام زیرنمونه خطای کمتری داشته و برای دسته‌بندی مناسب تر است[۱٥]. همچنین با مقایسه خطای طبقه‌بندی در حالت موازی نسبت به ترتیبی می‌توانیم نشان دهیم که دقت الگوریتم چه تغییری داشته است. در الگوریتم مورد نظر، این خطا به ازای داده با حجم نمونه ٥۰۰ و تعداد ویژگی ٦۰۰ ارزیابی شده است. خطا به ازای هر چهار زیرنمونه و برای ٤ ویژگی برتر به‌صورت موازی شده محاسبه شده است یعنی خطا به ازای چهار زیرنمونه ابتدا برای اولین ویژگی برتر، سپس دو ویژگی برتر، سه ویژگی برتر و در نهایت هر چهار ویژگی برتر محاسبه می‌شود که این چهار عمل به‌صورت موازی بر روی پردازنده‌ها اجرا می‌گردد. در هر مرحله، میانگین خطاهای حاصل از زیرنمونه‌های مختلف برآورد شده و به‌عنوان خطای طبقه‌بندی در نظر گرفته می‌شود.

22- Cross-Validation



جدول ۱
شکل ۳: مقایسه تخمین خطای طبقه‌بندی

شکل ۳ خطا را در حالت ترتیبی و موازی نشان می‌دهد. عدد قرمز رنگ نشـــان می‌دهد که کمتریـــن خطا برای چه تعداد از ویژگی‌ها حاصل شده است. در هر دو نمودار عدد ۲ نماینگر این اســـت که خطا بـــرای دو ویژگی برتر (یعنی ویژگی اول و دوم) کمترین میانگین خطا به‌دست آمده است و این خطا در حالت ترتیبی برابر با ۰/۰۷۸٤ و برای حالت موازی ۰/۰۷۸۵ برآورد شده اســـت. مشاهده می‌شود که میانگین خطای حاصل در دو حالت ترتیبی و موازی بسیار به هم نزدیک است. این بدین معناست که موازی‌سازی دقت الگوریتم را کاهش نداده است.

۵. پژوهش‌های مرتبط

الگوریتـــم اس.وی.ام-آر.اف.ای به‌دلیـــل اهمیتـــی که به‌عنوان یک الگوریتم انتخاب ویژگی در داده‌های ژنی دارد همواره مورد توجه پژوهشـــگران بوده و به‌دلیل داشـــتن پیچیدگی بالا، بحث موازی‌ســـازی در زمان استفاده از این الگوریتم همواره مطرح بوده است. در سال ۲۰۱۸، داس و همکارنش بسته نرم‌افزاری برای نرم‌افزار آر به نام سیگ فیچر[۲۳] ارائه کردند که قادر است ویژگی‌های قابل توجه و مهم[۲٤] را با اســـتفاده از الگوریتم ام.اس.وی.ام-آر.اف.ای و الگوریتم تی-استاتیک[۲۵] بیابد. در الگوریتم اس.وی.ام-آر.اف.ای بـــرای کاهش پیچیدگی‌ها، بخصـــوص زمانی که با داده حجیم ســـروکار داریم، از بســـته نرم‌افزاری پارالل[۲٦] برای موازی‌ســـازی بخش‌هایـــی از الگوریتم کـــه قابلیت موازی‌ســـازی دارند، استفاده شده اســـت[۱۸]. این بسته نرم‌افزاری از نظر انعطاف در طراحی موازی‌سازی نسبت به بسته نرم‌افزاری آر.ام.پی.آی، محدودتر بوده امّا استفاده از آن به‌دلیل داشتن دستورات مشخص برای موازی‌سازی ساده‌تر است.

در سال ۲۰۱٦، فرانکو و همکارانش پیاده‌سازی موازی از چندین الگوریتم از جمله الگوریتم اس.وی.ام-آر.اف.ای ارائه کردند. پیاده‌سازی ارائه شده برای الگوریتم اس.وی.ام-آر.اف.ای بـــرای ماشـــین دودویی و چنـــد رده‌ای قابل اجراســـت به این صورت که فرآیند رتبـــه بندی ویژگی‌ها توســـط هر رده، به‌صورت موازی پیاده‌سازی شده است، و دیگر مراحل الگوریتم به‌صورت ترتیبی اجرا می‌شود. در این پیاده‌سازی از اســـپارک[۲۷] برای موازی‌سازی استفاده شـــده اســـت و بســـته نرم‌افزاری ام.ال.لیب[۲۸] بدین منظور به‌کار گرفته شـــده است. بســـته نرم‌افزاری ام.ال.لیب، یکی از بســـته‌های نرم‌افزاری آپاچی اســـپارک اســـت که برای پردازش الگوریتم‌های یادگیری ماشین و داده‌کاوی مورد اســـتفاده قرار می‌گیرد. در این پیاده‌ســـازی، نرخ افزایش سرعت در بهترین حالت (تعداد نمونه‌ها نسبت به ویژگی‌ها بیشتر باشد) ۱,٤ به‌دست آمده است که نسبت به الگوریتم پیشنهادی این مقاله مقدار کمتری است [۱۹].

همچنین در سال ۲۰۱٤، هانگ‌یو و همکارانش، نسخه‌ای

23-SigFeature
24- Significant Features
25- t-Statis
26- parallel package (library(parallel))
27- Spark
28- MLlib

موازی شـــده از ماشین بردار پشـــتیبان ارائه کردند که با استفاده از فناوری نگاشت-کاهش[29] موازی شده بود. این فناوری برای داده توزیع شده مورد استفاده قرار می‌گیرد. این پیاده‌ســـازی برای بررســـی تعامل پروتئین-پروتئین مورد اســـتفاده قرار گرفت که در نهایـــت الگوریتم موازی شده نسبت به الگوریتم اصلی در بهترین حالت ۱٫۸ افزایش سرعت داشت[۲۰].

اگر بخواهیم مقایســـه‌ای از پژوهش‌های اشاره شده با الگوریتم مورد نظرمان داشته باشیم می‌توان به نکات زیر اشاره کرد:

۱) الگوریتـــم موازی شـــده در این مقالـــه، دقیقا همان الگوریتم موازی شـــده در دیگر پژوهش‌های معرفی شده نمی‌باشد. در پژوهش‌های اشاره شده از الگوریتم اس.وی.ام-آر.اف.ای اســـتفاده شـــده امّـــا در این مقاله از نســـخه پیچیده‌تر ایـــن الگوریتـــم یعنـــی ام.اس.وی.ام-آر.اف.ای استفاده شده است.

۲) پژوهش‌های اشـــاره شـــده و همچنین پژوهش ما، همگی در حوزه بیوانفورماتیک جای گرفته‌اند چون اساسا کاربرد الگوریتم اس.وی.ام-آر.اف.ای در این حوزه اســـت و هدف، کاهـــش پیچیدگی الگوریتم برای حل یک مســـئله بیوانفورماتیکی بوده اســـت. بنابراین داده مورد اســـتفاده در ایـــن پژوهش‌ها همگی از نوع داده زیســـتی بوده‌اند، با این حال حجم و نوع داده مورد استفاده در این پژوهش‌ها متفاوت اســـت که خـــود نقش تعیین کننده‌ای در ســـرعت الگوریتم‌ها دارد.

۳) در بعضی از پژوهش‌های مورد اشاره همانند پژوهش ما از محیط آر و بســـته‌های آن برای موازی‌سازی استفاده شده است و در بعضی دیگر، از فناوری‌ها مانند اوپن.ام.پی. آی، اسپارک و ... استفاده شده است. بنابراین فناوری مورد استفاده برای موازی‌سازی کاملا یکسان نیست.

٤) بعضی از پژوهش‌های مورد اشاره، در سیستم های توزیع شده انجام گرفته است حال آن که در این پژوهش ما با یک سیستم یکپارچه روبرو هستیم.

29- Map-Reduce

مقاله	الگوریتم مورد استفاده	تکنولوژی موازی‌سازی	افزایش سرعت
پژوهش ما	ام.اس.وی.ام-آر.اف.ای	آر.ام.پی.ای	۲٫۸
داس [۱۸]	اس.وی.ام-آر.اف.ای	پکیج پارالل	-
فرانکو[۱۹]	اس.وی.ام-آر.اف.ای	اسپارک	۱٫۴
هانگیو[۲۰]	ماشین بردار پشتیبان	نگاشت کاهش	۱٫۸

بنا به نکات ذکر شـــده، هرچند مقایســـهٔ میزانِ افزایش ســـرعتِ به‌دســـت آمده در پژوهش‌های مختلـــف، چندان صحیح به نظر نمی‌رسد با این حال الگوریتم مورد نظر ما در بهترین حالت با افزایش سرعتی در حدود ۲٫۸، توانسته اســـت افزایش سرعت بالاتری نســـبت به دیگر الگوریتم‌ها کسب کند (جدول۱).

۶. نتیجه‌گیری

در نمودار ۳ نرخ افزایش سرعت حاصل از موازی‌سازی مشاهده می‌شـــود. این بررسی روی دو رایانه با قابلیت ۲ و ٤ پردازش همزمان اجرا شـــده و در تمامی حالات، تعداد زیرنمونه‌ها نیز ٤ می‌باشـــد. این بررســـی بر روی داده با تعداد نمونه‌های مختلف اجرا شـــده اســـت. همان‌طور که مشاهده می‌شود نرخ سرعت ٤ فرآیند به یک فرآیند، نسبت بـــه دو فرآیند به یک فرآیند، رشـــد بهتـــری دارد. علت این است که در حالتی که ٤ هســـته داریم به هر پردازنده تنها یک نمونه تخصیص یافته امّا در حالت ۲ هســـته‌ای، به هر پردازنده دو زیرنمونه اختصاص می‌یابد و در نتیجه زمان بیشتری را از پردازنده می‌گیرد.

بنابراین بـــا افزایش تعداد پردازنده‌ها، شـــاهد افزایش ســـرعت بیشتری هســـتیم. در رایانه‌ای با ٤ پردازنده، نرخ افزایش ســـرعت حدود دو و نیم است. بهترین نرخ افزایش ســـرعت در هر دو حالت در داده ۵۰۰ اتفاق افتاده اســـت کـــه در حالت ٤ پردازنـــده‌ای این افزایش ســـرعت چیزی حدود ۲/۸ برابر بوده اســـت. و در حالـــت دو پردازنده ای حدود ۱/۹۰ است. می‌توان گفت در این حالت بهترین توزیع داده صورت گرفته اســـت که متناسب با تعداد پردازنده‌ها

نمودار ۶: بررسی نرخ افزایش سرعت

و زیرنمونه‌هـــا و همچنین حجم زیرنمونه‌ها بوده اســـت. هرچند که نمی‌توان انتظار داشت با چهار برابر شدن تعداد پردازنده‌ها، ســـرعت پردازش نیز افزایـــش چهار برابری داشته باشد امّا باید توجه داشت که درصد کمی از الگوریتم نیز قابل موازی‌ســـازی نبوده اســـت که علـــت تفاوت نرخ به‌دســـت آمده با حالت ایده آل است. به‌طور کلی در هر دو نمودار بیشتر از یک و نیم برابرِ تعداد پردازنده‌ها، افزایش سرعت داریم که با افزایش تعداد پردازنده‌ها و کاهش حجم بار محاسباتی هر پردازنده، شاهد افزایش سرعت بیشتری خواهیم بود. همچنین زمان اجرای الگوریتم موازی همواره نسبت به حالت ترتیبی کاهش چشمگیری داشته است.

مراجع

[1]A. Jemal et al., “Cancer Statistics,” CA. Cancer J. Clin., vol. 58, no. 2, pp. 71–96, Jan. 2008.

[2]D. Wu, C. M. Rice, and X. Wang, “Cancer bioinformatics: A new approach to systems clinical medicine,” BMC Bioinformatics, vol. 2, no. 1, p. 71, 2012.

[3]H. Moses and S. Nass, Cancer biomarkers: the promises and challenges of improving detection and treatment. 2007.

[4]G. Joshi, R. Kaur, and H. Kaur, “Biomarkers in cancer,” 2016.

[5]N. Dessì, E. Pascariello, B. P.-B. research international, “A comparative analysis of biomarker selection techniques,” hindawi.com, 2013.

[6]H. Liu and H. Motoda, Computational methods of feature selection, Chapman & Hall Crc Data Mining and Knowledge Discovery Series 2007.

[7]Quo CF, Kaddi C, Phan JH, et al. Reverse engineering biomolecular systems using -omic data: challenges, progress and opportunities. Brief Bioinform. 2012;13(4):430–445.

[8]Y. Saeys, I. Inza, and P. Larranaga, “A review of feature selection techniques in bioinformatics,” Bioinformatics, vol. 23, no. 19, pp. 2507–2167, Oct. 2007.

[9]Z. He, W. Y.-C. biology and chemistry, “Stable feature selection for biomarker discovery,” Computational biology and chemistry, 2010.

[10]Isabelle Guyon , Steve Gunn , Masoud Nikravesh , Lotfi A. Zadeh, Feature Extraction: Foundations and Applications (Studies in Fuzziness and Soft Computing), Springer-Verlag New York, Inc., Secaucus, NJ, 2006.

[11]Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. Gene selection for cancer classification using Support Vector Machines. Machine Learning, 2002, 46(1-3), 389-422.

[12]E. Hemphill, J. Lindsay, C. Lee, I. I. Măndoiu, and C. E. Nelson, “Feature selection and classifier performance on diverse bio- logical datasets,” BMC Bioinformatics, vol. 4, no. Suppl 2, p. S4, 2013.

[13]Yingbo Zhou , Utkarsh Porwal , Ce Zhang , Hung Ngo , XuanLong Nguyen , Christopher Ré , Venu Govindaraju, Parallel feature selection inspired by group testing, Proceedings of the 27th International Conference on Neural Information Processing Systems, p.3554-3562, December 08-13, 2014, Montreal, Canada

[14] Tomczak K, Czerwińska P, Wiznerowicz M. The Cancer Genome Atlas (TCGA): an immeasurable source of knowledge. Contemp Oncol (Pozn). 2015.

[15]Kai-Bo Duan, J. C. Rajapakse, Haiying Wang and F. Azuaje, “Multiple SVM-RFE for gene selection in cancer classification with expression data,” in IEEE Transactions on NanoBioscience, vol. 4, no. 3, pp. 228-234, Sept. 2005.

[16]john colby, “(multiple) Support Vector Machine Recursive Feature Elimination - mSVM-RFE [ColbyImaging].” [Online]. Available: http://www.colbyimaging.com/wiki/statistics/msvm-rfe. [Accessed: 25-May-2018].

[17]S. K. Mathur, Statistical Bioinformatics : with R, Academic Press, California, 2010.

[18] Das, Pijush, Susanta Roychudhury, and Sucheta Tripathy. “sigFeature: Significant feature selection using SVM-RFE & t-statistic.” (2018).

[19] C. Eiras-Franco, V. Bolón-Canedo, S. Ramos, J. González-Domínguez, A. Alonso-Betanzos, J. Touriño
Multithreaded and spark parallelization of feature selection filters J. Comput. Sci., 17 (2016), pp. 609-619.

[20] You, Zhu-Hong, et al. “A MapReduce based parallel SVM for large-scale predicting protein–protein interactions.” Neurocomputing 145 (2014): 37-43.