بسم الله الرحمن
الرحيم

دانشکده فنی و مهندسی

گروه مهندسی کامپیوتر
گرایش:نرم افزار

عنوان:

پیش بینی انواع لوسمی مبتنی بر ترکیب مدل ها با استفاده از الگوریتم ژنتیک

استاد راهنما:
جناب آقای دکتر عباس عکاسی

محققین:
کمیل آقابابایی، خاطره اصغری، امیر جعفری زاده، نوید چمنی، مونا رجبی فرد، مژگان زرافشان، سهیلا سالاری، مرتضی‌سهرابی، محبوبه صادقی، سارا عباس زاده، سعید ناصری، مریم یوسفی زاده

آذر ماه 1397

با تشکر فراوان از جناب آقای دکتر عباس عکاسی که مارا یاری کردند.

تقدیم به او که جهان در انتظارش است.

تقدیم به او که منجی عالم بشریت است.

اللهم عجل لولیک فرج ...

چکیده

لوسمی یکی از شایع ترین علل مرگ و میر در سراسر جهان است. پیش بینی دقیق نوع لوسمی برای استفاده از داروهای خاص مهم است. از این رو پژوهش در این زمینه ادامه دارد. تحلیل داده های ریز آرایه[1] یک روش کارآمد برای پیش بینی نوع لوسمی است، که بدون استفاده از الگوریتم های داده کاوی دشوار است. هدف این مقاله طبقه بندی[2] انواع لوسمی لنفوبلاستی حاد[3] و لوسمی میلوئید[4] حاد با استفاده از ترکیب 100طبقه بند است. در این مطالعه، توصیفی از داده های بیان 7129 ژن مربوط به 72 بیمار مبتلا به لوسمی استفاده شد. سپس یک الگوریتم از بین الگوریتم های ماشین بردار پشتیبان[5]، الگوریتم بیزین[6] و الگوریتم درخت تصمیم[7] بر اساس رای گیری اکثریت[8]، به عنوان طبقه بند پایه انتخاب شد. با الگوریتم طبقه بند پایه 100مدل ساخته شد. ترکیب 100مدل با استفاده از الگوریتم ژنتیک[9] انجام شد. نتایج حاصل از مقایسه ی روش پیشنهادی با سایر روش های ترکیب، عملکرد خوب الگوریتم زنتیک را نشان داد.

کلمه کلیدی: سرطان لوسمی، داده کاوی، الگوریتم ژنتیک، طبقه بندی، ارزیابی، یادگیری تجمعی

[1] Microarray
[2] Classification
[3] Acute lymphoblastic
[4] Acute myeloid
[5] Support vector machine
[6] Naïve byes algorithm
[7] Decision tree algorithm
[8] Majority voting
[9] Genetic algorithm

فهرست مطالب

عنوان
صفحه

فهرست شکل‌ها

عنوان
صفحه

فهرست جداول

عنوان
صفحه

فصل اول
کلیات تحقیق و
اصول کلی

1-1- مقدمه

طبقه بندی انواع لوسمی طی 30 سال گذشته بهبود زیادی یافته است و تلاش های بسیار زیادی در این زمینه انجام شده است، اما روش های کمی برای اختصاص دادن تومور به کلاس‌های شناخته شده (پیش بینی کلاس) وجود دارد. برای غلبه بر این چالش و رسیدن به راه حلی برای طبقه بندی لوسمی، استفاده از طبقه بندی لوسمی صرفا بر پایه بیان ژن می باشد. این روش یک استراتژی کلی برای کشف و پیش بینی لوسمی است و می تواند برای انواع دیگر سرطان، مستقل از دانش بیولوژیکی قبلی مورد استفاده قرار بگیرد.[1]

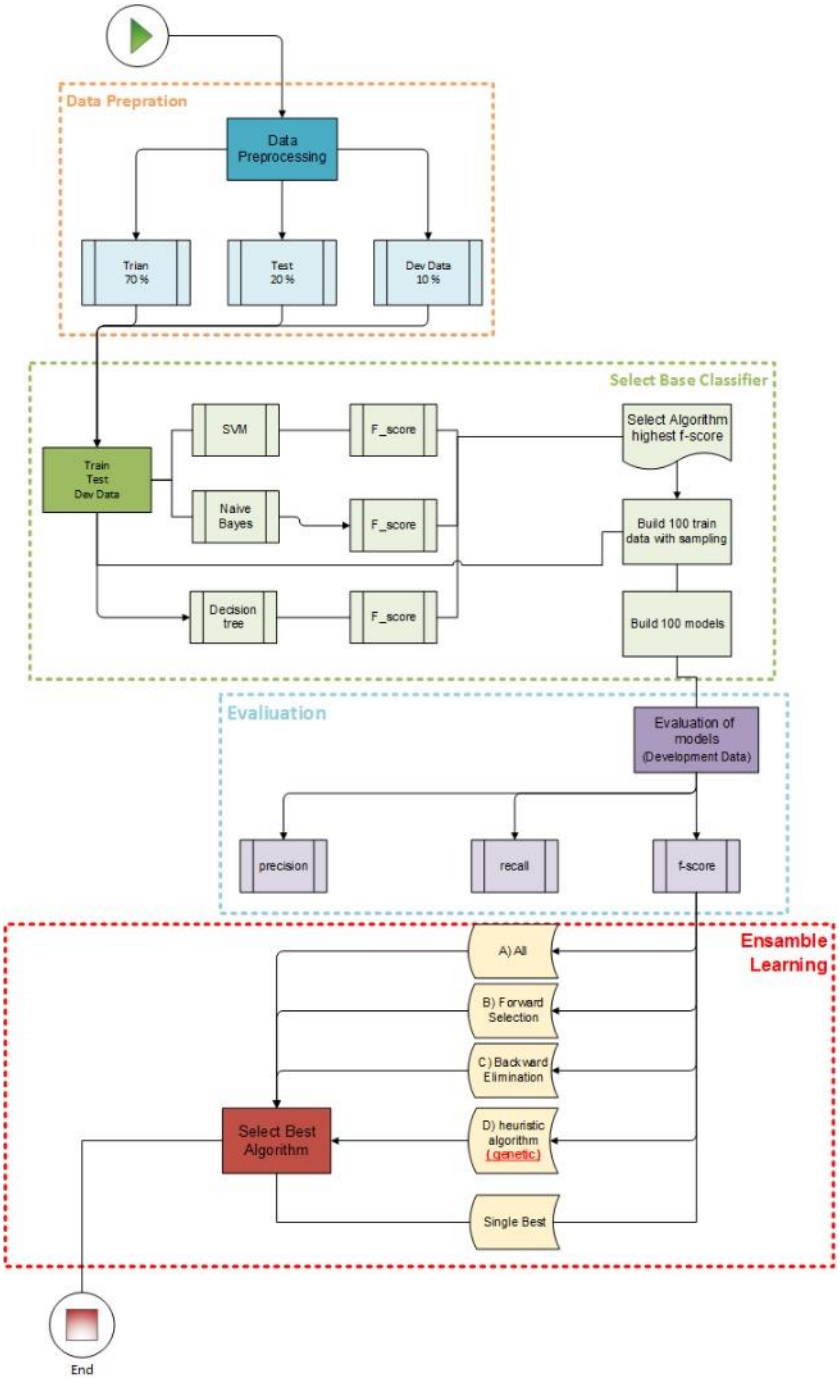
تکنولوژی ریز آرایه باعث می شود زیست شناسان قادر به نظارت بر بیان هزاران ژن در یک آزمایش واحد، در یک تراشه کوچک باشند. داده های مجموعه ای از بیان ژن ریز آرایه در اینترنت در دسترس عموم هستند. این مجموعه داده ها شامل تعداد زیادی از مقادیر بیان ژن هستند ، از این رو نیاز به یک روش دقیق برای استخراج دانش و اطلاعات مفید از مجموعه داده های بیان ژن احساس می شود. [2]

نحوه طبقه بندی مهمترین چالش بعد از انتخاب نوع داده است. در این زمینه نیز تاکنون تلاش های بسیار زیادی انجام شده است که از جمله آن‌ها می توان به دسته بندی لوسمی به وسیله الگوریتم SVM[3]، الگوریتم درخت تصمیم[4] والگوریتم بیزین[5] اشاره کرد. مهمترین مشکل استفاده از این روش‌ها پایین بودن قابلیت اطمینان آنهاست.

در این مطالعه برای بالابردن قابلیت اطمینان بجای استفاده از یک داده آموزش و یک طبقه بند از 100داده ی آموزش و 100طبقه بند استفاده شد، برای انتخاب بهترین زیر مجموعه از بین طبقه بند ها از الگوریتم ژنتیک استفاده شد.

در ادامه به بررسی روش پیشنهادی و در فصل دوم نتایج و در آخر فهرست منابع آورده می شود.

2-1- روش پیشنهادی



شکل 1-1- گردش کار روش پیشنهادی

با توجه به شکل 1-1، روش پیشنهادی به چهار بخش تقسیم گردید، در هر بخش ابتدا مفاهیم مورد استفاده تعریف می شود، سپس روش کار بیان می شود.

3-1- آماده سازی داده

- **مجموعه داده[1]**

مجوعه داده leukemia از معتبرترین و استانداردترین مجوعه داده های معرفی شده در زمینه تشخیص نوع سرطان می باشد که در پژوهش های بسیاری مورد استفاده قرار گرفته است. این مجوعه داده یک رویکرد عمومی برای طبقه بندی سرطان خون بر اساس نظارت بر ژن توسط Microarrays DNA توصیف می کند. این مجوعه داده دارای 72 نمونه می باشد که از این تعداد، 47 نمونه مربوط به نوع لوسمی حاد لنفوبلاستیکی (ALL) و 25 نمونه مربوط به لوسمی حاد میلوئید (ALL) می باشد. ساختار دیتاست متفاوت با ساختار دیتاست های مرسوم می باشد و هر نمونه به جای سطری به صورت ستونی ذخیره شده اند و برچسب تمام نمونه ها در سطر اول ذکر شده است. حداکثر تعداد ویژگی های هر نمونه برابر با 7128 می باشد که در نهایت یک ماتریس 7128*72 را ایجاد می‌کند.[6]

جدول 1-1- تعداد نمونه ها و ویژگی های انواع لوسمی

AML	ALL	
25	47	تعداد نمونه
7128	7128	تعداد ویژگی

- **پیش پردازش مجموعه داده**

ابتدا مجموعه داده مسئله فراخوانی شد، سپس جداسازی مجموعه داده به بخش های داده های آموزش[2]، داده های آزمون[3] و داده های توسعه[4] به ترتیب با اندازه های 70% ،20% و 10% گرفت. این بخش ها با رعایت توزیع کلاس انتخاب شد، به این صورت که برای هر بخش به میزان مساوی از هر کلاس انتخاب شد.

4-1- انتخاب الگوریتم پایه و ساخت 100 طبقه بند با استفاده از آن

از الگوریتم های طبقه بندی درخت تصمیم، ماشین بردار پشتیبان و الگوریتم بیزین استفاده شد.

1-4-1- الگوریتم درخت تصمیم

این الگوریتم با طرح این سوال که چه صفتی باید در ریشه درخت آزمایش شود آغاز می شود. برای پاسخ به این سوال، با استفاده از یکی از انواع آزمایش های آماری برای تعیین مناسب ترین صفت برای دسته بندی مثال های آموزشی، تصمیم براساس هر صفت نمونه را ارزیابی می کند. سپس بهترین صفت را انتخاب کرده و به عنوان تست در گرهٔ ریشهٔ درخت استفاده می کند. برای هر مقدار ممکن صفت تست شده در ریشه، یک گره متناظر ایجاد شده و مثال های آموزشی براساس مقادیر صفت تست، بین این گره ها افراز می شوند. تمام فرایند ذکر شده،

[1] Data set
[2] Train data
[3] Test data
[4] Development data

با استفاده از مثال های آموزشی نسبت داده شده به هر گره، برای انتخاب بهترین صفت در گره های درخت تکرار می شود. این روش جستجوی حریصانه را برای یک درخت تصمیم قابل قبول ارائه می دهد که در این الگوریتم، هیچ گاه برای در نظر گرفتن دوباره انتخاب های قبلی، به عقب برگشت نمی شود.

درخت تصمیم که هدف اصلی آن، دسته‌بندی داده‌هاست، مدلی در داده‌کاوی است که ساختاری درخت مانند را جهت اخذ تصمیم و تعیین کلاس و دسته یک داده خاص به ما ارائه می‌کند. همان‌طور که از نام آن مشخص است، این درخت از تعدادی گره و شاخه تشکیل‌شده است به‌گونه‌ای که برگ‌ها کلاس‌ها یا دسته‌بندی‌ها را نشان می‌دهند و گره‌های میانی هم برای تصمیم‌گیری با توجه به یک یا چند صفت خاصه به‌کارمی‌روند [7].

1-4-2- الگوریتم ماشین بردار پشتیبان

این الگوریتم یک جداکننده است که با معیار قرار دادن بردارهای پشتیبان، بهترین دسته بندی و تفکیک بین داده ها را برای ما مشخص می کند .بردارهای پشتیبان به زبان ساده، مجموعه ای از نقاط در فضای n بعدی داده ها هستند که مرز دسته ها را مشخص می کنند و مرزبندی و دسته بندی داده ها براساس آنها انجام می شود و با جابجایی یکی از آن ها، خروجی دسته بندی ممکن است تغییر کند. در فضای دو بعدی بردارهای پشتیبان، یک خط، در فضای سه بعدی یک صفحه و در فضای n بعدی یک ابر صفحه را شکل خواهند داد. در این الگوریتم فقط داده های قرار گرفته در بردارهای پشتیبان مبنای یادگیری ماشین و ساخت مدل قرار می گیرند و این الگوریتم به سایر نقاط داده حساس نیست و هدف آن هم یافتن بهترین مرز در بین داده هاست به گونه ای که بیشترین فاصله ممکن را از تمام دسته ها (بردارهای پشتیبان آنها) داشته باشد [8].

1-4-3- الگوریتم بیزین

تئوری بیز یکی از روش های آماری برای رده بندی به شمار می آید. در این روش کلاس های مختلف، هر کدام به شکل یك فرضیه دارای احتمال در نظر گرفته می شوند. هر رکورد آموزشی جدید، احتمال درست بودن فرضیه های پیشین را افزایش و یا کاهش می دهد و در نهایت، فرضیاتی که دارای بالاترین احتمال شوند، به عنوان یك کلاس در نظر گرفته شده و برچسبی بر آن ها زده می شود. این تکنیك با ترکیب تئوری بیز و رابطه سببی بین داده ها، به طبقه بندی می پردازد [9].

1-5- روش کار

این بخش به سه قسمت تقسیم شد:

- هر سه الگوریتم با داده های آموزشی، آموزش داده شد، سپس طبقه بند به دست آمده توسط داده های توسعه ارزیابی شد و برای هر الگوریتم F_score ،precision و recall محاسبه شد، در نهایت الگوریتمی که دارای بهترین F_score بود ، به عنوان الگوریتم پایه انتخاب شد.
- 100مجموعه داده آموزشی متفاوت به صورت رندوم با استفاده از نمونه برداری و با جایگذاری، ساخته شد. .

- 100 مجموعه داده آموزشی به الگوریتم پایه داده شد و 100مدل از آنها ساخته شد.

6-1- ارزیابی مدل ها

100مدل ساخته شده در بخش قبلی روی داده های توسعه ارزیابی شد ، برای هر مدل F_score , precision و recall محاسبه شد .مدل ها در یک آرایه بر منای F_score به صورت صعودی مرتب شد و بهترین مدل[1] انتخاب شد.

7-1- یادگیری تجمعی [2]

یادگیری تجمعی ترکیبی از چندین مدل برای تولید یک مدل پیش بینی مطلوب است. [10]

این بخش شامل دو مرحله می شود :

1-7-1- انتخاب طبقه بندها

در یادگیری تجمعی فرض می کنیم استخری از طبقه بندها را داریم و می خواهیم زیر مجموعه ای از آن ها را انتخاب کنیم به گونه ای که ترکیب اعضای این زیرمجموعه بهترین نتیجه را بدهد.در این بخش ارزیابی روی داده های توسعه و داده های آزمایش مورد ارزیابی قرار گرفت.

2-7-1-ترکیب طبقه بندها

برای ترکیب طبقه بندها روش های متفاوتی وجود دارد ،در این مطالعه روش های ترکیب زیر مورد بررسی قرار داده شد.

1-2-7-1- ترکیب همه ی طبقه بندها

در این نوع ترکیب برای هر نمونه با توجه به صورت مسئله تعدادی طبقه بند وجود دارد که با توجه به خروجی هر طبقه بند، خروجی که بیشترین تکرار را داشته باشد به عنوان خروجی نهایی در نظر گرفته می شود.

به طور مثال 5 نمونه و 4 طبقه بند داریم که نتیجه را به صورت رای اکثریت نشان می دهد و در مواردی که تعداد هر دو کلاس با هم برابر باشد به صورت رندم یکی را به عنوان خروجی نهایی در نظر می گیرد.

جدول 1-2- ترکیب مدل ها برمبنای رای گیری اکثریت

نمونه	E1	E2	E3	E4	Out
1	A	A	B	A	A
2	A	B	B	B	B
3	A	A	B	B	A
4	A	B	A	A	A
5	B	B	A	B	B

[1] Single best
[2] Ensemble learning

1-7-2-2- انتخاب رو به جلو[1]

انتخاب رو به جلو یک نوع رگرسیون گام به گام است که با یک مدل خالی شروع می شود و طبقه بندهایی را که به صورت نزولی بر مبنای یکی از معیارهای ارزیابی مرتب شده اند را یک به یک اضافه می کند و بهترین متغیر ، با برخی از معیارهای از پیش تعیین شده تعیین می شود و به مدل اضافه می گردد. معیار استفاده شده برای تعیین اینکه کدام مدل اضافه شود ، متفاوت است. در هر مرحله رو به جلو، شما یک طبقه بند را اضافه می کنید که بهترین نتیجه را برای شما به ارمغان می آورد این یکی از دو روش معمول استفاده از رگرسیون گام به گام است. تغییر بیش از حد زمانی اتفاق می افتد که ما متغیرهای بیشتری را در اختیارمان قرار دهیم تا اینکه برای مدل مناسب باشد. به طور معمول یک اطلاعات بسیار دقیق از داده های مورد استفاده در رگرسیون را نشان می دهد، اما این مدل از نقاط داده های اضافی دور خواهد ماند و برای اینترلوسیون خوب نیست. [11]

- در این روش مدل ها بر مبنای F_score به صورت نزولی مرتب شد، در ابتدا اولین مدل مورد ارزیابی قرار گرفت و سپس مدل دوم با مدل اول ترکیب شد و ترکیب این دو، مورد ارزیابی قرار گرفت اگر f-score ترکیب این دو مدل بهتر از مدل اول به تنهایی باشد که ترکیب دو مدل را ذخیره می کنیم و مدل بعد را اضافه می کنیم در غیر این صورت مدل دوم حذف می شود و سومین مدل ترکیب می شود ،به همین ترتیب تا آخرین مدل (100 مدل) انتخاب گردید .

با فرض داشتن سه مدل E1,E2,E3 که به صورت نزولی بر مبنای F_score مرتب شده اند جدول 2 و 3 را بدست آوردیم.

جدول 1-3- عملکرد روش انتخاب رو به جلو

Best F_score	Best classifier
F(E2)	E2
F(E2+E1)	E2+E1

IF F(E2+E1)> F(E2) Select F(E2+E1) ,else remove E1

1-7-2-3- حذف رو به عقب[2]

در این روش ابتدا مدل ها را برمبنای F_score به صورت صعودی مرتب کردیم سپس همه ی مدل ها با هم ترکیب شد .ترکیب مدل ها ارزیابی شد و اولین مدل را حذف کردیم و نتیجه را مورد ارزیابی قرار دادیم به همین ترتیب تا آخرین مدل ادامه دادیم .[11]

[1] Forward selection
[2] Backward elimination

با فرض داشتن دو مدل,E1,E2 که به صورت صعودی بر مبنای F_score مرتب شده اند جدول 4 را بدست آوردیم.

جدول 1-4- عملکرد روش حذف رو به عقب

Best F_score	Best classifier
F(E2+E1)	E2+E1
F(E1)	E1

remove E2 if F(E2+E1) <F(E1) ,else select E2+E1

4-2-7-1- ترکیب با استفاده از الگوریتم ژنتیک

الگوريتم ژنتيك، الهامي از علم ژنتيك و نظرية تكامل داروين است و بر اساس بقاي برترين‌ها يا انتخاب طبيعي استوار است. يك كاربرد متداول الگوريتم ژنتيك، استفاده از آن بعنوان تابع بهينه‌كننده است. الگوريتم ژنتيك ابزار سودمندي دربازشناسي الگو ،انتخاب ويژگي،درك تصويرو يادگيري ماشيني است[12]. درالگوريتم ژنتيك، نحوه تكامل ژنتيكي موجودات زنده شبيه‌سازي مي‌شود.

اين الگوريتم براي كاربردهاي مهندسي و به صورت امروزي آن نخستين بار توسط جان هلند[1] متخصص علوم كامپيوتر دانشگاه ميشيگان در سال 1975 پيشنهاد گرديد. كار وي آغاز تمامي كوشش ها براي كاربرد الگوريتم ژنتيك در مهندسي است .

در يك الگوريتم ژنتيك يك جمعيت از افراد طبق مطلوبيت آنها در محيط بقا مي يابند. افرادي با قابليتهاي برتر، شانس ازدواج وتوليد مثل بيشتري را خواهند يافت. بنابراين بعد از چند نسل فرزنداني با كارايي بهتر بوجود مي‌آيند. در الگوريتم ژنتيك هر فرد از جمعيت بصورت يك كروموزوم معرفي مي‌شود. كروموزوم ها در طول چندين نسل كاملتر مي‌شوند. در هر نسل كروموزوم ها ارزيابي مي‌شوند و متناسب با ارزش خود امكان بقا و تكثير مي‌يابند. توليد نسل در بحث الگوريتم ژنتيك با عملگرهاي توليد مثل[2] و جهش[3] صورت مي‌گيرد. والدين برتر بر اساس يك تابع برازندگي انتخاب مي‌شوند[13].

در هر مرحله از اجراي الگوريتم ژنتيكي، يك دسته از نقاط فضاي جستجو مورد پردازش‌هاي تصادفي قرار مي‌گيرند. به اين صورت كه به هر نقطه دنباله‌اي از كاراكترها نسبت داده مي‌شود و بر روي اين دنباله‌ها، عملگرهاي ژنتيكي اعمال مي‌شود. در آخر براساس اين كه تابع هدف در هر يك از نقاط چه مقدار باشد، احتمال شركت نمودن آنها در مرحله بعد تعيين مي‌گردد. [14]

اين الگوريتم را مي‌توان يك روش بهينه‌سازي تصادفي جهت‌دار دانست كه به تدريج به سمت نقطه بهينه حركت مي‌كند. در مورد ويژگي‌هاي الگوريتم ژنتيك در مقايسه با ديگر روش‌هاي بهينه سازي مي‌توان گفت كه الگوريتمي است كه بدون داشتن هيچ گونه اطلاعي از مسئله و هيچ گونه محدوديتي بر نوع متغيرهاي آن براي هر گونه مسئله اي قابل اعمال است و داراي

[1] John Holland
[2] Cross over
[3] Mutation

كارآيي اثبات شده‌اي در يافتن بهينه كلي[1] مي‌باشد. توانايي اين روش در حل مسائل پيچيده بهينه‌سازي، است كه روش‌هاي كلاسيك يا قابل اعمال نيستند و يا دريافتن بهينه كلي قابل اطمينان نيستند[14].

5-2-7-1- ساختار الگوريتم ژنتيك

به طور كلي، الگوريتم‌هاي ژنتيكي از اجزاء زير تشكيل مي‌شوند:

- **كروموزوم[2]**

در الگوريتم‌هاي ژنتيكي، هر كروموزوم نشان دهنده يك نقطه در فضاي جستجو و يك راه‌حل ممكن براي مسئله مورد نظر است. خود كروموزوم‌ها (راه حل‌ها) از تعداد ثابتي ژن[3] (متغير) تشكيل مي‌شوند. براي نمايش كروموزوم‌ها، معمولاً از كدگذاري‌هاي دودويي (رشته‌هاي بيتي) استفاده مي‌شود[14].

- **جمعيت[4]**

مجموعه‌اي از كروموزوم‌ها يك جمعيت را تشكيل مي‌دهند. با تاثير عملگرهاي ژنتيكي بر روي هر جمعيت، جمعيت جديدي با همان تعداد كروموزوم تشكيل مي‌شود.

- **تابع برازندگي[5]**

به منظور حل هر مسئله با استفاده از الگوريتم‌هاي ژنتيكي، ابتدا بايد يك تابع برازندگي براي آن مسئله ابداع شود. براي هر كروموزوم، اين تابع عددي غير منفي را برمي‌گرداند كه نشان دهنده شايستگي يا توانايي فردي آن كروموزوم است.

- **عملگرهاي ژنتيكي**

در الگوريتم‌هاي ژنتيكي، در طي مرحله توليد مثل[6] ازعملگرهاي ژنتيكي استفاده مي‌شود. با تاثير اين عملگرها بر روي يك جمعيت، نسل[7] بعدي آن جمعيت توليد مي‌شود. عملگرهاي انتخاب[8] ، آميزش[9] و جهش[10] معمولاً بيشترين كاربرد را در الگوريتم‌هاي ژنتيكي دارند.

- **عملگر انتخاب**

اين عملگر از بين كروموزوم‌هاي موجود در يك جمعيت، تعدادي كروموزوم را براي توليد مثل انتخاب مي‌كند. كروموزوم‌هاي برازنده‌تر شانس بيشتري دارند تا براي توليد مثل انتخاب شوند.

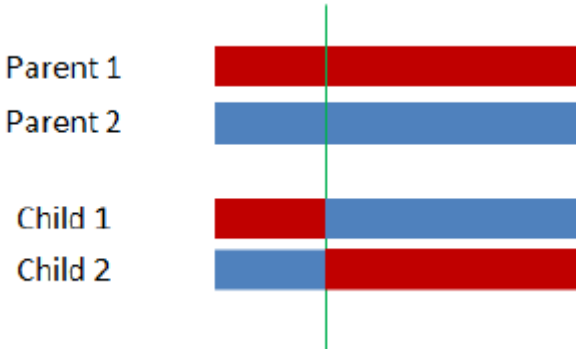
- **عملگر آميزش**

---

[1] Global Optimum
[2] Chromosome
[3] Gene
[4] Population
[5] Fitness Function
[6] Reproduction
[7] Generation
[8] Selection
[9] Crossover
[10] Mutation

عملگر آميزش بر روي يك زوج كروموزوم از نسل مولد عمل كرده و يك زوج كروموزوم جديد توليد مي‌كند. عملگرهاي آميزش متعددي از قبيل، آميزش تك نقطه اي[1] و آميزش دو نقطه اي[2] وجود دارد.

در آميزش تك نقطه اي، يك موقعيت تصادفي بين دو ژن در نظر گرفته مي‌شود. سپس تمامي ژن‌هاي طرف راست يا طرف چپ اين موقعيت در كروموزوم‌هاي والد با يكديگر جابجا مي‌شوند تا كروموزوم‌هاي جديد بدست آيند. در شكل 2 آميزش تك نقطه اي نشان داده شده است. در آميزش دو نقطه اي، دو موقعيت به صورت تصادفي انتخاب مي‌شود و تمامي ژن‌هاي بين اين دو موقعيت در كروموزوم‌هاي والد با يكديگر جابجا مي‌شوند. لازم به ذكر است كه آميزش معمولاً بر روي همه زوج كروموزوم‌هاي انتخاب شده براي جفتگيري به كار برده نمي‌شود. معمولاً احتمال آميزش براي هر زوج كروموزوم بين 0/6 تا 0/95 در نظر گرفته مي‌شود كه به اين عدد نرخ آميزش[3] يا احتمال آميزش[4] گفته مي‌شود و با P_c نمايش داده مي‌شود. در صورتي كه بر روي يك زوج كروموزوم عمل آميزش صورت نگيرد، فرزندان با تكرار نمودن والدين توليد مي‌شوند[15].



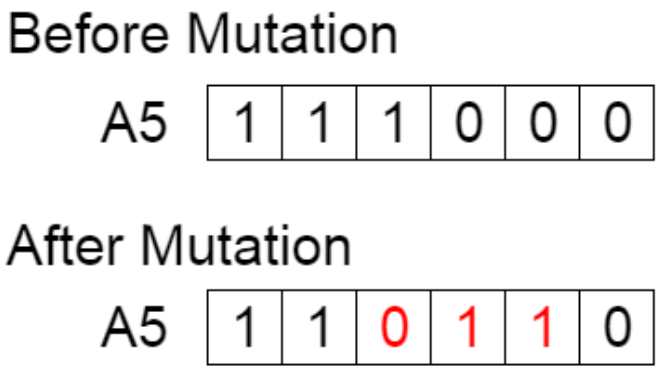
شكل 1-2- آميزش تک نقطه ايی [16]

- **عملگر جهش**

پس از اتمام عمل آميزش، عملگر جهش بر روي كروموزوم‌ها اثر داده مي‌شود. اين عملگر يك ژن از يك كروموزوم را به طور تصادفي انتخاب نموده و سپس محتواي آن ژن را تغيير مي‌دهد. اگر ژن از جنس اعداد دودويي باشد، آن را به وارونش تبديل مي‌كند و چنانچه متعلق به يك مجموعه باشد، مقدار يا عنصر ديگري از آن مجموعه را به جاي آن ژن قرار مي‌دهد. در شكل 3 چگونگي جهش يافتن پنجمين ژن يك كروموزوم نشان داده شده است[17].

احتمال انجام عمل جهش بر روي هر كروموزوم را نرخ جهش[5] يا احتمال جهش[6] مي‌گويند و با P_m نمايش مي‌دهند. معمولاً اين عدد را بسيار كوچك (مثلاً 0/001) در نظر مي‌گيرند. پس از اتمام عمل جهش، كروموزوم‌هاي توليد شده به عنوان نسل جديد شناخته شده و براي دور بعد اجراي الگوريتم ارسال مي‌شوند[18].

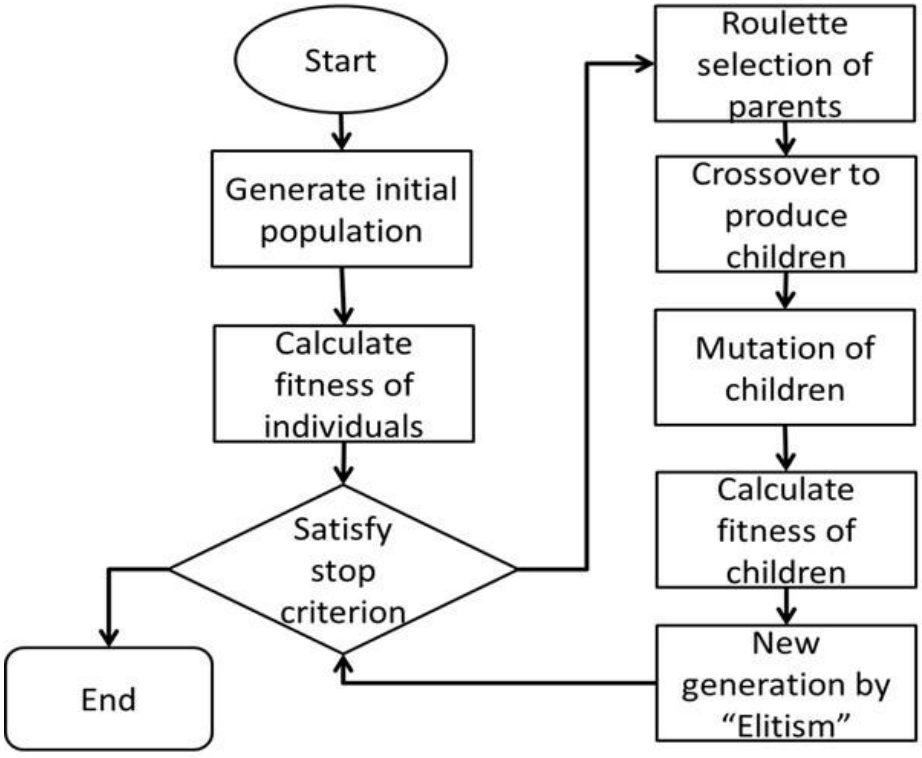
[1] One-point Crossover
[2] Two-point Crossover
[3] Crossover Rate
[4] Crossover Probability
[5] Mutation Rate
[6] Mutation Probability



شکل 1-3- عمل جهش [19]

1-7-2-6- روند الگوريتم ژنتيك

در شكل 4 نمودار گردشي الگوريتم‌هاي ژنتيكي نشان داده شده است. قبل از اين كه يك الگوريتم ژنتيكي بتواند اجرا شود، ابتدا بايد نمايش مناسبي براي مسئله مورد نظر پيدا شود. همچنين يك تابع برازندگي نيز بايد ابداع شود تا به هر راه حل كدگذاري شده ارزشي را نسبت دهد. در طي اجرا، والدين براي توليد مثل انتخاب مي‌شوند و با استفاده از عملگرهاي آميزش و جهش با هم تركيب مي‌شوند تا فرزندان جديدي توليد كنند. اين فرآيند چندين بار تكرار مي‌شود تا نسل بعدي جمعيت توليد شود. سپس اين جمعيت بررسي مي‌شود و و در صورتي كه ضوابط همگرايي برآورده شوند، فرآيند فوق خاتمه مي‌يابد[20].



شكل1-4- گردش كار الگوريتم ژنتيك [21]

1-7-2-7- روشهاي انتخاب

روشهاي انتخاب متعددي براي استفاده در الگوريتم‌هاي ژنتيكي پيشنهاد شده‌اند كه در ادامه اين بخش، برخي از اين روش معرفي مي‌شوند[22].

- **نمونه‌برداري به روش چرخ رولت**

در اين روش، به هر فرد قطعه‌اي[1] از يك چرخ رولت مدور اختصاص داده مي‌شود. اندازه اين قطعه متناسب با برازندگي آن فرد است. چرخ N بار چرخانده مي‌شود كه N تعداد افراد در جمعيت است. در هر چرخش، فرد زير نشانگر چرخ انتخاب مي‌شود و در مخزن والدين نسل بعد قرار مي‌گيرد.

- **انتخاب تورنمنت**

در انتخاب تورنمنت[2] دو فرد از جمعيت به صورت تصادفي انتخاب مي‌شوند. سپس، يك عدد تصادفي r بين 0 و 1 انتخاب مي‌شود. اگر $r<k$ (كه k يك پارامتر است، براي مثال 0/75) باشد، فرد برازنده‌تر و در غير اينصورت فردي كه برازندگي كمتري دارد، به عنوان والد انتخاب مي‌شود. اين دو سپس به جمعيت اوليه بازگردانده مي‌شوند و دوباره در فرآيند انتخاب شركت داده مي‌شوند[23].

1-7-2-8- شرط پايان الگوريتم

1-رسيدن به جواب بهينه

2-رسيدن به تعداد تكرار از پيش تعيين شده

1-7-2-9- برخي از كاربرد الگوريتم‌هاي ژنتيكي

الگوريتم‌هاي ژنتيكي در حل بسياري از مسائل علمي و مهندسي به كار گرفته شده‌اند. برخي از موارد كاربرد اين الگوريتم‌ها عبارتند از:

بهينه‌سازي، برنامه‌سازي خودكار، يادگيري ماشين، تكامل تدريجي و يادگيري، اكولوژي و سيستم‌هاي اجتماعي.

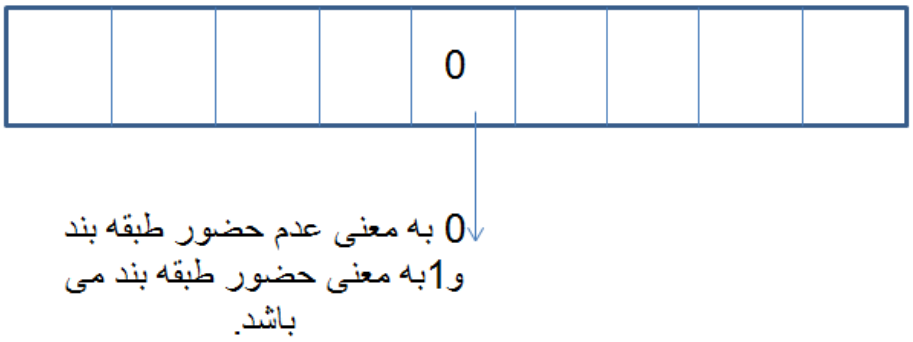
1-7-2-10- شرط پايان الگوريتم

- رسيدن به تعداد تكرار مورد نظر
- بدست اوردن جواب بهينه

1-7-2-11- روش كار

نگاشت مسئله به الگوريتم ژنتيک به اين صورت است كه يک كروموزم به طول تعداد مدل ها (100) ايجاد می كنيم، متغير های اين بردار با مقادير 0و1 پر می شوند، به اين معنی كه هر خانه ی اين بردار متناظر با يک مدل است. مقدار 0 به معنی عدم حضور مدل مربوطه در تركيب و مقدار 1 به معنی حضور مدل در تركيب است. در نهايت همه ی مدل هايی كه خانه ی متناظر با آنها با مقدار يک پر شود در تركيب حضور دارند.

[1] Slice
[2] Tournament Selection



شکل 1-5- بردار راه حل

جمعیت اولیه الگوریتم ژنتیک، 300مقدار دهی شد .(سه برابر طول بردار راه حل) نرخ ضریب جهش 0.01در نظر گرفته شد ، نرخ ضریب آمیزش 0.45 مقدار دهی شد . برای انجام عمل cross over از روش انتخابی roulette wheel برای انتخاب والدین استفاده شد و ارث بری فرزندان از والدین به شیوه ی single point انجام گرفت. برای محاسبه ی تابع ارزیابی (fitness function) بعد از ترکیب مدل ها f_score بدست آورده شد.

فصل دوم
نتایج

2-1- نتایج جداسازی مجموعه داده

جدول 2-1- جدا سازی مجموعه داده

	تعداد نمونه	تعداد ویژگی
X_ train	48	7128
Y_ train	48	-
X_ validate	3	7128
Y_ validate	3	-
X_ test	18	7128
Y_ test	18	-

مراحل پیش پردازش داده ها منجر به جابجایی سطر و ستون های مجموعه داده شد و در نهایت کلاس ها در ستون آخر قرار گرفت و سپس جدا سازی داده ها با توزیع یکسان به صورت تصادفی به 3 دسته ی (داده های آموزش، داده های آزمایش و داده های توسعه) تقسیم شد که از این بین 70% برای داده های آموزش، 20% داده های آزمایش و 10% داده های توسعه در نظر گرفته شد و در نهایت به برچسب AML عدد 0 و برچسب ALL عدد 1 اختصاص داده شد.تعداد نمونه ها برای داده های آموزش ، آزمایش و داده های توسعه به ترتیب 48 و18 و 3 بدست امد.

2-2- نتایج ارزیابی مدل های ساخته شده با الگوریتم های ماشین بردار پشتیبان و بیز و درخت تصمیم روی development data:

جدول 2-2- مقایسه طبقه بند ماشین بردار پشتیبان ، درخت تصمیم و بیز

	Precision	Recall	F_ score
SVM	0.726	1.0	0.741
naïve byes	0.95	1.0	0.976
decision tree	0.95	1.0	0.962

با توجه به نتایج بدست امده از ارزیابی طبقه بندهای SVM وNaïve byes و decision tree برمبنای F_score ، الگوریتم naïve byes به عنوان الگوریتم پایه برای ساخت 100 مدل انتخاب شد.

2-3- ایجاد 100 مدل با الگوریتم پایه

جدول2-3- نتایج ارزیابی مدل ها ی ایجاد شده با الگوریتم پایه

	precision	recall	F _ score
مدل 1تا92	1.0	1.0	1.0
مدل 93تا100	0.6666	0.6666	0.6666

100مدل با استفاده از الگوریتم پایه ساخته شد ، سپس ارزیابی آن ها روی داده های توسعه انجام شد . که این نتایج در فایل allModels_F-score.xls ضمیمه شده است .با توجه به تعداد کم اعضای داده ی توسعه که ارزیابی 100مدل ایجاد شده روی آنها انجام شد ، نتایج ارزیابی از مدل 1تا 92 کاملا یکسان و از مدل 93تا 100نیز مشابه بدست آمد.

2-4- مقایسه ی روش های ترکیب

جدول 2-4- نتایج روش های ترکیب

	precision	**recall**	**F_ score**
ترکیب همه مدل ها			
انتخاب رو به جلو			
حذف رو به عقب			
الگوریتم ژنتیک			
بهترین مدل			

منابع

[1] http://bioinformatics.upmc.edu/Help/ UPITTGED.html

[2] http://www-genome.wi.mit.edu/cgi-bin/ cancer/datasets.cgi

[3] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. "Gene selection for cancer classification using support vector machines", Machine Learning, 2000.

[4] Yuh-Jye Lee and O. Mangasarian. "DT: A smooth Decision Tree for classification", Computational Optimization and Applications, 20, 5-22, 2001.

[5] L. Kaufman. "Solving the quadratic programming problem arising in support vector classification", in Nave-Bayes Classification, B. Schölkopf, C. J. C. Burges, and A. J. Smola, eds., MIT Press, 47-167, 1999.

[6] http://web.stanford.edu/~hastie/CASI_files/DATA/leukemia.html

[7] https://medium.com/deep-math-machine-learning-ai/chapter-4decision-trees-algorithms-b93975f7a1f1

[8] http://www.statsoft.com/textbook/support-vector-machines

[9] http://www.statsoft.com/textbook/naive-bayes-classifier

[10] https://towardsdatascience.com/ensemble-methods-in-machinelearning-what-are-they-and-why-use-them-68ec3f9fef5f

[11] Brusco Cradit, Steinly.An exact algorithm for hierarchically well-formulated subsets in second-order polynomial regression .Technometrics, 51(3):306-315, 2009

[12] Elattar, E. E. (2015). A hybrid genetic algorithm and bacterial foraging approach for dynamic economic dispatch problem. International Journal of Electrical Power & Energy Systems, 69, 18–26.

[13] Man, K.F., Tang, K.S., Kwong, S.: Genetic Algorithms, pp. 5–10. Springer, Concepts Des. (2000) .

[14] Bagchi T (1999) Multiobjective scheduling by genetic algorithms. Kluwer, Boston

[15] Goldberg DE (1989) Genetic algorithms for search, optimization, and machine learning. Addison-Wesley, Reading .

[16] https://www.researchgate.net/figure/Single-pointcrossover_fig21_265969600

[17]http://web.cs.ucdavis.edu/~vemuri/classes/ecs271/Genetic%20Algorithms%20Short%20Tutorial.htm

[18] Coello CAC, Toscano G (2000) A micro-genetic algorithm for multi-objective optimization. Technical report Lania-RI-2000–06, Laboratoria Nacional de Informatica Avanzada, Xalapa, Veracruz.

[19] Zahhad Abo, M., Ahmed Triumph No, S., Sasaki, S.: The new energy-efficient protocol for adaptive genetic algorithm to the collection month, and improve wireless sensor networks. Int. J. Energy Inf. Commun. **5**(3), 47–72 (2014) .

[20] Goldberg DE (1989) Genetic algorithms for search, optimization, and machine learning. Addison-Wesley, Reading

[21] https://www.researchgate.net/figure/Flowchart-of-the-Genetic-algorithm-workflow-The-first-population-was-generated-randomly_fig9_50264725

[22] Deb K (1999) Solving goal programming problems using multi-objective genetic algorithms. In: Proceedings of the CEC, Washington, and pp 77–84 .

[23] Deb K, Agrawal S, Pratap A, Meyarivan T (2002) a fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6:182–197