یازدهمین کنفرانس سیستم‌های هوشمند ایران
۹ و ۱۰ اسفند ۱۳۹۱
(ICIS۲۰۱۳)
11th Iranian Conference on Intelligent Systems
February 27th & 28th, 2013

یک الگوریتم جدید ممتیک مبتنی بر اتوماتای یادگیر برای حل مساله انتخاب ویژگی

مهدی رضاپور میرصالح[1]، محمد رضا میبدی[2] و محمد رحمتی[3]
[1و2و3]دانشگاه صنعتی امیرکبیر ، (mrezapoorm, mmeybodi, rahmati)@aut.ac.ir

چکیده – پیدا کردن زیرمجموعه‌ای از یک مجموعه بزرگِ ویژگی‌ها، مسئله‌ای است که در بسیاری از زمینه‌ها با آن روبرو هستیم. از آنجایی که افزایش تعداد ویژگی‌ها، هزینه محاسباتی یک سیستم را افزایش می‌دهد، طراحی و پیاده سازی سیستم‌ها با کمترین تعداد ویژگی با کارایی قابل قبول، ضروری به نظر می‌رسد. این موضوع ما را به سمتی هدایت می‌کند که از روش‌های جستجو برای پیدا کردن زیرمجموعه‌ای بهینه از ویژگی‌ها استفاده کنیم. در این مقاله یک الگوریتم ممتیک جدید، مبتنی بر اتوماتای یادگیر برای حل مساله انتخاب ویژگی ارائه شده است. روش ارائه شده یک روش تکاملی است که از ترکیب الگوریتم ژنتیک سنتی و جستجوی محلی حاصل شده است. در این روش از اتوماتون یادگیر با ساختار ثابت جهت بازنمایی کروموزم‌ها و انجام جستجوی عمومی استفاده شده و با استفاده از دانش مساله، یک جستجوی محلی جهت ترکیب با الگوریتم ژنتیک ارائه شده است. مقایسه روش ارائه شده با سایر روش‌های متاهیوریستیک برتری این الگوریتم را نشان می‌دهد.

کلید واژه- مساله انتخاب ویژگی، مساله طبقه‌بندی، اتوماتای یادگیر، الگوریتم ممتیک.

۱- مقدمه

پیشرفت‌های بوجود آمده در جمع‌آوری داده و قابلیت‌های ذخیره‌سازی آن در طی دهه‌های اخیر، باعث شده است که در بسیاری از علوم با حجم بزرگی از اطلاعات روبرو شویم. محققان در زمینه‌های مختلف مانند مهندسی، ستاره شناسی، زیست شناسی و اقتصاد هر روز با مشاهدات بیشتر و بیشتری روبرو می‌شوند. بسترهای داده‌ای امروزی در مقایسه با بسترهای داده‌ای قدیمی، چالش‌های جدیدی را در تحلیل داده‌ها بوجود آورده‌اند. امروزه روش‌های آماری سنتی به دو دلیل کارائی خود را از دست داده‌اند. علت اول افزایش تعداد مشاهدات است، و علت دوم که از اهمیت بالاتری برخوردار است افزایش تعداد متغیرهای مربوط به یک مشاهده می‌باشد. تعداد متغیرهایی که برای هر مشاهده باید اندازه گیری شود ابعاد داده نامیده می‌شود. بسترهای داده‌ای که دارای ابعاد زیادی هستند علیرغم مزایای زیادی که دارند، چالش‌های محاسباتی زیادی را ایجاد می‌کنند. یکی از مشکلاتِ داده‌های با ابعاد بزرگ اینست که در بیشتر مواقع تمام ویژگی‌های داده‌ها برای یافتن دانشی که در داده‌ها نهفته است مهم و حیاتی نیستند. به همین دلیل در بسیاری از زمینه‌ها کاهش ابعاد داده یکی از مباحث قابل توجه باقی مانده است. روش‌های کاهش ابعاد داده به دو دسته تقسیم می‌شوند:

روش‌های مبتنی بر استخراج ویژگی: این روش‌ها یک فضای چند بعدی را به یک فضای با ابعاد کمتر نگاشت می‌کنند. در واقع با ترکیب مقادیر ویژگی‌های موجود، تعداد کمتری ویژگی بوجود می‌آورند بطوریکه این ویژگی‌ها دارای تمام (یا بخش اعظمی از) اطلاعات موجود در ویژگی‌های اولیه باشند. این روش‌ها به دو دسته‌ی خطی و غیر خطی تقسیم می‌شوند[۱,۲].

روش‌های مبتنی بر انتخاب ویژگی: این روش‌ها سعی می‌کنند با انتخاب زیرمجموعه‌ای از ویژگی‌های اولیه، ابعاد داده‌ها را کاهش دهند. در پاره‌ای از اوقات تحلیل‌های داده‌ای نظیر طبقه‌بندی برروی فضای کاسته شده نسبت به فضای اصلی بهتر عمل می‌کند. هدف اصلی از مساله انتخاب ویژگی، کاهش بعد بردار

ویژگی در طبقه‌بندی است بطوریکه نرخ طبقه‌بندی قابل قبولی نیز حاصل شود. در این شرایط ویژگی‌هایی که قدرت تمایز کمتری دارند، حذف شده و یکسری از ویژگی‌ها که شامل اطلاعات مناسبی برای تمایز کلاس‌های الگو هستند، باقی می‌مانند[۱,۲].

مساله انتخاب ویژگی، یکی از مسائل مهمی است که در مبحث یادگیری ماشین، شناسائی آماری الگو و غیره مطرح است. این مساله در بسیاری از کاربردها اهمیت به سزائی دارد، زیرا در این کاربردها تعداد زیادی ویژگی وجود دارد که بسیاری از آنها یا بلااستفاده هستند و یا اینکه بار اطلاعاتی چندانی ندارند. حذف نکردن این ویژگی‌ها مشکلی از لحاظ اطلاعاتی ایجاد نمی‌کند ولی بار محاسباتی را برای کاربرد مورد نظر بالا می‌برد و علاوه بر این باعث می‌شود که اطلاعات غیر مفید زیادی را به همراه داده‌های مفید ذخیره کنیم[۴]. برای مساله انتخاب ویژگی، راه حل‌ها و الگوریتم‌های فراوانی ارائه شده است که بعضی از آنها قدمت سی یا چهل ساله دارند. مشکل بعضی از الگوریتم‌ها در زمانی که ارائه شده بودند، بار محاسباتی زیاد آنها بود، اگر چه امروزه با ظهور کامپیوترهای سریع و منابع ذخیره سازی بزرگ، این مشکل به چشم نمی‌آید ولی از طرف دیگر، مجموعه‌های داده‌ای بسیار بزرگ برای مسائل جدید باعث شده است که همچنان پیدا کردن یک الگوریتم سریع برای این کار مهم باشد. روش‌های انتخاب ویژگی به دو گروه روش‌های فیلتری[۱] و روش‌های روکشی[۲] تقسیم بندی می شوند[۳]. روش های فیلتری با استفاده از یک ضابطه به وزندهی ویژگی ها پرداخته، سپس به انتخاب ویژگی‌های وزن داده شده می پردازند. این روش‌ها، دارای هزینه محاسباتی پائین بوده و با توجه به کیفیت روش وزندهی، دارای دقتهای متفاوتی می باشند. در روش‌های روکشی، ویژگی‌ها با توجه به تاثیرشان روی بهبود دقت دسته بندی، انتخاب می‌شوند. این روش ها دارای دقت بسیار بالا و در مقابل، هزینه محاسباتی بسیار زیادی هستند.[۴]

سازماندهی ادامه این مقاله بدین صورت است: بخش ۲ شامل تعریف این مساله و بررسی کلی مکانیزم انجام آن است. در بخش ۳ روشهای مختلف متاهیوریستیک بررسی می‌شود. در بخش ۴ یک روش جدید ممتیک که مبتنی بر اتوماتای یادگیر می‌باشد، ارائه شده است. بخش ۵ به مقایسه روش‌های مذکور اختصاص داده شده است و نهایتا در بخش ۶ به جمعبندی موارد مطرح شده پرداخته شده است.

۲- تعریف مساله

مساله انتخاب ویژگی عبارت است از انتخاب یک زیرمجموعه X با d ویژگی از مجموعه اولیه V که دارای N ویژگی است به طوریکه $d < N$ باشد و مقدار یک تابع معیار[۳] J(X) برای زیرمجموعه مورد نظر، نسبت به سایر زیرمجموعه‌های هم‌اندازه دیگر بهینه باشد به عبارت دیگر مقدار J(X) از رابطه ۱ حاصل می‌شود.

$$J(X) = \max_{Z \subseteq V, |Z|=d} J(Z) \qquad (۱)$$

در این مقاله برای محاسبه J(X) به این صورت عمل می‌کنیم: فرض کنید C مجموعه‌ای از نمونه‌هایی است که در دو کلاس طبقه‌بندی شده باشند و تعداد نمونه‌های هر کلاس تقریبا با یکدیگر برابر است. C را به دو زیر مجموعه C۱ و C۲ تقسیم می‌کنیم به گونه‌ای که $C = C1 \cup C2$. تعداد نمونه‌های موجود و تعداد نمونه های متعلق به هر کلاس در دو مجموعه C۱ و C۲ تقریبا با یکدیگر برابر است. به ازای هر نمونه C۲ فاصله اقلیدسی آنرا نسبت به تمام نمونه های C۱ محاسبه کرده و شماره کلاس نمونه‌ای که دارای کوتاهترین فاصله باشد را به عنوان کلاس نمونه مورد بررسی در نظر می‌گیریم. مقدار J(X) برابر است با درصد نمونه هایی که درست طبقه بندی می شوند یعنی تعداد نمونه‌هایی که کلاس آنها با کلاس واقعیشان یکسان باشد[۲].

۳- روش‌های متاهیوریستیک

۳-۱- الگوریتم GRASP

این الگوریتم یک فرایند تکراری است که هر مرحله آن از دو بخش تشکیل شده است. این دو فاز عبارتند از: فاز ساخت و فاز جستجوی محلی[۶]. یک جواب ممکن در فاز ساخت ایجاد می‌شود و سپس در فاز جستجوی محلی تمام همسایه آن مورد

بررسی قرار می‌گیرند. در فاز ساخت که از یک روش تکراری استفاده می‌شود، در هر مرحله مجموعه RLC[4] ایجاد می‌شود که شامل تعدای از ویژگی‌هایی است که در مجموعه جواب قرار ندارند. نحوه ایجاد مجموعه RLC به این صورت است که ابتدا مقدار برازندگی هر ویژگی، با فرض قرار داشتن آن در مجموعه جواب، توسط تابع g ارزیابی می‌شود و سپس ویژگی‌هایی که مقدار ارزیابی آنها در رابطه ۲ صدق کند در مجموعه RLC قرار می گیرند.

$$g(v_j) \leq \propto . g_{max} + (1-\propto). g_{min} \qquad (۲)$$

سپس به صورت تصادفی یکی از ویژگی‌های موجود در این مجموعه انتخاب و به مجموعه جواب اضافه می‌شود. در این رابطه g_{max} ، g_{min} به ترتیب مقدار می نیمم و ماکزیمم برازندگی ویژگی‌هایی است که توسط تابع g محاسبه شده اند و در مجموعه جواب قرار ندارند. اگر مقدار ۱ =∝ در نظر گرفته شود، مجموعه RLC شامل تمام ویژگی‌های انتخاب نشده خواهد بود و لذا انتخاب یک ویژگی به صورت کاملا تصادفی است و اگر مقدار ۰ =∝ فرض شود، مجموعه RLC تنها شامل ویژگی‌هایی است که دارای مقدار برازندگی g_{min} باشند. معمولا مقدار ۰.۸۵ =∝ در نظر گرفته می‌شود. در فاز جستجوی محلی با جابجایی متوالی ویژگی‌های مجموعه جواب (تشکیل شده در فاز ساخت) با ویژگی‌های همسایه‌ی آن، جواب مناسب‌تری پیدا خواهد شد. بدین صورت که اگر X مجموعه جواب به دست آمده و N(X) مجموعه همسایه‌های آن باشد، این مجموعه از رابطه ۳ به دست می‌آید:

$$N(X) = \{X'/X' = X \cup \{v_{j'}\} - \{v_j\}, \forall v_j \in X, v_{j'} \notin X\} \qquad (۳)$$

۳-۲- روش جستجوی تابو

در روش تابو همانند روش قبل هر یک از ویژگی‌های مجموعه جواب با جابجایی متعدد با همسایه‌ی آن، مجموعه جواب جدیدی را تشکیل می دهد[۷]. برای اینکه در جابجایی ویژگی‌ها از ایجاد حلقه اجتناب شود، تعداد تکرار خارج شدن هر ویژگی از مجموعه جواب باید در نظر گرفته شود. بدین منظور تعداد مراحلی که در آن ویژگی v_j از مجموعه جواب X خارج شده است با *vector_tabu* (v_j) نشان داده شده است. روش جستجوی تابو در شکل ۱ نشان داده شده است که در آن X مجموعه جواب فعلی و X^* مجموعه بهترین جواب می باشد. پارامتر *Tabu_Tenure* تعداد تکرار مراحلی را نشان میدهد که یک ویژگی نمی‌تواند به مجموعه جواب X اضافه شود. با توجه به آزمایشات متعدد انجام گرفته بهترین مقدار پارامتر *Tabu_Tenure* برابر با d می‌باشد.

Read initial solution X
Do $vector_tabu(v_j) = -Tabu_Tenure, \quad j = 1,\ldots,m; \quad niter = 0,$
$iter_better = 0$ *and* $X^* = X$
Repeat
- $niter = niter + 1$
- *Calculate* $\Delta_{jj'} = J(X \cup \{v_{j'}\} - \{v_j\})$
- *Determine* $\Delta_{j^*j'^*} = max\{\Delta_{jj'} / \forall v_j \in X, v_{j'} \notin X$ *verifying:* $niter > vector_tabu(v_j) + Tabu_Tenure$ *or* $\Delta_{jj'} > J(X^*)$ *('aspiration criterion')}*
- $DoX = X \cup \{v_{j^*}\} - \{v_{j'^*}\}$ *and* $vector_tabu\ (v_{j^*}) = niter$
- *If* $J(X) > J(X^*)$ *then do:* $X^* = X, J^* = J$ *and* $iter_better = niter;$

until $niter > iter_better + 2 \cdot m$

شکل ۱ -روش جستجوی تابو

هماهنگونه که در شکل نشان داده شده است، در این روش هنگامی که به تعداد $2.m$ مرحله بهبودی حاصل نشود، الگوریتم پایان نخواهد یافت.

۳-۳- الگوریتم ممتیک

الگوریتم های ممتیک از ترکیب الگوریتم ژنتیک و یک جستجوی محلی تشکیل می‌شوند. در این روش با استفاده از دانش مساله عملگرهای جهش و بازترکیبی تعریف می‌شوند که نقش بسیار مهمی در حل مساله دارند. این الگوریتم در مقالات مختلف با نام‌های مختلفی نظیر الگوریتم ژنتیک ترکیبی، الگوریتم ژنتیک موازی، جستجوی محلی ژنتیک و ... مورد استفاده قرار گرفته است. در این روش ابتدا یک جمعیت اولیه از موجودات (راه حل های ممکن)، به صورت تصادفی ایجاد می‌شود. سپس در هر نسل با استفاده از عملگرهای ژنتیکی و جستجوی محلی در جهت بهبود موجودات اقدام می‌شود. اولین گام در این روش بازنمایی موجودات در قالب کروموزوم است. در مرحله بعد شایستگی هر کروموزوم با توجه به تابع هدف اندازه-

گیری می‌شود و سپس با توجه به میزان شایستگی هر کروموزوم تعدادی از آنها انتخاب شده و عملگرهای جهش و بازترکیبی روی آنها اعمال می‌شود. در ادامه با انجام جستجوی محلی روی فرزندان، به راه‌حل‌های بهتری دست خواهیم یافت. در نهایت اگر شایستگی فرزندان ایجاد شده از والدینشان بهتر باشد، آنها را جایگزین والدین می کنیم. در این روش از بازنمایی باینری برای نمایش کروموزوم ها استفاده می‌شود. به عنون مثال $X =$ (۱،۰،۰،۰،۱،۰،۱،۰،۰،۱) یک جواب از مساله است که شامل ویژگی‌های ۱،۵،۷ و ۱۰ می باشد[۸]. در این الگوریتم از بازترکیبی تک‌نقطه‌ای استفاده شده است. بدین صورت که اگر راه‌حل‌های $X = (x_1, x_2, \dots, x_m)$ و $X' = (x'_1, x'_2, \dots, x'_m)$ را به عنوان والدین در نظر بگیریم، با انتخاب عددی تصادفی بین ۱ و m-۱، دو راه حل فرزند به صورت زیر ایجاد خواهد شد.

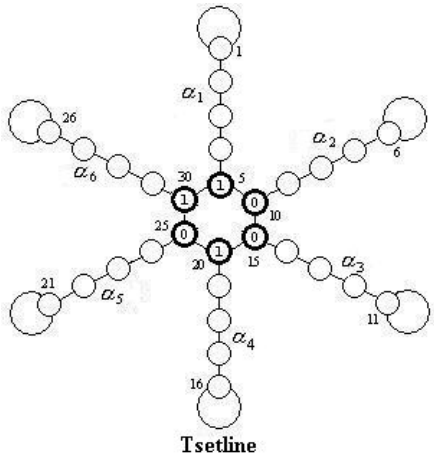
$$X^* = (x_1, x_2, \dots, x_{pto_{cruce}}, x'_{pto_{cruce}+1}, \dots, x'_m)$$
$$X^{**} = (x'_1, x'_2, \dots, x'_{pto_{cruce}}, x_{pto_{cruce}+1}, \dots, x_m)$$

همچنین عملگر جهش که وظیفه حفظ تنوع راه‌حل‌ها را بر عهده دارد، با احتمال p وضعیت وجود یک ویژگی در مجموعه جواب X را به عدم وجود تغییر می دهد(با تبدیل ۱ به ۰) و ویژگی جدیدی را به مجموعه جواب اضافه می کند و یا اینکه وضعیت عدم وجود یک ویژگی در مجموعه جواب X را به وجود تغییر می دهد (با تبدیل ۰ به ۱) و یکی از ویژگی‌های دیگر را از مجموعه جواب خارج می کند.

۴- روش ترکیبی مبتنی بر اتوماتای یادگیر

با ترکیب الگوریتم ژنتیکی و اتوماتون یادگیر و تلفیق مفاهیم ژن، کروموزوم، اقدام و عمق، می‌توان به یک روش جستجوی کارا برای حل مساله انتخاب ویژگی دست یافت. الگوریتم پیشنهادی در این بخش کوششی است در این جهت. خودترمیمی، تولید مثل، جریمه و پاداش از ویژگی‌های مهم الگوریتم ترکیبی است. در این روش ضمن بهره جستن از جستجوی عمومی الگوریتم ژنتیک در فضای حالت حل مساله، با استفاده از بستر اتوماتای یادگیر، یک جستجوی محلی به آن اضافه می‌کنیم. با توجه به این که در این روش از ترکیب الگوریتم ژنتیک و اتوماتون یادگیر و ادغام جستجوی عمومی و جستجوی محلی استفاده می‌شود، روش پیشنهادی نوعی الگوریتم ممتیک خواهد بود[۵]. در الگوریتم پیشنهادی برخلاف الگوریتم‌های ژنتیک کلاسیک، از کدگذاری دودویی برای کروموزوم‌ها استفاده نمی‌شود. هر کروموزوم توسط یک اتوماتون یادگیر از نوع مهاجرت اشیاء نشان داده می‌شود بطوریکه هر کدام از ژن‌های کروموزوم به یکی از اقدام‌های اتوماتون نسبت داده می‌شود و در یک عمق مشخصی از آن اقدام قرار می‌گیرند. در این اتوماتون $\underline{\alpha} = \{\alpha_1, \alpha_2, \dots, \alpha_k\}$ مجموعه اقدام‌های مجاز برای اتوماتون یادگیر است. این اتوماتون k اقدام دارد (تعداد اقدام‌های این اتوماتون با تعداد کل ویژگی‌ها برابر است). به عبارت دیگر هر اقدام اتوماتون معادل یک ویژگی در نظر گرفته می‌شود. به هر یک از این اقدام ها مقدار ۱ (به معنای وجود ویژگی در مجمو عه جواب) یا مقدار ۰ (به معنای عدم وجود ویژگی در مجمو عه جواب) نسبت داده می‌شود. $\underline{\varphi} = \{\varphi_1, \varphi_2, \dots, \varphi_{kN}\}$ مجموعه وضعیت‌ها و N عمق حافظه برای اتوماتون می‌باشد. مجموعه وضعیت‌های این اتوماتون به k (تعداد اقدام ها) زیرمجموعه افراز می‌شود و هر یک از ویژگی‌ها بر اساس این‌که در کدام وضعیت قرار داشته باشند دسته‌بندی می‌گردند. در مجموعه وضعیت‌های اقدام j وضعیت $\varphi_{(j-1)N+1}$ وضعیت داخلی و وضعیت φ_{jN} را وضعیت مرزی می‌نامیم. در اثر پاداش دادن یا جریمه کردن یک اقدام، وضعیت ویژگی معادل آن اقدام، تغییر می‌کند. پاداش دادن یک اقدام باعث حرکت ویژگی وابسته به آن به سمت وضعیت داخلی آن اقدام می‌شود و جریمه کردن آن باعث حرکت ویژگی به سمت وضعیت مرزی می‌گردد. اگر یک ویژگی در وضعیت مرزی یک اقدام قرار داشته باشد، جریمه شدن آن باعث تغییر وضعیت انتخاب ویژگی وابسته به آن می‌شود و در نتیجه باعث ایجاد ترکیب جدیدی از مجموعه جواب می‌گردد. مجموعه {۱،۰،۰،۱،۰،۱} یک انتخاب از مجموعه ویژگی‌ها را نشان می دهد که درآن ویژگی‌های ۱، ۴ و ۶ انتخاب شده اند و ویژگی‌های ۲، ۳ و ۵ انتخاب نشده اند. این انتخاب توسط یک اتوماتون یادگیر با اتصال‌های مشابه اتوماتون ستلین در شکل ۲ نشان داده شده است. این اتوماتون دارای ۶ اقدام $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6\}$ (به

تعداد ویژگی‌ها) و عمق ۵ می‌باشد. مجموعه وضعیت‌های {۲۶و۲۱و۱۶و۱۱و۶و۱} وضعیت های داخلی و مجموعه وضعیت های {۳۰و۲۵و۲۰و۱۵و۱۰و۵} وضعیت های مرزی اتوماتون هستند. در ابتدا هر یک از ویژگی‌ها در وضعیت مرزی اقدام مربوطه قرار دارند. در الگوریتم ترکیبی هر ژن از کروموزوم معادل یک اقدام اتوماتون می‌باشد و لذا می‌توان در ادامه این دو واژه را به جای یکدیگر بکار برد. این اتوماتون یادگیر(کروموزوم ژنتیک) دارای ۶ اقدام (ژن) می‌باشد و هر اقدام (ژن) دارای ۵ وضعیت داخلی است.



شکل ۲ - نمایش مجموعه ویژگی‌های {۱،۰،۰،۱،۰،۱}بوسیله اتوماتون یادگیر با اتصال‌های مشابه اتوماتون ستلین

- جمعیت اولیه

هر عضو جمعیت که یک اتوماتون یادگیر است، دارای k(به تعداد کل ویژگی‌ها) اقدام است. برای ایجاد جمعیت اولیه به صورت تصادفی به هریک از اقدام های اتوماتون مقدار ۱ یا ۰ تخصیص می دهیم. به عنوان مثال برای جمعیت اولیه به تعداد ۶ و با استفاده از اتوماتاهای با اتصالات مشابه ستلین، ابتدا ۶ اتوماتای ستلین ایجاد می‌شود و سپس در هر یک از آنها به صورت تصادفی به هر اقدام مقدار ۱ یا ۰ در وضعیت مرزی آن تخصیص داده می‌شود.

- تابع شایستگی

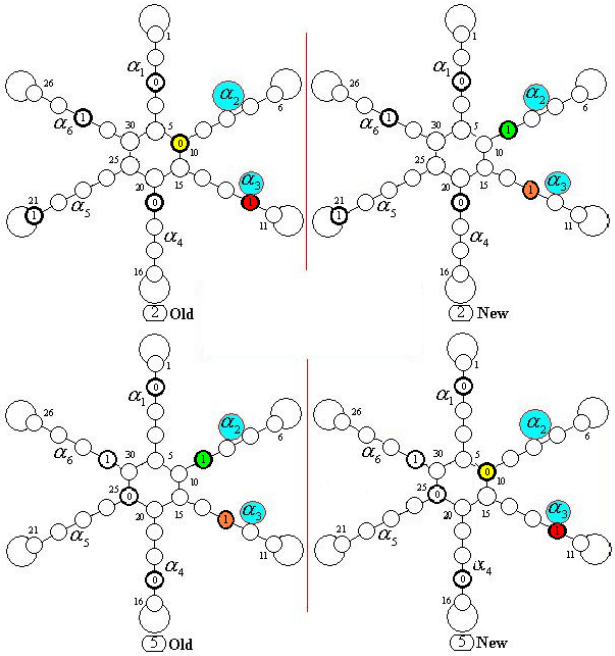
در الگوریتم‌های ژنتیک تابع شایستگی شاخص زنده ماندن کروموزوم‌ها است و در مساله انتخاب ویژگی به دنبال یافتن کروموزومی هستیم که تابع معیار (J(X)) را ماکزیمم کند لذا تابع شایستگی را می توان معادل تابع هدف در نظر گرفت.

- عملگرها

از آنجا که در الگوریتم ترکیبی، کروموزوم به صورت یک اتوماتون یادگیر نشان داده می شود عملگرهای جابجایی و جهش مشابه عملگرهای سنتی ژنتیکی نیست. در این قسمت به بررسی عملگرهای مورد نیاز در این الگوریتم می پردازیم.

الف) عملگر انتخاب: در الگوریتم ترکیبی، انتخاب والدین برای انجام عملگرهای جهش و بازترکیبی و انتخاب بازماندگان بر اساس مکانیزم رتبه بندی انجام می شود.

ب) عملگر بازترکیبی: در این عملگر دو کروموزوم والد انتخاب می‌شوند و به صورت تصادفی دو ژن i و j در یکی از دو کروموزوم والد انتخاب می‌شوند سپس همین دو ژن در کروموزوم دیگر انتخاب می‌شوند. مجموعه ژن‌های با شماره‌های بین i و j را مجموعه جابجایی می‌نامیم. سپس ژن‌های هم شماره با همان وضعیت در دو مجموعه جابجایی با یکدیگر جابجا می‌شوند (مثلا ژن شماره i از مجموعه جابجایی اول با ژن شماره i از مجموعه جابجایی دوم، ژن شماره i+۱ از مجموعه جابجایی اول با ژن شماره i+۱ از مجموعه جابجایی دوم و ...). با این عمل دو کروموزوم جدید حاصل می‌شوند که اصطلاحا فرزندان دو اتوماتای والد خوانده می‌شوند. با توجه به آزمایش‌هایی که انجام گرفته است نرخ اعمال این عملگر بایستی بزرگ و در حد ۰,۹ در نظر گرفته شود. به عنوان مثال اتوماتاهای LA_2,LA_5 از جمعیت تشکیل شده به صورت تصادفی انتخاب می‌شوند. سپس با انتخاب تصادفی دو محل $\{\alpha_2, \alpha_3\}$ مجموعه جابجایی $\{\alpha_2, \alpha_3\}$ حاصل می‌شود. و در نهایت مطابق شکل ۳ با جابجایی اقدام‌های متناظر در فاصله جابجایی، دو کروموزوم جدید که به ترتیب مجموعه جوابهای {۰،۱،۱،۰،۱،۱} و {۰،۰،۱،۰،۰،۱} را نشان می‌دهند، حاصل می‌شود.

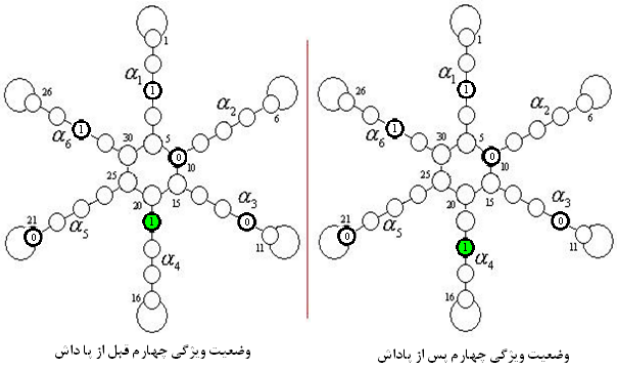


شکل ۳ -نحوه انجام عملگر جابجایی

ج) عملگر جهش: در عملگر جهش که بر روی یک اتوماتا (کروموزوم) انجام می‌شود، یک اقدام (ژن) به صورت تصادفی انتخاب شده و مقدار وابسته به آن تغییر می یابد. آزمایش‌ها نشان داده‌اند که نرخ اعمال این عملگر بایستی در حد ۰٫۳ در نظر گرفته شود.

د) عملگر جهش اول: استراتژی محلی که در این مساله از آن استفاده شده است، عملگر جهش اول، نام دارد. این عملگر همانگونه که از نام آن پیداست همان عملگر جهش است با این تفاوت که تنها روی بهترین موجود هر نسل اعمال می شود. در واقع با این عمل یک جستجوی محلی حول بهترین موجود انجام می شود و در صورتی که موجود شایسته تری پیدا شود، آنرا جایگزین بهترین موجود می‌کند. این عملگر شبیه روش تپه نوردی است با این استثناء که در هر مرحله تمام همسایه‌های بهترین جواب بررسی نمی‌شوند. یکی دیگر ار محاسن این عملگر، عدم تخریب سیر تکاملی جمعیت است که بوسیله چستجوی عمومی حاصل شده است. به عبارت دیگر چون این عملگر تنها بر روی بهترین موجود جمعیت اعمال می‌شود، اثرات ناشی از جستجوی عمومی بر روی سایر موجودات دست نخورده باقی خواهد ماند.

ه) عملگر جریمه و پاداش: از آنجا که هر کروموزوم به صورت یک اتوماتون یادگیر نشان داده شده است در هر یک از اتوماتا پس از بررسی میزان شایستگی یک اقدام (ویژگی یا ژن)، آن ژن پاداش یا جریمه می‌شود. نحوه جریمه یا پاداش یک اقدام بدین صورت است که، ابتدا مقدار تابع هدف را برای کروموزوم مربوطه بدست می آوریم و آنرا f_1 می نامیم، سپس با فرض تغییر یافتن مقدار وابسته به اقدام (از ۰ به ۱ یا از ۱ به ۰)، تابع هدف را برای کروموزوم حاصله محاسبه می‌کنیم و آنرا f_2 می نامیم اگر مقدار f_1 از f_2 بزرگتر باشد به اقدام مورد نظر پاداش می‌دهیم در غیر اینصورت آنرا جریمه می‌کنیم. در اثر پاداش دادن یا جریمه کردن یک اقدام، وضعیت ویژگی در مجموعه وضعیت های اقدام مربوطه، تغییر می کند. اگر یک ویژگی در وضعیت مرزی یک اقدام قرار داشته باشد، جریمه شدن آن باعث تغییر مقدار وابسته به آن اقدام و در نتیجه باعث ایجاد مجموعه جواب جدیدی می‌شود. نرخ این عملگر باید پایین باشد زیرا این عملگر، یک عملگر جستجوی تصادفی است و اگر با نرخ بالا اعمال شود باعث کاهش در کارایی الگوریتم می‌شود. عملگر جریمه و پاداش با توجه به نوع اتوماتون یادگیر متفاوت خواهد بود. به عنوان مثال در اتوماتون با اتصال های مشابه اتوماتان ستلین اگر ویژگی چهارم (معادل اقدام یا ژن چهارم) که در مجموعه وضعیت‌های {۲۰و۱۹و۱۸و۱۷و۱۶} قرار دارد، انتخاب شده باشد (مقدار وابسته به آن ۱ باشد)، و مشمول پاداش قرار گیرد، مقدار وابسته به آن به سمت وضعیت های داخلی تر حرکت می‌کند. اگر مقدار وابسته در وضعیت داخلی (وضعیت شماره ۱۶) قرار داشته باشد و پاداش بگیرد در همان وضعیت باقی می‌ماند. شکل ۴ وضعیت اتوماتون مربوطه قبل و بعد از اعمال پاداش را نشان می‌دهد. با اعمال این عملگر، مقدار ژنتیکی کروموزوم مربوطه تغییر نمی-یابد، و این عملگر تنها باعث تغییر وضعیت‌های داخلی هر اقدام (ژن) خواهد شد.

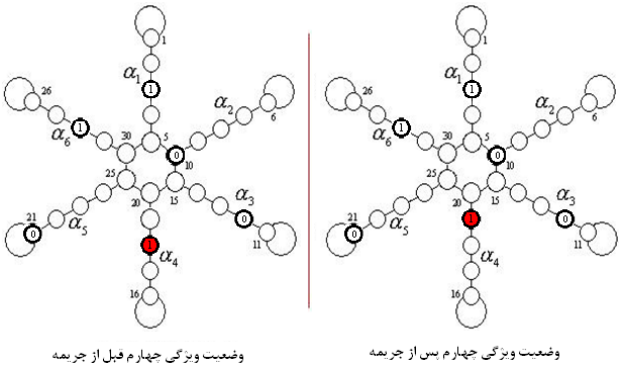


شکل ۴ -نحوه پاداش یک ویژگی

جریمه کردن یک اقدام با توجه به وضعیت فعلی مقدار وابسته به دو صورت می‌تواند باشد:

الف- مقدار وابسته در وضعیتی غیر از وضعیت مرزی قرار داشته باشد: جریمه نمودن این ویژگی باعث حرکت مقدار وابسته به سمت وضعیت مرزی اقدام می‌شود. به عبارت دیگر اگر مقدار وابسته ۱ باشد، جریمه کردن آن به معنی اینست که انتخاب این ویژگی در مجموعه جواب، احتمالا انتخاب درستی نیست. اگر مقدار وابسته ۰ باشد، جریمه کردن آن به معنی اینست که عدم انتخاب این ویژگی در مجموعه جواب، احتمالا کار درستی نبوده است. نحوه حرکت چنین اقدامی در شکل ۵ نشان داده شده است.

ب- مقدار وابسته در وضعیت مرزی قرار دارد: در این حالت جریمه کردن یک اقدام باعث تغییر مقدار وابسته می شود.



شکل ۵ -نحوه جریمه کردن یک ویژگی که مقدار وابسته آن در وضعیتی غیر از وضعیت مرزی قرار داشته باشد.

۵- مقایسه روش‌های متاهیوریستیک

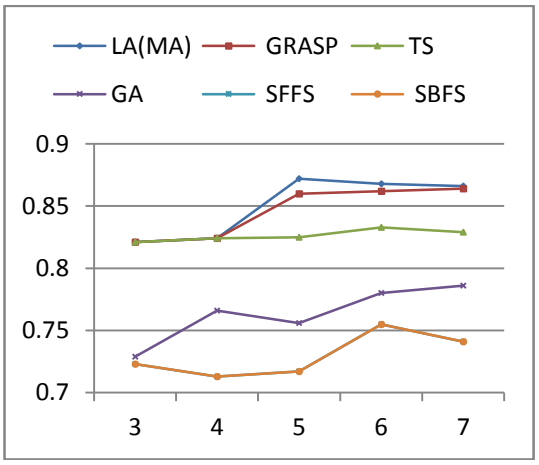
در این بخش به بررسی نتایج آزمایشات انجام شده روی دیتابیس های مختلف و مقایسه کارایی روش ارائه شده LA(MA) با الگوریتم‌های تابو (TS)، GRASP، GA، SFFS و SBFS می‌پردازیم. از شش دیتابیس معروف برای انجام مقایسه ها استفاده شده است[۲].

جدول ۱ - مشخصات دیتابیس مورد استفاده

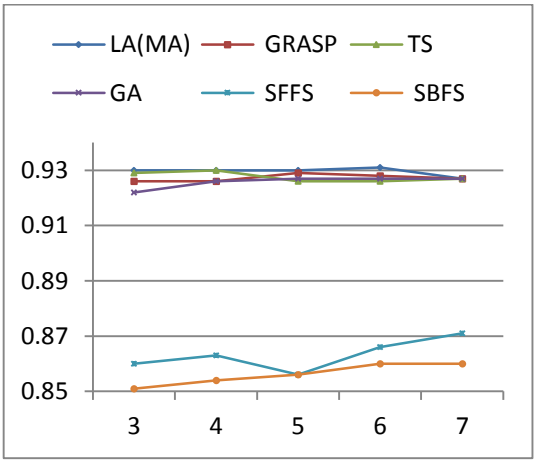
دیتابیس	تعداد نمونه	تعداد کلاس	تعداد ویژگی
Spambase	۴۶۰۱	۲	۵۷
Ionosphere	۳۵۱	۲	۳۴
German	۱۰۰۰	۲	۴۴

در هر دیتابیس از یک ۱*۱۰ Cross-Validation برای اندازه گیری و تست روش‌های مختلف استفاده شده است.

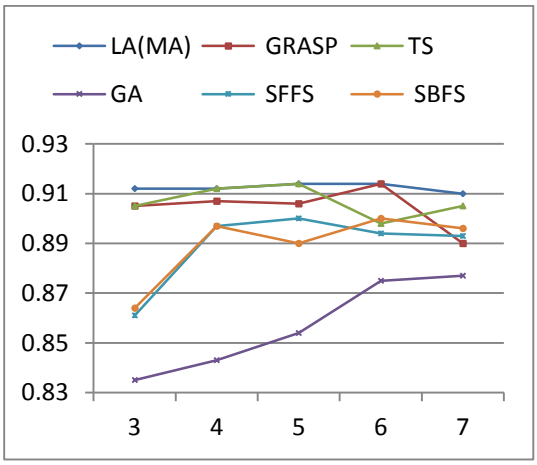
در شکل ۶ متوسط مقادیر تابع هدف در ۵۰ آزمایش، در الگوریتم های مختلف به ترتیب برای دیتابیس‌های Spam، Ionosphere و German نشان داده شده است. همانگونه که مشخص است کارایی الگوریتم ممتیک مبتنی بر اتوماتای یادگیر نسبت به الگوریتم سایر الگوریتم‌ها بهتر است. مهمترین دلیلی که می‌توان برای این مساله بیان کرد اینست که در الگوریتم ترکیبی به دلیل استفاده از عملگرهای پاداش و جریمه تنوع جمعیت همیشه شده و از قرار گرفتن در مینیمم محلی جلوگیری می‌شود. در حالیکه در روش‌های GA و TS احتمال قرار گرفتن در می‌نیمم محلی زیاد است. همچنین استفاده از جستجوی محلی باعث تغییر اتوماتاتیک جستجوی عمومی و محلی می‌شود.



شکل ۶- متوسط مقادیر تابع هدف در الگوریتم های مختلف برای دیتابیس Spam (محور افقی d و محور عمودی J را نشان می‌دهد)



شکل ۷- متوسط مقادیر تابع هدف در الگوریتم های مختلف برای دیتابیس German (محور افقی d و محور عمودی J را نشان می‌دهد)



شکل ۸- متوسط مقادیر تابع هدف در الگوریتم های مختلف برای دیتابیس Lonosphra (محور افقی d و محور عمودی J را نشان می‌دهد)

۶- نتیجه‌گیری

در این مقاله ضمن بررسی مساله انتخاب ویژگی، روش‌های مختلف جستجو و ارزیابی مجموعه جواب‌ها را بررسی نمودیم. همچنین الگوریتم جدید ممتیکی مبتنی بر اتوماتای یادگیر ارائه شد. این الگوریتم، ترکیبی از الگوریتم ژنتیک سنتی جهت انجام جستجوی عمومی و یک روش جستجوی محلی است. کروزموزوم‌های جمعیت توسط اتوماتون یادگیر با ساختار ثابت بازنمایی شدند و عملگرهای متناسب با این نوع بازنمایی تعریف گردید. همچنین کارایی این روش در مقایسه با سایر روش‌ها نشان داده شد. بررسی و یافتن مقدار بهینه پارامترهای مختلف این الگوریتم، می‌تواند در پژوهش‌های بعدی مورد بررسی قرار گیرد.

مراجع

۱. Lazar, Cosmin, et al. "A survey on filter techniques for feature selection in gene expression microarray analysis." *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)* ۹,٤ (۲۰۱۲): ۱۱۰٦-۱۱۱۹.

۲. Yusta, Silvia Casado. "Different metaheuristic strategies to solve the feature selection problem." *Pattern Recognition Letters* ۳۰,٥ (۲۰۰۹): ٥۲٥-٥۳٤.

۳. Zhu, Zexuan, Yew-Soon Ong, and Manoranjan Dash. "Wrapper–filter feature selection algorithm using a memetic framework."

Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on ۳۷,۱ (۲۰۰۷): ۷۰-۷۶.

٤. Dash, Manoranjan, and Huan Liu. "Feature selection for classification." *Intelligent data analysis* ۱,۱-٤ (۱۹۹۷): ۱۳۱-۱٥٦.

٥. Rezapoor Mirsaleh, M. and Meybodi, M. R., "A Hybrid Algorithm for Solving Graph Isomorphism Problem", *Proceedings of the Second International Conference on Information and Knowledge Technology (IKT۲۰۰٥), Tehran, Iran*,(۲۰۰٥).

۶. Feo, Thomas A., and Mauricio GC Resende. "Greedy randomized adaptive search procedures." *Journal of global optimization* ۶,۲ (۱۹۹٥): ۱۰۹-۱۳۳.

۷. Zhang, Hongbin, and Guangyu Sun. "Feature selection using tabu search method." *Pattern recognition* ۳٥,۳ (۲۰۰۲): ۷۰۱-۷۱۱.

۸. Oh, Il-Seok, Jin-Seon Lee, and Byung-Ro Moon. "Hybrid genetic algorithms for feature selection." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* ۲٦,۱۱ (۲۰۰٤): ۱٤۲٤-۱٤۳۷.

۷- پانویس‌ها

[۱] Filtering Methods

[۲] Wrapper Methods

[۳] Criterion Function

[٤] Restricted Candidate List