واکشی قوانین دسته‌بندی با استفاده از الگوریتم ژنتیک

امین عینی‌پور[۱]؛ عبدالنبی انصاری اصل[۲]

چکیده

امروزه به دلیل زیاد بودن حجم داده‌ها و پیچیدگی آنها و نیاز بشر به دانش نهفته درآنها، استفاده از روشی کارآمد امری ضروری به نظر می‌رسد. داده‌کاوی فرایندی است که ما را در کشف چنین دانشی یاری می‌دهد و اخیراً در زمینه‌های گسترده‌ای مورد استفاده قرار گرفته است. در این مقاله یک روش داده‌کاوی کارآمد برای واکشی دانش از مجموعه داده‌های ورودی ارائه می‌شود. داده‌کاوی شامل دو مرحله اصلی است؛ ابتدا داده‌های خام مورد پردازش قرار گرفته و به قالبی مورد استفاده برای فرایند داده کاوی تبدیل می‌شوند سپس داده‌های بدست آمده از مرحله قبل، به کمک اعمالی نظیر دسته‌بندی و خوشه‌بندی به منظور تشخیص الگو مورد استفاده قرار می‌گیرند. روش پیشنهادی مطرح شده، به کمک الگوریتم ژنتیک، مجموعه‌ای از قوانین if-then را ایجاد کرده تا به کمک آنها عمل دسته‌بندی انجام شود. این الگوریتم که بر اساس ایده‌های بیولوژیکی بنا نهاده شده یکی از روش‌های موجود در الگوریتم‌های تکاملی است و در طی سال‌های اخیر در سیستم‌های داده‌کاوی و هوش مصنوعی نیز مورد استفاده قرار گرفته است. الگوریتم ژنتیک پیشنهادی، با توجه به معیارهایی نظیر دقت و قابلیت تفسیر، به جستجوی مجموعه‌ای از قوانین if-then در فضای حالت مربوط به مجموعه قوانین می‌پردازد که بهترین کارایی را داشته باشد و به کمک مکانیزم‌هایی از راه حل‌های بهینه محلی می‌گریزد. نتایج حاصل از پیاده‌سازی نرم‌افزاری این روش بر روی مجموعه داده‌های UCI نشان می‌دهد که روش پیشنهادی در مقایسه با روش‌های معروف در این زمینه از کارایی مطلوبی برخوردار است.

کلمات کلیدی

الگوریتم ژنتیک؛ داده‌کاوی؛ دسته‌بندی؛ تشخیص الگو

Discovering of classification rules using Genetic Algorithm

Amin Einipour; Abdolnabi Ansari Asl

Abstract

Nowadays because of massive complex volume of data and due to the fact they are in, that need of human because of their hidden nature, applying them is very urgent. Thus data mining is a process that can come to our help to discover such knowledge. Recently it has been used in many fields. In this paper an efficient data mining has been put forth to discover the knowledge using input data set. Data mining has two basic steps. First, the raw data are preprocessed and then are transformed into applicable format to be used for data mining. Then the generated data from previous step by making use of classification, clustering are used for pattern recognition. The proposed method performs the classification task and extracts required knowledge using Genetic Algorithm based systems which consist of if-then rules. Genetic Algorithm is based on biological ideas which are one of the present evolutionary algorithms. During recent years it has been used in the data mining and artificial intelligence. Proposed Genetic Algorithm, because of their accuracy and also interpretability in detecting a set of if-then rules in the related space it process the rules that has the best efficiency, also by means of a mechanism it escapes the local optima. The results of software implementation of this method on the UCI dataset show that the proposed method in comparison of well known method has more efficiency.

Keywords

Genetic Algorithm, Data mining, Classification, pattern recognition

۱- مقدمه

امروزه به دلیل زیاد بودن حجم داده‌ها و پیچیدگی آنها، ابزار مناسبی برای تحلیل این داده‌های موجود و دستیابی به دانش نهفته در آنها نیاز است. این تمایلات و علاقه‌ها برای حصول به دانش نهفته در داده‌ها باعث رشد چشمگیر داده‌کاوی شده است. داده‌کاوی در سال‌های اخیر، تأثیرات

[۱]- عضو هیأت علمی دانشگاه آزاد اسلامی واحد اندیمشک a.einipour@gmail.com

۲- دانشجوی کارشناسی ارشد دانشگاه آزاد اسلامی واحد علوم و تحقیقات اهواز

شگرفی در محیط‌های آکادمیک و صنعتی ایجاد کرده و کاربردهای فراوانی در زمینه‌های مختلف یافته است. به عنوان نمونه می‌توان به کاربردهای تجاری، مدیریت و کشف فریب، پزشکی، ورزشی، متن کاوی و وب کاوی اشاره نمود [۱]. اصطلاح داده‌کاوی به تمام جنبه‌های یک فرایند خودکار و یا نیمه خودکار برای استخراج دانش و الگوهای ناشناخته و سودمند از پایگاه‌های داده‌ای بزرگ اشاره می‌کند. این فرایند از دو مرحله اصلی تشکیل شده است؛ مرحله اول پیش پردازش داده‌ها است که شامل پاکسازی، یکپارچه‌سازی، انتخاب صفات و تبدیل داده‌ها به قالب مورد استفاده برای داده‌کاوی، است. در مرحله دوم، داده‌های بدست آمده از مرحله اول به منظور تشخیص الگو مورد استفاده قرار می‌گیرند که این امر به کمک الگوریتم‌هایی نظیر دسته‌بندی و خوشه‌بندی صورت می‌گیرد. سپس الگوهای بدست آمده بر اساس یک سری از معیارها نظیر دقت و قابلیت تفسیر دانش مورد ارزیابی قرار می‌گیرند. در این مرحله برای کشف الگو یک مدل یادگیری ایجاد می‌شود که این مدل به مرور زمان و با تکرار فرایند داده کاوی بهبود می‌یابد. خروجی این مرحله دانش کسب شده است که به کمک ابزارهایی نمایش داده می‌شود.

عمل دسته‌بندی شامل قرار دادن هر نمونه(شی یا رکورد) در یک کلاس از مجموعه کلاس‌های از پیش تعیین شده است. این کار بر اساس مقادیر برخی ویژگی‌های مربوط به نمونه مورد نظر انجام می‌شود. همانطور که اشاره شد، هدف کلی در داده‌کاوی کشف دانشی است که علاوه بر صحیح بودن باید قابل درک و مورد علاقه کاربر باشد. ازاین‌رو، کاربر می‌تواند نتایج تولید شده توسط سیستم را درک کرده و آنها را با دانش خود ترکیب کرده و به جای اعتماد کورکورانه به یک سیستم با نتایج تولیدشدهٔ غیر قابل تفسیر ، یک تصمیم مطلع و با بصیرت را اخذ کند.

در داده‌کاوی، دانش کشف شده غالباً به‌شکل قوانین پیش‌گویانه یا قوانین دسته‌بندیِ اگر-آنگاه(IF-THEN) و به‌صورت زیر نمایش داده می‌شوند:

IF <condition> THEN <class>

بخش <condition> یا مقدم قانون، شامل ترکیب منطقی از ویژگی‌های پیشگویی‌کننده و به شکل term۱ AND term۲ AND… است. هر term یا عبارت، شامل یک سه‌تایی به‌شکل <attribute, operator, value> است، مثل <Gender=female>. بخش <class> یا نتیجه قانون شامل کلاس پیش‌بینی شده برای نمونه‌هایی است که ویژگی‌های پیش‌بینی‌کنندهٔ آن، بخشِ <condition> قانون را برآورده می‌کند.

سیستم‌های مبتنی بر قوانین if-then در زمینه‌های کاربردی زیادی با موفقیت مورد استفاده قرار گرفته‌اند. اخیراً روش‌های متنوعی برای تولید و اصلاح این قوانین به صورت خودکار پیشنهاد شده‌اند. یکی از این روش‌ها الگوریتم‌های ژنتیک است [۲].الگوریتم ژنتیک هم به صورت تئوری و هم به صورت تجربی ثابت کرده که می‌تواند به جستجوی قابلیت‌های موجود در فضاهای جستجوی پیچیده پرداخته و یک روش معتبر را برای مسائلی که نیازمند جستجوی کارا و مؤثری هستند، ارائه کند. الگوریتم‌های ژنتیک[۳] به عنوان ابزارهای تولید و اصلاح قانون در طراحی سیستم‌های مبتنی بر قانون به کار می‌روند[۴-۹]. الگوریتم‌های ژنتیک در طراحی سیستم‌های قانون‌مند را اغلب روش‌های genetic-based machine learning (GBML) می‌نامند که به دو دسته عمده Pittsburgh و Michigan تقسیم می‌شوند[۱۰]. درروش Pittsburgh مجموعه‌ای از قوانین if-then در یک قالب رشته کد می‌شوند[۱۱-۱۲] (هر individual یک سیستم مبتنی بر قانون است) در حالی که در روش Michigan یک قانون if-then به‌صورت یک رشته کد می‌شود[۸,۹] (هر individual یک قانون if-then است).

در روش Pittsburgh کارایی هر مجموعه از قوانین (هر individual) به عنوان درجه شایستگی آن در نظر گرفته می‌شود. از آنجایی که یک جمعیت (نسل) شامل تعدادی مجموعه و هر مجموعه نیز شامل تعدادی قانون است، لذا زمان اجرای طولانی مدت و فضای حافظه فراوانی باید مصرف شود. از سوی دیگر در روش Michigan یک قانون if-then در قالب یک رشته کد و به عنوان یک individual در نظر گرفته می‌شود و کارایی یک قانون به عنوان درجه شایستگی آن مورد استفاده قرار می‌گیرد. در این روش کارایی مجموعه قانون جاری در کل ارزیابی نمی‌شود، بلکه کارایی قوانین به صورت تک تک بررسی می‌گردد [۲]. به این ترتیب و با توجه به ویژگی‌های این دو روش به نظر می‌رسد که ترکیبی از این دو روش بتواند کارایی مطلوبی را فراهم کند که توانایی‌های هر دو روش را به کار برد.

روش ارائه شده در این مقاله که روشی ترکیبی محسوب می‌شود، اساس را بر الگوریتم Pittsburgh قرار می‌دهد. همان طور که گفته شد هر individual یک مجموعه قانون است. در طی اجرای الگوریتم، mutation به معنای جهشی است که یک رشته یا individual را که نمایانگر یک مجموعه قانون است تغییر می‌دهد و از آنجایی که هدف بهینه کردن قوانین مجموعه‌ها است، برای mutation در یک مجموعه قانون، الگوریتم Michigan بر روی آن مجموعه اجرا می‌شود. به عبارتی دیگر از الگوریتم Michigan به عنوان عملگر جهش در الگوریتم Pittsburgh استفاده می‌شود. به این ترتیب از مزایای هر دو روش استفاده می‌شود و در نهایت نتایجی که از این الگوریتم‌های ترکیبی به دست می‌آیند بهتر از اجرای تک تک هر یک از این دو الگوریتم است.

در ادامه و در بخش دوم این مقاله، الگوریتم ژنتیک معرفی می‌شود؛ بخش سوم، الگوریتم ژنتیک پیشنهادی را تشریح می‌کند؛ در بخش چهارم، نتایج و ارزیابی روش پیشنهادی ارائه شده و در بخش پنجم نیز خلاصه و نتیجه‌گیری بحث می‌آید؛ نهایتاً مراجع مورد استفاده نیز در بخش پایانی مقاله آورده شده است .

۲- معرفی الگوریتم ژنتیک

استفاده از اصل زیست‌شناسی تکامل طبیعی در سیستم‌های مصنوعی، بیش از چهار دهه قبل ارائه شد که در چند سال اخیر رشد مؤثر و چشمگیری از خود نشان داده است. معمولاً در بررسی اصطلاحات الگوریتم‌ها یا محاسبات تکاملی به دامنه‌های تحقیقاتی از قبیل الگوریتم‌های ژنتیک، استراتژی‌های تکامل، برنامه‌نویسی تکاملی و برنامه‌نویسی ژنتیک برخورد می‌کنیم[۱۵-۱۳]. چنین الگوریتم‌هایی که امروزه متداول شده‌اند در مسائل زیادی از دامنه‌های مختلف از قبیل مسائل بهینه‌سازی، برنامه‌نویسی خودکار، یادگیری ماشین، اقتصاد، پزشکی و طراحی سخت‌افزار به طور موفقیت‌آمیز مورد استفاده قرار گرفته‌اند. در این مقاله از الگوریتم‌های ژنتیک به عنوان روش تکاملی مورد نظر استفاده می‌شود.

الگوریتم ژنتیک یک روال تکراری است که شامل جمعیتی از افراد یا کروموزوم‌ها (individuals) است که هر یک توسط رشته‌ای از نمادها تحت عنوان ژن در یک راه حل ممکن در فضای مسأله داده شده کدگذاری می‌شود. منظور از این فضا، فضای جستجویی است که شامل تمام راه‌حل‌های ممکن در مسأله مورد بررسی است. الگوریتم‌های ژنتیک معمولاً در فضاهایی که برای جستجو بسیار وسیع هستند به کار می‌روند. نمادهای الفبایی مورد استفاده غالباً دودویی هستند اما در سال‌های اخیر از کدگذاریهای مبتنی بر کاراکتر، کدگذاریهای مقادیر واقعی، نمایش درختی و نمایش‌های دیگر نیز استفاده شده است[۱۵] .

الگوریتم ژنتیک استاندارد معمولاً به این صورت ارائه می‌شود که یک جمعیت اولیه از individualها به‌صورت تصادفی یا اکتشافی تولید می‌شود. در هر مرحله از تکامل که یک نسل نامیده می‌شود، individualهای موجود در جمعیت جاری رمزگشایی شده و بر اساس یکسری معیارها از قبل تعیین شده مثل شایستگی یا تابع شایستگی تکامل می‌یابند. برای تشکیل یک جمعیت جدید(نسل بعدی)، individualها بر اساس شایستگی مورد نظر انتخاب می‌شوند. اخیراً روال‌های انتخاب زیادی مورد استفاده قرار گرفته‌اند که یکی از ساده‌ترین آنها انتخاب شایستگی نسبی است که در آن individualها با یک احتمال نسبی و متناسب با شایستگی مربوطه انتخاب می‌شوند. این روش اطمینان می‌دهد که در تعداد زمان‌های مورد انتظار یک individual به طور تقریبی متناسب با کارایی مربوطه در جمعیت انتخاب می‌شود؛ بنابراین individualهای با بهترین شایستگی شانس بیشتری برای قرار گرفتن در نسل تولیدی جدید دارند در حالیکه individualهای با درجه شایستگی پائین شانس کمتری دارند.

تابع انتخاب(selection) به تنهایی نمی‌تواند individualهای جدید را در جمعیت تولید کند، به عبارت دیگر این تابع نمی‌تواند نقاط جدیدی را در فضای جستجو پیدا کند. این کار توسط عملگرهای ژنتیکی انجام می‌شود که معروف‌ترین آنها عملگر تبادل(crossover) و عملگر جهش(mutation) هستند. عملگر تبادل با احتمال P_C بر روی دو individual انتخاب شده(والدین) و با تبادل بخش‌هایی از ژن‌ها برای ایجاد دو individual جدید(فرزند) انجام می‌شود. در ساده‌ترین حالت بعد از انتخاب احتمالی نقطه تبادل، زیررشته‌ها مبادله می‌شوند. این عملگر فرآیند تکامل را قادر می‌سازد که به سمت نواحی مورد نظر و امیدوارکننده موجود در فضای جستجو حرکت کند. عملگر جهش برای جلوگیری از همگرایی زودرس به سمت بهینه محلی معرفی می‌شود که بصورت تصادفی و با احتمال P_M انجام می‌شود که معمولاً مقدار این احتمال یک مقدار کوچک در نظر گرفته می‌شود. الگوریتم‌های ژنتیک جزء فرآیندهای تصادفی تکراری هستند که لزوماً همگرایی را تضمین نمی‌کند اما با در نظر گرفتن راهکارهایی می‌توان این احتمال را افزایش داد که سعی شده در روش ارائه شده در این مقاله این کار انجام شود. شرط توقف تکرارهای این الگوریتم نیز می‌تواند توسط برخی مقادیر ثابت از پیش تعیین شده مثل بیشترین تعداد نسل‌ها یا رسیدن به سطح شایستگی قابل قبول مشخص شود. شکل ۱ الگوریتم ژنتیک استاندارد را به صورت شبه کد نشان می‌دهد.

```
Begin GA
  g:=۰ {generation counter}
  Initialize population p(g)
  Evaluate population p(g) {i.e. , compute fitness value}
  While not done do
    g:= g+۱
    Select p(g) from p(g-۱)
    Crossover p(g)
    Mutate p(g)
    Evaluate p(g)
  End While
End GA
```

شکل(۱) شبه کد الگوریتم ژنتیک استاندارد

الگوریتم‌های ژنتیک در طراحی سیستم‌های قانون‌مند به دو دسته Pittsburgh و Michigan تقسیم می‌شوند. در روش Pittsburgh کارایی هر مجموعه از قوانین (هر individual) به عنوان درجه شایستگی آن در نظر گرفته می‌شود. بنابراین جستجو به دنبال مجموعه‌های با کارایی بالاتر معادل است با جستجو به دنبال سیستم مبتنی بر قانون کارامدتر. تعدادی از مجموعه قوانین بدون هیچ تغییری به عنوان individualهای ممتاز از جمعیت کنونی به نسل بعدی منتقل می‌شوند. در روش Pittsburgh یک قانون در داخل مجموعه خود حائز اهمیت است. چه بسا که قوانین خوبی

در مجموعه ضعیفی قرار بگیرند و در روند به روز رسانی یک نسل، نادیده گرفته شوند. از آنجایی که یک جمعیت (نسل) شامل تعدادی مجموعه و هر مجموعه نیز شامل تعدادی قانون است، لذا زمان اجرای طولانی مدت و فضای حافظه فراوانی باید مصرف شود.

از سوی دیگر در روش Michigan یک قانون if-then در قالب یک رشته کد و به عنوان یک individual در نظر گرفته می‌شود و کارایی یک قانون به عنوان درجه شایستگی آن مورد استفاده قرار می‌گیرد. بنابراین فقط یک سیستم از قوانین وجود دارد که در آن به جستجوی قوانین کارآمدتر و اصلاح قوانین ضعیف‌تر پرداخته می‌شود. در اینجا نیز تعدادی از قوانین بدون هیچ تغییری به عنوان individualهای ممتاز به نسل بعدی منتقل می‌شوند. کارایی مجموعه قانون جاری در کل ارزیابی نمی‌شود، بلکه کارایی قوانین به صورت تک تک بررسی می‌گردد. بنابراین چه بسا که مجموعه قانون کنونی خوب باشد و پس از به روز رسانی قوانین آن، اگرچه قوانین قوی‌تری جایگزین می‌شوند ولی کارایی مجموعه در کل نزول یابد. از آنجایی که در روش Michigan جمعیت مورد بررسی در هر لحظه فقط شامل تعدادی قانون ساده است، زمان محاسبات و فضای حافظه مورد نیاز بسیار کمتر از روش Pittsburgh است. الگوریتم Michigan بر تکامل قوانین و بهینه‌تر کردن آنها توجه دارد و کلاً قوانین اصلاح می‌شوند و نه لزوماً مجموعه قانون. این در حالی است که در الگوریتم Pittsburgh در هر نسل مجموعه‌های با شایستگی بالاتر مورد توجه بیشتر قرار دارند ولی به کیفیت قوانین درون یک مجموعه به تنهایی و آن گونه که در الگوریتم Michigan مورد توجه است، مستقیماً توجه نمی‌شود. استفاده از امکانات این دو روش در الگوریتم‌های ترکیبی به دست می‌آید که اساس کار ارائه شده در این مقاله است. همانطور که اشاره شد در روش Michigan هر individual به صورت یک قانون منفرد کدگذاری می‌شود. الگوریتم ژنتیک مورد استفاده قرار می‌گیرد تا یک قانون منفرد را تولید کند، بنابراین این روش یک راه حل جزئی محسوب می‌شود. الگوریتم ژنتیک به صورت تکراری و برای کشف قوانین جدید مورد استفاده قرار می‌گیرد تا زمانیکه یک قانون مناسب تولید شود. برای جلوگیری از تولید قوانین تکراری(قوانین با بخش مقدم تکراری) یک شِمای جریمه‌ای در هر بار که قانون جدید اضافه می‌شود، اعمال می‌گردد. در روش ترکیبی مورد نظر سرعت روش Michigan با سادگی ارزیابی شایستگی در روش Pittsburgh ترکیب می‌شود که در مقایسه با سایر روش‌های ساخت افزایشی قوانین، این روش به افراز بهینه‌ای در فضای مقدم منجر می‌شود.

۳- الگوریتم ژنتیک پیشنهادی

در این قسمت ابتدا روش کدگذاری قوانین که در این مقاله استفاده شده تشریح می‌شود سپس مراحل اجرای الگوریتم پیشنهادی به تفصیل بررسی می‌شود. هر قانون if-then به صورت یک رشته کد می‌شود. علامت‌های مورد استفاده شامل چهار مقدار زبانی است که در جدول شماره ۱ قابل ملاحظه است. به عنوان مثال قانونی که به صورت (A_3,A_2,A_4,A_1) کد شده باشد بهصورت زیر تفسیر می‌شود:

If $0.5 \le x_1 < 0.75$ AND $0.25 \le x_2 < 0.5$ AND $0.75 \le x_3 < 1$ AND $0 \le x_4 < 0.25$ THEN class C_j with $C_F = C_{Fj}$

جدول(۱) نمادهای مورد استفاده در کدگذاری قوانین

نماد	بازه مورد نظر
A_1	$0 \le x < 0.25$
A_2	$0.25 \le x < 0.5$
A_3	$0.5 \le x < 0.75$
A_4	$0.75 \le x < 1$

شکل کلی الگوریتم پیشنهادی مبتنی بر الگوریتم Pittsburgh است که در طول اجرا به نحوی از روش Michigan هم استفاده می‌کند. مراحل اصلی این روش را می‌توان به صورت زیر بیان کرد:

مرحله ۱) تولید جمعیت اولیه از individualها که شامل مجموعه‌ای از قوانین اولیه هستند و با استفاده از نمادهای مربوطه کد شده‌اند.

مرحله ۲) ارزیابی میزان شایستگی جمعیت اولیه با استفاده از تابع شایستگی مورد نظر.

تکرار مراحل زیر و به تعداد مشخص و برگرداندن بهترین مجموعه قوانین:

مرحله ۳) استفاده از عملگر selection برای انتخاب قوانین والد و تولید قانون جدید از روی آنها(تولید نسل جدید قوانین).

مرحله ۳-۱) استفاده از عملگر تبادل با احتمال P_C بر روی قوانین انتخاب شده.

مرحله ۳-۲) استفاده از عملگر جهش با احتمال P_M بر روی قوانین انتخاب شده.

مرحله ۴) محاسبه میزان شایستگی قانون جدید با استفاده از تابع شایستگی محلی مربوطه.

مرحله ۵) محاسبه میزان شایستگی مجموعه قوانین جدید با استفاده تابع شایستگی سراسری مورد نظر.

مرحله ۶) قرار دادن مجموعه قوانین جدید در مجموعه قوانین فعلی به شرط بیشتر بودن مقدار شایستگی نسل جدید از شایستگی نسل قبلی قوانین و پذیرفتن نسل جدید با یک احتمال خاص.

در ادامه این مراحل با تفصیل بیشتر مورد بررسی قرار می‌گیرند.

۳-۱- تولید جمعیت اولیه(نسل اول قوانین)

همانطور که در بخش‌های قبلی اشاره شد جمعیت مورد نظر در اینجا مجموعه‌ای از individualها است که هر individual نشان دهنده یک قانون if-then است. در این مرحله باید به نحوی مجموعه‌ای از این قوانین را ایجاد کرد. در این مقاله از روشی احتمالی برای ایجاد نسل اول قوانین استفاده می‌شود. در بخش قبل اشاره شد که می‌توان هر قانون را به صورت کدگذاری خاصی در نظر گرفت. در روش تصادفی مورد نظر، با نسبت دادن تصادفی نمادهای A_i ، بخش مقدم قوانین به صورت تصادفی ایجاد می‌شود. استفاده از روش تصادفی به این دلیل حائز اهمیت است که بر روی هر مسأله و با هر تعداد مقدار زبانی مقدم و دسته نتیجه به خوبی اجرا شده و نتایج خوبی را نیز حاصل می‌کند[۱۶]. بعد از انتخاب تصادفی بخش مقدم قوانین، با استفاده از روال‌های معروفی که در این زمینه وجود دارد دسته نتیجه و درجه قطعیت هر یک از قوانین نیز مشخص می‌شود. این روال‌ها در قالب معادلاتی و به صورت زیر بیان می‌شوند:

- محاسبه درجه سازگاری هر نمونه آموزشی $x_p = (x_{p1}, x_{p2}, ..., x_{pn})$ با قانون if-then R_j که به کمک عمل حاصلضرب بدست می‌آید.

$$\mu_{R_j}(x_p) = \mu_{A_{j1}}(x_{p1}) \times ... \times \mu_{A_{jn}}(x_{pn}), \quad p = 1, 2, ..., m \tag{۱}$$

که $\mu_{A_{ji}}(.)$ عضویت A_{ji} را در بازه مربوطه در جدول ۱ نشان می‌دهد. p تعداد نمونه‌های آموزشی و n تعداد صفات هر نمونه است.

- محاسبه مجموع درجه‌های سازگاری برای هر دسته:

$$\beta_{Class\ h}(R_j) = \sum_{x_p \in Class\ h} \mu_{R_j}(x_p) \quad ,h = 1, 2, ..., c \tag{۲}$$

c تعداد دسته‌ها است.

- یافتن دسته نتیجه C_j که بیشترین مقدار $\beta_{Class\ h}(R_j)$ را در بین c دارد.

$$\beta_{Class\ C_j}(R_j) = \max\{\beta_{Class\ 1}(R_j), ..., \beta_{Class\ c}(R_j)\}. \tag{۳}$$

اگر بیش از یک دسته دارای بیشترین مقدار باشند آنگاه دسته نتیجه به صورت یکتا مشخص نمی‌شود و در این حالت یک مقدار تهی به عنوان دسته نتیجه قانون در نظر گرفته می‌شود.

- هنگامی که دسته نتیجه C_j به کمک رابطه (۳) تعیین شد، درجه قطعیت به صورت زیر مشخص می‌شود:

$$CF_j = \frac{\beta_{Class\ C_j}(R_j) - \overline{\beta}}{\sum_{h=1}^{c} \beta_{Class\ h}(R_j)} \tag{۴}$$

و $\overline{\beta}$ از رابطه زیر بدست می‌آید:

$$\overline{\beta} = \frac{\sum_{h \neq C_j} \beta_{Class\ h}(R_j)}{c - 1} \tag{۵}$$

۳-۲- تولید نسل جدید قوانین

در این قسمت نحوه ساخت نسل جدید قوانین از روی نسل قبلی قوانین توضیح داده می‌شود. برای ساخت نسل جدید قوانین، یک جفت قانون از میان قوانین if-then جمعیت کنونی(نسل کنونی) انتخاب می‌شود تا از روی آنها قوانین if-then جدید برای نسل بعدی تولید شوند. هر قانون if-then در جمعیت کنونی توسط یک روال انتخاب دوره‌ای و تصادفی که در ادامه توضیح داده می‌شود انتخاب و بروز می‌شود. عملگر تبادل با یک

نرخ تبادل از قبل تعیین شده بر روی جفت قانون انتخاب شده تصادفی اعمال می‌شود. باید توجه داشت که قوانین انتخاب شده برای عملگر تبادل نباید یکسان باشند. بعد از انجام عملگر تبادل، دسته‌های نتیجه قوانین جدید نیز مشخص می‌شوند. اگر این دسته‌ها با دسته‌های قوانین والد یکسان باشند،آنگاه قوانین جدید حاصله از عملگر تبادل پذیرفته می‌شود، در غیر اینصورت و تا زمانیکه دسته نتیجه قانون جدید مشابه والدین خود باشد یا تا زمانیکه تعداد تکرارها از حد مجاز خارج نشود عملگر تبادل تکرار می‌شود. بعد از اعمال عملگر تبادل، نوبت به عملگر جهش می‌رسد به این صورت که با یک نرخ جهش مشخص، بخش مقدم قانونی که به صورت تصادفی اتخاب شده و عملگر تبادل نیز بر روی آن اعمال شده با مقادیر زبانی جدید(نمادهای تعریف شده در جدول شماره ۱) تغییر می‌کند. بعد از انجام عملگر جهش، دسته نتیجه قانون جهش یافته نیز مشخص می‌شود. اگر دسته نتیجه قانون جهش یافته مشابه دسته نتیجه قانون قبل از جهش باشد، قانون جهش یافته مورد پذیرش قرار می‌گیرد، در غیر اینصورت می‌توان تا زمانیکه تعداد تکرارها از حد مجاز بیشتر نشده باشد عملگر جهش را تکرار کرد تا شاید دسته نتیجه قانون جهش یافته مشابه دسته نتیجه قبل از جهش شود. روال توضیح داده شده که شامل عملیات تبادل و جهش است و به روال تولید نسل معروف است تا زمانیکه تعداد جفت قانون‌های if-then تولید شده به اندازه از قبل تعیین شده نرسیده باشد یعنی تا زمانیکه تعداد نسل‌ها از حد بیشینه مشخص شده بیشتر نشود، تکرار می‌شود.

۳-۳- نحوه ارزیابی قوانین

یکی از مهم‌ترین مراحل موجود در این الگوریتم مرحله ارزیابی مجموعه قوانین است که هم بر روی مجموعه قوانین ابتدایی و هم بر روی مجموعه قوانین تولیدی جدید انجام می‌شود. بعد از اینکه با عملگرهای معروف ژنتیک از قبیل عملگرهای انتخاب، تبادل و جهش، قانون جدیدی تولید شد باید ارزیابی کرد که قانون جدید باعث بهبودی کارایی سیستم می‌شود یا خیر. برای ارزیابی شایستگی از دو نوع ارزیابی استفاده می‌شود؛ یک نوع تابع به ارزیابی شایستگی تنها قانون تولیدی جدید مربوط می‌شود و نوع دیگر ارزیابی، شایستگی مجموع قوانین جدید را نشان می‌دهد. در ادامه به نحوه انجام این نوع ارزیابی‌ها پرداخته می‌شود.

در ابتدا به نحوه ارزیابی شایستگی و کیفیت تک قانون تولید شده اشاره می‌شود. کیفیت قانون ساخته شده توسط نمادی به نام Q و به‌وسیله فرمول Q=sensitivity * specificity محاسبه می‌شود که به‌صورت معادله (۶) تعریف می‌شود:

$$Q = \left(\frac{TruePos}{TruePos + FalseNeg}\right) \times \left(\frac{TrueNeg}{FalsePos + TrueNeg}\right) \qquad (۶)$$

که در آن:

- TruePos (true positive) ، تعداد نمونه‌های پوشش داده شده توسط قانون است که دارای دسته پیش‌بینی شده مشابه توسط این قانون هستند.
- FalsePos (false positive) ، تعداد نمونه‌های پوشش داده شده توسط قانون است که دارای دسته‌ای متفاوت از دسته پیش‌بینی شده توسط این قانون هستند.
- FalseNeg (false negative) ، تعداد نمونه‌هایی هستند که توسط قانون پوشش داده نشده‌اند در حالیکه دارای دسته‌ای پیش‌بینی شده توسط این قانون هستند.
- TrueNeg (true negative) ، تعداد نمونه‌هایی هستند که توسط قانون پوشش داده نشده‌اند و دارای دسته‌ای پیش‌بینی شده توسط این قانون نیز نیستند.

بیشتر بودن مقدار Q، کیفیت قانون را بالا می‌برد. باید توجه داشت که مقدار Q در بازهٔ [۰،۱] متغیر است.

بعد از اینکه کیفیت قانون جدید ارزیابی شد و مورد تایید قرار گرفت باید تأثیر این قانون را بر مجموعه کل قوانین نیز بررسی کرد تا بتوان دقت این قوانین را با دقت قوانین نهایی ارزیابی کرد.

تابع ارزیابی مجموعه قوانین به صورت زیر تعریف می‌شود:

$$EF(S) = W_{NNCP}.NNCP(S) + W_S.(|S|) + W_L.(Length(S)) \quad , (W_{NNCP} + W_S + W_L = 1). \qquad (۷)$$

در رابطه شماره (۷)، $NNCP(S)$ تعداد نمونه‌هایی است که توسط مجموعه قوانین به صورت اشتباه دسته بندی شده است و یا اصلا دسته‌بندی نشده‌اند. $NNCP(S)$ از رابطه زیر به دست می‌آید:

$$NNCP(S) = m - \sum_{R_j \in S} NCP(R_j) \quad (۸)$$

|S| اندازه یک مجموعه یا به عبارت دیگر، تعداد قوانین if-then موجود در مجموعه قوانین است. *Length(S)* مجموع طول قوانین موجود در مجموعه قوانین است. W_{NNCP} و W_S و W_L وزن‌هایی مثبتی هستند که برای این معیارها در نظر گرفته می‌شود. در صورتی که دقت و قابلیت دسته‌بندی مدل پیشنهادی از اهمیت بالایی برخوردار باشد، W_{NNCP} بسیار بیشتر از W_S و W_L در نظر گرفته می‌شود. در حالتی که سادگی، فشردگی و قابلیت تفسیر از درجه اهمیت بالاتری برخوردار باشد آنگاه پارامتر W_{NNCP} را تقریبا برابر با مجموع W_S و W_L در نظر می‌گیرند. در پیاده‌سازی، W_{NNCP} و W_S و W_L به ترتیب ۰.۹۹ و ۰.۰۰۵ و ۰.۰۰۵ در نظر گرفته شده است.

۴- نتایج روش پیشنهادی

الگوریتم ژنتیک پیشنهادی بر روی چهار مجموعه داده پزشکی از مخزن دادهٔ UCI (university of California at Irvin)، ارزیابی می‌شود. ویژگی‌های اصلی مجموعه داده‌های به کار رفته درارزیابی الگوریتم در جدول شماره (۲) خلاصه شده است. اولین ستون این جدول مجموعه داده مورد نظر را مشخص می‌کند در حالیکه سایر ستون‌ها به ترتیب، تعداد نمونه‌های موجود در آن مجموعه داده، تعداد ویژگی‌های با مقادیر گروهی، تعداد ویژگی‌های با مقادیر واقعی و تعداد دسته‌های نتیجه مجموعه داده مورد نظر را مشخص می‌کند.

جدول(۲) مجموعه داده‌های مورد استفاده در ارزیابی الگوریتم ژنتیک پیشنهادی

مجموعه داده	#Cases	#Nom.	#Real	#Cla.
سرطان سی‌نه (Ljubljana)	۲۸۳	۹	۰	۲
سرطان سی‌نه (Wisconsin)	۶۹۹	۰	۹	۲
هپاتی‌ت	۱۷۸	۱۳	۶	۲
بی‌ماری قلبی(Cleveland)	۳۰۳	۸	۵	۵

همانطور که در بخش سوم توضیح داده شد، برای اجرای یکسری از روال‌های الگوریتم پیشنهادی باید یکسری پارامترها را از قبل مشخص کرد. این پارامترها شامل اندازه جمعیت(تعداد قوانین)، نرخ تبادل، نرخ جهش و تعداد تکرارها برای پایان روال‌ها و همچنین پایان کل الگوریتم(تعداد نسل‌ها) است. این پارامترها را می‌توان در جدول شماره (۳) ملاحظه نمود.

جدول(۳) پارامترهای مشخص شده در اجرای الگوریتم پیشنهادی

نام پارامتر	مقدار
اندازه جمعی‌ت(N_P)	۲۰
نرخ تبادل(P_C)	۰.۹
نرخ جهش(P_M)	۰.۱
تعداد تکرار برای پای‌ان روالها	۱۰۰
تعداد نسل‌ها	۱۰۰۰

کارایی الگوریتم پیشنهادی در مقایسه با سه الگوریتم معروف در زمینه استنتاج قوانین در جدول شماره (۴) نشان داده شده است. اولین الگوریتم، الگوریتم C۴.۵ است[۱۷]. این الگوریتم بر پایه درخت‌های تصمیم گیری می‌باشد و از یک معیار مبتنی بر آنتروپی استفاده می‌نماید. همچنین از تکنیک‌های هرس کردن برای از بین بردن شاخه‌های اضافی استفاده می‌کند. الگوریتم دوم، الگوریتم نزدیکترین همسایه یا k-NN است؛ روال این الگوریتم به این صورت است که برای هر نمونه جدید با مقایسه آن با k نمونه آموزشی نزدیکتر، دسته نتیجه را مشخص می‌کنند، بنابراین لازم است معیاری برای تعیین فاصله بین نمونه‌ها مشخص شود. برای تعیین فاصله بین دو نمونه از فاصله اقلیدسی استفاده می‌شود. نهایتا از الگوریتم XCS برای مقایسه استفاده می‌شود[۱۸]. این روش مکانیزمی مشابه الگوریتم Michigan دارد که در آن هرقانون

if-then فازی به صورت یک رشته کد می‌شود. نتایج حاصل از مقایسه روش پیشنهادی با سه روش مذکور در جدول شماره (۴) مشخص شده است. البته باید توجه داشت که در انجام آزمایش هم نتایج مجموعه آموزشی(train set) و هم نتایج مجموعه آزمایشی(test set) بدست می‌آید که با توجه به اهمیت مجموعه آزمایش و اینکه دقت نهایی هر سیستم از روی مجموعه آزمایشی آن به دست می‌آید لذا در جدول زیر فقط دقت مجموعه آزمایشی آورده شده است.

جدول(۴) پارامترهای مشخص شده در اجرای الگوریتم پیشنهادی(نتایج به درصد بیان شده)

روش	دقت مجموعه آزمایشی (سرطان سی‌نه L)	دقت مجموعه آزمایشی (سرطان سی‌نه W)	دقت مجموعه آزمایشی (هپاتیت)	دقت مجموعه آزمایشی (بی‌ماری قلبی)
C۴.۵	۷۵.۰۰	۹۵.۴۴	۸۶.۱۳	۵۹.۲۴
k-NN	۷۶.۵۹	۹۶.۹۹	**۹۶.۶۵**	۶۲.۵
XCS	۹۵.۱۰	۹۵.۸۷	۹۶.۲۵	۶۴.۷۸
Proposed (GA based)	**۹۷.۸۵**	**۹۷.۲۳**	**۹۶.۴۲**	**۶۸.۹۶**

این نتایج با استفاده از زبان برنامه‌نویسی C++ و بر روی یک PC با مشخصات Pentium IV و کلاک ۱.۸۰MHZ و حافظهٔ اصلی ۱GB بدست آمده است. با توجه به دقت‌های مختلفی که در اجراهای مختلف بدست می‌آید لذا برای توزیع عادلانه هر مجموعه داده، از روش ۱۰CV استفاده می‌شود. در این روش هر مجموعه داده به ۱۰ نمونه آموزشی-آزمایشی مختلف تقسیم می‌شود، سپس با این ده زیر مجموعه مختلف، الگوریتم مربوطه به طور مستقل ۱۰ بار اجرا می‌شود و با میانگین گرفتن از نتایج بدست آمده نتیجه نهایی ثبت می‌شود. همان‌طور که در جدول نشان داده شده است دقت الگوریتم ژنتیک پیشنهادی در همه مجموعه داده‌ها از سایر روش‌ها بیشتر است به جز در مجموعه داده هپاتیت که دقت آن از روش نزدیکترین همسایه کمتر است.

۵- خلاصه و نتیجه‌گیری

در این مقاله بر اساس ایده تکاملی ژنتیک، الگوریتمی ارائه شد که با تولید مجموعه‌ای قوی از قوانین if-then، می‌تواند با دقت بالایی به دسته‌بندی مجموعه داده‌های مختلف بپردازد؛ به این ترتیب که از مجموعه‌ای اولیه و با کیفیت پایین از قوانین به مجموعه‌ای بهینه و با کیفیت بالا می‌رسد که با دقت قابل قبولی به دسته‌بندی نمونه‌های مختلف می‌پردازد . الگوریتم‌های ژنتیکی که در این زمینه مورد استفاده قرار می‌گیرند در دو دسته معروف Michigan و Pittsburgh قرار دارند. نقطه قوت الگوریتم پیشنهادی در ترکیب مناسب این دو روش نهفته است به طوریکه از ویژگی‌ها و نقاط قوت محلی و سراسری بودن هر دو الگوریتم استفاده می‌کند. در نهایت الگوریتم پیشنهادی بر روی چهار مجموعه داده از مخزن داده UCI مورد آزمایش قرار گرفت و در مقایسه با الگوریتم‌های معروف دیگری که در این زمینه مورد استفاده قرار گرفته اند نتایج امیدوارکننده‌ای حاصل شد. با توجه به نتایج خوبی که در این روش ترکیبی حاصل شد به نظر می‌رسد که استفاده از ترکیب الگوریتم‌های مختلف در این زمینه امیدوارکننده به نظر برسد؛ به خصوص در مورد ترکیب الگوریتم‌های تکاملی مختلف از قبیل الگوریتم‌های ژنتیک، مورچگان، لیست ممنوعه و غیره می‌توان پیش‌بینی کرد که در عمل نتایج خوبی حاصل شود.

٦- مراجع

[۱] Larose, Daniel T. *"Discovering knowledge in data: an introduction to data mining"*. John Wiley & Sons, Inc., ۲۰۰۵.

[۲] Ishibuchi H., Nakashima T., and Murata T. *"Three-objective genetics-based machine learning for linguistic rule extraction"*. Information Sciences, vol. ۱۳٦, no. ۱-٤, pp. ۱۰۹-۱۳۳, ۲۰۰۱.

[۳] J. H. Holland. *"Adaptation in Natural and Artificial Systems"*. University of Michigan Press, Ann Arbor, MI, ۱۹۷۵.

[٤] C. L. Karr, and E. J. Gentry. *"Fuzzy control of pH using genetic algorithms"* IEEE Trans. on Fuzzy Systems, vol. ۱, pp. ٤٦-۵۳, February, ۱۹۹۳.

[۵] Herrera, M. Lozano, and J. L. Verdegay. *"Tuning fuzzy logic controllers by genetic algorithms"*. International J. of Approximate Reasoning, vol. ۱۲, no. ۳/٤, pp. ۲۹۹-۳۱۵, April/May, ۱۹۹۵.

[٦] B. Carse, T.C. Fogarty, and A. Muntro. *"Evolving fuzzy rule based controllers using genetic algorithms"*. Fuzzy Sets and Systems, vol. ۸۰, no. ۳, ۳, pp. ۲۷۳-۲۹۳, June, ۱۹۹٦.

[۷] M. Valenzuela-Rendon. *"The fuzzy classifier system: A classifier system for continuously varying variables"*. Proc. Of ٤th International Conference on Genetic Algorithms, pp. ۳٤٦-۳۵۳, University of California, San Diego, CA, July, ۱۹۹۱.

[٨] H. Ishibuchi, and T. Nakashima. *"Improving the Performance of Fuzzy Classifier Systems for Pattern Classification Problems with Continuous Attributes"*. IEEE Transactions on Industrial Electronics, vol. ٤٦,no. ٦, December ١٩٩٩.

[٩] H. Ishibuchi, T. Nakashima and T. Muratam. *"Performance evaluation of fuzzy classifier systems for multi-dimensional pattern classification problem"*. IEEE Trans. On Systems, Man, and Cybernetics, October, ١٩٩٩.

[١٠] H. Ishibuchi, T. Nakashima, and T. Kuroda. *"A hybrid fuzzy genetics-based machine learning algorithm: hybridization of Michigan approach and Pittsburgh approach"*. In: Proceedings of ١٩٩٩ IEEE international conference on systems, man, and cybernetics, vol. ١,pp. ٢٩٦–٣٠١, Oct. ١٩٩٩.

[١١] S. F. Smith. *"A learning system based on genetic algorithms"*. Ph.D. Dissertation, University of Pittsburgh, Pittsburgh, PA, ١٩٨٠.

[١٢] B. Carse, T.C. Fogarty, and A. Muntro. *"Evolving fuzzy rule based controllers using genetic algorithms"*. Fuzzy Sets and Systems, vol. ٨٠, no. ٣, ٣, pp. ٢٧٣-٢٩٣, June, ١٩٩٦.

[١٣] Fogel DB. Evolutionary Computation. *"Toward a New Philosophy of Machine Intelligence.Piscataway"*. NJ: IEEE Press, ١٩٩٥.

[١٤] Koza JR. *"Genetic Programming"*. Cambridge, MA: MIT Press, ١٩٩٢.

[١٥] Michalewicz Z. *"Genetic Algorithms, Data Structures_Evolution Programs"*. ٣rd edition. Heidelberg: Springer-Verlag, ١٩٩٦.

[١٦] H. Ishibuchi, and T. Nakashima. *"Improving the Performance of Fuzzy Classifier Systems for Pattern Classification Problems with Continuous Attributes"*. IEEE Transactions on Industrial Electronics, vol. ٤٦, no. ٦, December ١٩٩٩.

[١٧] Quinlan, J.R. *"C٤.٥: Programs for Machine Learning"*. Morgan-Kaufmann, San Mateo, CA. ١٩٩٣.

[١٨] Sharpe, P.K., Glover, R.P. *"Efficient GA based technique for classification"*. Applied Intelligence ١١, ٢٧٧–٢٨٤, ١٩٩٩.