

استخراج قوانین با استفاده از CLA-EC

زهره گلشنی	فرشاد جعفریه	محمد رضا میبدی	وحید آزادگان
دانشگاه آزاد اسلامی واحد قزوین	دانشگاه آزاد اسلامی واحد قزوین	دانشگاه صنعتی امیرکبیر	دانشگاه عبدالرحمن صوفی رازی
Golshani_Zahra@yahoo.com	Jafarieh_Farshad@yahoo.com	Meybodi@ce.aut.ac.ir	V.Azadegan@gmail.com

(CLA)^۴ و محاسبات تکاملی^۵ (EC) می باشد جهت یافتن قوانین طبقه بندی استفاده خواهیم نمود.

قوانین IF-THEN در سلولهای که مقادیر باینری دارند و شامل کلیه ویژگیها در مجموعه داده^۶ می باشند، کد می شوند. وظیفه یافتن قوانین در یک مسئله بهینه سازی با اجزایی جهت رسیدن به دقت بالا و کارایی مناسب فرموله می گردد.

ادامه مقاله بدینصورت سازماندهی شده. بخش ۲ به تعریف سیستم خبره و بخشهای ۳ و ۴ و ۵ به ترتیب اتوماتای سلولی، اتوماتای یادگیر و اتوماتای یادگیر سلولی شرح داده شده است. در بخش ۶ به معرفی CLA-EC و سپس به نحوه نمایش قانون در بخش ۷ پرداخته شده و در بخش ۸ ارزیابی قانون مطرح شده است. نتایج مقاله در بخش ۹ آورده شده و بخش انتهایی نیز نتیجه گیری می باشد.

۲- تعریف سیستم خبره

سیستم های خبره، برنامه های کامپیوتری هوشمندی هستند که دانش و روشهای استنباط و استنتاج را بکار می گیرند تا مسائلی را حل کنند که برای حل آنها به مهارت انسانی نیاز است. در واقع سیستم های خبره کاربر را قادر به مشاوره با سیستم های کامپیوتری در مورد یک مسئله و یافتن دلایل بروز مسئله و راه حل های آن می کند. در این حالات مجموعه سخت افزار و نرم افزار تشکیل دهنده سیستم خبره، مانند فرد خبره اقدام به طرح سئوالات مختلف و دریافت پاسخ های کاربر، مراجعه به پایگاه دانش (تجربیات قبلی) و استفاده از یک روش منطقی برای نتیجه گیری و نهایتاً ارائه راه حل می نماید [۱].

سیستم های خبره برخلاف سیستم های اطلاعاتی که بر روی داده ها عمل می کنند، بر دانش^۷ متمرکز شده است.

۳- اتوماتای سلولی

چکیده: استخراج دانش یکی از مهمترین مسائل داده کاوی می باشد. قوانینی که بصورت اگر - آنگاه مطرح می گردند این توانایی را دارند که به صورت اعداد باینری در هر سلول - به صورت تمام مقادیری که در مجموعه داده می تواند وجود داشته باشد - قرار بگیرند. روش پیشنهادی در این مقاله استفاده از الگوریتم اتوماتای یادگیر سلولی برپایه محاسبات تکاملی (CLA-EC) برای استخراج قوانین می باشد. در روش پیشنهادی استخراج قوانین را به صورت یک مسئله بهینه سازی در آورده و هدف به دست آوردن قانونی با اطمینان، عمومیت و قابلیت درک بالا می باشد. الگوریتم پیشنهادی CLA-EC بر روی هر دو دسته داده ی پیوسته و طبقه بندی شده پیاده سازی شده و آزمایشات بر روی مجموعه داده ی ZOO با مقادیر طبقه بندی شده و مجموعه داده ی CANCER با مقادیر پیوسته تست گردید. نتایج نشان می دهند، مجموعه قوانینی با قابلیت درک و اطمینان بالا و بصورت مختصر تولید می گردند.

واژه های کلیدی: استخراج قوانین، CLA-EC، بهینه سازی، محاسبات تکاملی، مجموعه داده.

۱- مقدمه

سیستم های خبره^۱ به دسته ای خاص از نرم افزارهای رایانه ای اطلاق می شود که در راستای کمک به کاردانا و متخصصان انسانی و یا جایگزینی جزئی آنان در زمینه های محدود تخصصی تلاش دارند. این سامانه ها معمولاً اطلاعات را به شکل واقعیات^۲ و قواعد^۳ در محلی به نام پایگاه دانش به شکل ساختاری ذخیره نموده، و سپس با استفاده از روشهایی خاص از این داده ها نتایج مورد نیاز استنتاج می شود. در این مقاله از الگوریتم CLA-EC که ترکیبی از اتوماتای یادگیر سلولی

⁴ Cellular Learning Automata

⁵ Evolutionary Computing

⁶ Data set

⁷ Knowledge

¹ Expert systems

² Facts

³ Rules

یک مدل ریاضی برای سیستمهایی است که در آنها (CA) اتوماتای سلولی چندین مؤلفه ساده برای تولید الگوهای پیچیده با هم همکاری می‌کنند. در اتوماتای سلولی یک مجموعه منظم از سلولها وجود دارد که هر کدام می‌توانند با چند وضعیت مختلف مقداردهی شوند. برای هر سلول یک همسایگی از سلولها در نظر گرفته می‌شود. این سلولها در زمانهای گسسته بر طبق یک قانون محلی بهنگامرسانی می‌شوند. در تعیین وضعیت جدید برای هر سلول، مقادیر سلولهای همسایه نیز تأثیرگذار هستند.

۴- اتوماتاهای یادگیر

یک اتوماتای یادگیر یک ماشین با حالات محدود^۸ است که می‌تواند تعداد محدودی عمل را انجام دهد. هر عمل انتخاب شده توسط محیطی احتمالی ارزیابی می‌گردد و پاسخی به اتوماتای یادگیر داده می‌شود. اتوماتای یادگیر از این پاسخ استفاده نموده و عمل بعدی خود را انتخاب می‌کند. در طی این فرایند، اتوماتای یادگیر یاد می‌گیرد که چگونه بهترین عمل را از بین اعمال مجاز خود انتخاب نماید. محیط را می‌توان توسط سه تایی $E \equiv \{\alpha, \beta, c\}$ نشان داد که در آن $\alpha \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ مجموعه ورودیها، $\beta \equiv \{\beta_1, \beta_2, \dots, \beta_m\}$ مجموعه خروجیها و $c \equiv \{c_1, c_2, \dots, c_r\}$ مجموعه احتمالاتی جریمه می‌باشد. هرگاه β مجموعه دو عضوی باشد محیط از نوع P می‌باشد. در چنین محیطی $\beta_1 = 1$ به عنوان جریمه و $\beta_2 = 0$ به عنوان پاداش در نظر گرفته می‌شود. در محیط Q، $\beta(n)$ می‌تواند به طور گسسته یک مقدار از مقادیر محدود در فاصله $[0, 1]$ در محیط از نوع S، $\beta(n)$ هر مقدار در فاصله $[0, 1]$ را اختیار کند. c_i احتمال اینکه عمل α_i نتیجه نا مطلوب داشته باشد، می‌باشد. اتوماتای یادگیر به دو گروه اتوماتاهای یادگیر با ساختار ثابت و اتوماتاهای یادگیر با ساختار متغیر تقسیم می‌شوند. در این مقاله از اتوماتاهای یادگیر با ساختار متغیر استفاده شده است که در ادامه این بخش به شرح مختصری از آن می‌پردازیم.

اتوماتاهای یادگیر با ساختار متغیر^۹: اتوماتای یادگیر با ساختار متغیر توسط ۴ تایی $\{\alpha, \beta, p, T\}$ نشان داده می‌شود که در آن $\alpha = \{\alpha_1, \dots, \alpha_r\}$ مجموعه عملهای اتوماتا، $\beta = \{\beta_1, \dots, \beta_m\}$ مجموعه ورودیهای اتوماتا، $p = \{p_1, \dots, p_r\}$ بردار احتمال انتخاب هریک از عملها و $p(n+1) = T[\alpha(n), \beta(n), p(n)]$ الگوریتم یادگیری می‌باشد. در این نوع از اتوماتاهای یادگیر، اگر عمل α_i در مرحله nام انجام شود و پاسخ مطلوب از محیط دریافت نماید، احتمال $p_i(n)$ کاهش یافته و سایر احتمالات افزایش می‌یابند. در هر حال، تغییرات به گونهای صورت می‌گیرد تا حاصل جمع $p_i(n)$ ها

⁸ Finite State Machine

⁹ Variable structure

همواره ثابت و مساوی یک باقی بماند. الگوریتم زیر یک نمونه از الگوریتم های یادگیر خطی است.

الف - پاسخ مطلوب

$$p_i(n+1) = p_i(n) + a[1 - p_i(n)]$$

$$p_j(n+1) = (1-a)p_j(n) \quad \forall j \neq i$$

ب - پاسخ نامطلوب

$$p_i(n+1) = (1-b)p_i(n)$$

$$p_j(n+1) = (b/r - 1) + (1-b)p_j(n) \quad \forall j \neq i$$

در روابط فوق، a پارامتر پاداش و b پارامتر جریمه است.

۵- اتوماتای یادگیر سلولی

اتوماتای یادگیر سلولی، CLA، مدلی برای سیستم هایی است که از اجزاء ساده‌ای تشکیل شده اند و رفتار هر جزء بر اساس رفتار همسایگانش و نیز تجربیات گذشته اش تعیین و اصلاح می‌شود. هراتوماتای یادگیر سلولی، از یک اتوماتای سلولی تشکیل شده است که هر سلول آن به یک یا بیشتر اتوماتای یادگیر مجهز می‌باشد که وضعیت این سلول را مشخص می‌سازند [9]. همانند اتوماتای سلولی، قانون محلی در محیط حاکم است و این قانون تعیین می‌کند که آیا عمل انتخاب شده توسط یک اتوماتا در یک سلول بایستی پاداش داده شود و یا جریمه شود. برای اطلاعات بیشتر به [2-6] مراجعه نمایید.

۶- مدل CLA-EC

به منظور ساده کردن ارائه عملکرد CLA-EC فرض می‌کنیم فضای جستجوی مورد نظر یک فضای متناهی دودویی باشد. بنابراین یک مساله بهینه سازی می‌تواند به صورت زیر طرح گردد.

$$\max \{f(\underline{X}) | \underline{X} \in B^n\}$$

به طوریکه $f(\cdot)$ یک تابع حقیقی و $B^n = \{0, 1\}^n$ فضای جستجوی دودویی مورد نظر است. هر ژنوم در این الگوریتم دارای دو مولفه رشته ژنومی و مدل ژنومی می‌باشد. رشته ژنومی همان راه حل های میانی مساله مورد نظر می‌باشند. مدل ژنومی متشکل از تعدادی اتوماتای یادگیر می‌باشد که بر اساس تجارب گذشته خود و ژنومهای دیگر آموزش می‌بینند و بدین ترتیب فرایند تکامل به طریقی هدایت می‌شود تا ارزش رشته ژنومی بر اساس تابع ارزیابی بهبود یابد. حال برای ادامه بحث یک اتوماتای یادگیر سلولی، $(CLA(\underline{L}_1, \dots, \underline{L}_k))$ ، با k سلول که هر سلول مجهز به n اتوماتای یادگیر است در نظر بگیرد. حالت هر سلول توسط یک رشته ژنومی به طول n نشان داده میشود و نتیجتاً هر سلول میتواند 2^n حالت مختلف اختیار کند. با فرض همگام بودن اتوماتای یادگیر سلولی، در زمان t هر سلول، i ،

که $u(.)$ تابع پله ای است.

۷- نمایش قوانین

مقادیر هر رشته در سلول به صورت باینری می باشد و به سه قسمت تقسیم می شود. هر رشته شامل عناصر زیر می باشد:

صفات، عملگر بین صفات و مقادیر، مقادیر هر یک از صفات بر اساس نوع آنها در Data set. در قسمت مربوط به صفات در صورت ۱ بودن صفت مبنی بر حضور صفت در قانون می باشد و صفت موجود اغلب انواع مختلفی دارند و بصورت طبقه بندی شده ^{۱۰} یا پیوسته ^{۱۱} می باشد. بنابراین الگوریتم CLA-EC برای استخراج قوانین بایطد روشی را برای تبدیل این نوع صفت ها به یکدیگر فراهم کند. در آرایه عملگرها ابتدا باید نوع صفت مربوط به آن مورد توجه قرار گیرد. زمانی که صفت از نوع پیوسته باشد، اگر i امین صفت در Operator_array ۱ باشد از عملگر '≤' و گرنه از '≥' استفاده می کنیم.

برای صفات از نوع طبقه بندی شده، عددی یا اسمی ^{۱۲}، اگر i امین قسمت، از آرایه عملگرها ۱ باشد از '=' و گرنه از '≠' استفاده می کنیم. تبدل یا برگرداندن مقادیر صفات Attribute_array نسبتاً پیچیده است به دلیل اینکه نوع صفات نیز باید مورد توجه قرار گیرد.

برای انواع صحیح ^{۱۳}، تبدل ها به صورت زیر است:

$$V_{org}[i] = \text{ceil}(v_i * (V_i \text{ max} - V_i \text{ min}) + V_i \text{ min}) \quad (5)$$

تبدیل ها به صورت زیر است: برای انواع حقیقی ^{۱۴}

$$V_{org}[i] = v_i * (V_i \text{ max} - V_i \text{ min}) + V_i \text{ min} \quad (6)$$

تبدیل ها به صورت زیر است: برای انواع اسمی ^{۱۵}

$$V_{org}[i] = \text{ValArr}_i(\text{ceil}(v_i * \text{Count}_i)) \quad (7)$$

در اینجا $V_{org}[i]$ به معنی مقدار برگردانده شده برای i امین صفت رشته باینری می باشد. در تساوی (۵) و (۶)، مقدار ماکزیمم $V_i \text{ max}$ و مقدار مینیمم $V_i \text{ min}$ نوع صفت عددی می باشد. در رابطه (۷) نوع i امین صفت اسمی است. در این تساوی ValArr_i به معنی آرایه ای از مقادیر مختلف ذخیره شده از i امین صفت و Count_i به معنی تعداد مقادیر مختلف برای i امین صفت

رشته های ژنومی خود و همسایگان خود را مورد بررسی قرار داده و بر اساس تابع ارزیابی از میان آنها تعدادی را به عنوان ژنومهای مناسب انتخاب می کند ژنوم بر اساس ژنومهایی که انتخاب نموده است، یک بردار سیگنال تقویتی را ساخته و به اتوماتاهای یادگیر خود می دهد.

فرض کنید $X_i^t \in D_f$ وضعیت سلول i در گام t باشد. برای تولید وضعیت سلول i در گام $t+1$ ، X_{t+1}^i ، هر کدام از اتوماتاهای یادگیر مقدار بیت متناظر خود در ژنوم را اعلام می کنند. در صورتی که ژنوم جدید، new_{t+1}^i ، ارزش بیشتری نسبت به ژنوم قبلی داشته باشد، در سلول ژنوم جدید جایگزین ژنوم قبلی می شود، در غیر این صورت سلول رشته ژنومی قبلی را حفظ می کند.

$$X_{t+1}^i = \begin{cases} X_t^i & f(X_t^i) > f(\text{new}_{t+1}^i) \\ \text{new}_{t+1}^i & f(X_t^i) \leq f(\text{new}_{t+1}^i) \end{cases}$$

یک قانون برای CLA-EC از دو بخش تشکیل شده است. بخش اول استراتژی انتخاب و بخش دوم استراتژی تولید سیگنال تقویتی می باشد. منظور از استراتژی انتخاب سلول i ام نحوه انتخاب تعداد مشخص Se ، $1 \leq Se \leq m$ ، رشته ژنومی از میان m ژنوم همسایه، P^i ، می باشد. در استراتژی مورد استفاده، Se ژنوم از بهترین ژنومهای P^i را انتخاب می کنیم و آن را P_{se}^i می باشد. بردار سیگنال تقویتی به روشهای گوناگون قابل محاسبه است. در این بخش به شرح روش مورد استفاده در این مقاله می پردازیم.

فرض کنید $\vec{\beta}_t^i = (\beta_t^{i,1}, \dots, \beta_t^{i,n})$ بردار مورد نظر باشد که $\beta_t^i \in \{0,1\}^n$ ، به عنوان جریمه و ۰ به عنوان پاداش در نظر گرفته می شود. اگر $X_t^{i,1} = (X_t^{i,1}, \dots, X_t^{i,n})$ رشته ژنومی سلول i ام در زمان t باشد. برای هر متغیر $X_t^{i,j}$ ، $1 \leq j \leq n$ و به ازای مقادیر $N_{i,j}(k)$ ، $k = 0,1$ به صورت زیر محاسبه می شود.

$$N_{i,j}(k) = \sum_l \partial_l (X_t^{i,j} = k | P_{se}^i)$$

به طوریکه اگر متغیر $X_t^{i,j}$ در i امین سلول، مقدار k را داشته باشد $\partial_l (X_t^{i,j} = k | P_{se}^i) = 1$ و در غیر این صورت $\partial_l (X_t^{i,j} = k | P_{se}^i) = 0$ سپس با استفاده از $N_{i,j}(k)$ مقدار $\beta_t^{i,j}$ به صورت زیر بدست می آید:

$$\beta_t^{i,j} = u(N_{i,j}(1) - N_{i,j}(0)) \text{ If } X_t^{i,j} = 0$$

$$\beta_t^{i,j} = u(N_{i,j}(0) - N_{i,j}(1)) \text{ If } X_t^{i,j} = 1$$

¹⁰ Categorical

¹¹ Continues

¹² Nominal

¹³ Integer

¹⁴ Real

¹⁵ Nominal

است که می‌تواند داشته باشد و تابع $ceil(.)$ یک مقدار گرد شده به سمت بالا را برمی‌گرداند.

۸- ارزیابی قانون

هر قانون به صورت زیر تعریف می‌شود:

(۸) IF A THEN C

در اینجا A بخش مقدم قانون است. که اجتماعی از شرایط می‌باشد و C بخش تالی قانون است که یک کلاس از قبل تعیین شده (پیش‌بینی شده) است. صحت دسته بندی یک قانون به صورت تساوی زیر تعریف می‌شود:

$$(۹) \text{Accuracy} = TP / (TP + FP)$$

که برای اندازه گیری درجه اطمینان قانون مورد استفاده قرار می‌گیرد. زیر محاسبه می‌شود: مقدار شمول^{۱۶} هر قانون نیز از طریق تساوی

$$(۱۰) \text{Coverage} = TP / (TP + FN)$$

که از آن به نسبتی از نمونه‌ها که شرایط قانون ما را ارضا می‌کنند بر کل نمونه‌ها تعبیر می‌شود.

$TP = \text{Number of examples satisfying } A \text{ and } C.$

$FP = \text{Number of examples satisfying } A \text{ but not } C.$

$FN = \text{Number of examples not satisfying } A \text{ but satisfying } C.$

$TN = \text{Number of examples not satisfying } A \text{ not } C.$

کمترین مقدار شرط در قسمت مقدم یک قانون اغلب نشان دهنده قابل درک بودن آن می‌باشد. در روند استخراج قانون از تساوی زیر برای اطمینان پیدا کردن از کوتاهی قانون استفاده می‌کنیم.

$$(۱۱) \text{succinctness} = 1 - (\text{countAnt} - 1) / \text{attributeCount}$$

در اینجا countAnt به معنی تعداد شریط موجود در قسمت مقدم قانون و attributeCount به معنی تعداد صفات تصمیم‌گیری در مجموعه داده^{۱۷} می‌باشد. محاسبه تابع شایستگی با استفاده از سه معیار مذکور بصورت زیر انجام می‌شود:

$$(۱۲) \text{Fitness} = w_1 * (\text{Accuracy} * \text{Coverage}) + w_2 * \text{Succinctness} + w_3 * \text{Interesting}$$

در اینجا w_1 , w_2 , w_3 یک مقدار ثابت هستند که برای متعادل کردن وزن سه معیار در روند استخراج قانون مورد استفاده قرار می‌گیرند. از آنجایی که علاقه^{۱۸} در حد زیادی به کاربر وابسته است در این

مقاله مورد توجه قرار نمی‌دهیم. در آزمایش ما w_1 , w_2 , w_3 به ترتیب ۰.۸، ۰.۲ و ۰ در نظر گرفته می‌شوند.

۹- نتایج

برای آزمایشات از CLA-EC همگام با جمعیت ۲۰۰ و شعاع همسایگی ۱۰ انتخاب شده است. اتوماتای یادگیر، با نرخ یادگیری $a=b=0.1$ و $Se=10$ استفاده شده است. برای سهولت در ارائه از نماد $CLA-EC(automata(a,b),r,se,q)$ برای اشاره کردن به الگوریتم CLA-EC با q سلول، شعاع همسایگی r ، تعداد سلولهای انتخابی se ، اتوماتای یادگیر با پارامتر پاداش a و b جریمه است. همچنین آزمایشات بر روی دو دیتاست ZOO و CANCER انجام شد که مشخصات هر کدام یک از دیتاست‌ها در جدول ۱ آمده است.

جدول ۱: مشخصات دیتاست‌های مورد آزمایش

دیتاست		مشخصات
CANCER	ZOO	
۲	۷	تعداد کلاس
۵۶۹	۱۰۱	تعداد نمونه‌ها
۳۰	۱۶	تعداد صفات
پیوسته	طبقه بندی	نوع داده‌ها

نتایج آزمایشات بر روی مجموعه داده‌ی Zoo در جدول ۲ نشان داده شده است. همانطور که ملاحظه می‌گردد هفت کلاس در نظر گرفته شده است که الگوریتم پس از انجام تعداد تکرارهای متوالی توانسته است قوانین زیر را تولید نماید.

جدول ۲: نتیجه یادگیری الگوریتم در مجموعه داده Zoo

Class	Rule	A&C	A
1	If milk=1 then class=1	41	41
2	If feathers=1 toothed=0 then class=2	20	20
3	If hair=0 feathers=0 aquatic=0 backbone=1 legs!=8 then class=3	4	4
4	If milk=0 fins=1 then class=4	13	13
5	If milk=0 aquatic=1 breathes=1 legs!=2 catsize=0 then class=5	4	4
6	If aquatic=0 legs=6 then class=6	8	8
7	If airborne=0 backbone=0 then class=7	10	12

در جدول ۲ قوانین کلیه کلاسها با تعداد ۱۰۰ بار تکرار نشان داده شده است. همانطور که ملاحظه می‌نمایید بعنوان مثال کلاس ۱ متعلق به حیواناتی از مجموعه داده Zoo است که شیرده هستند. $Milk = 1$ به معنای آن است که حیوان شیر دارد. همچنین با تقسیم هر مجموعه داده به دو قسمت ۶۶٪ جهت آموزش مجموعه و ۳۴٪ جهت تست سیستم و اعمال الگوریتم CLA-EC باینری بر روی مجموعه داده‌ی آموزشی، قوانین استخراج شده و توسط مجموعه ۳۴٪ ی تست،

^{۱۶} Coverage

^{۱۷} Dataset

^{۱۸} Interesting

- in Complex Systems, Vol. 7, Nos. 3-4, pp.295-320, September/December 2004.
- [3] Khojasteh, M. R. and Meybodi, M. R. "Cooperation in Multi-Agent Systems Using Learning Automata", Iranian Journal of Electrical and Computer Engineering, Vol. 1, No. 2, pp.81-91, 2004.
- [4] Mars, P., Chen, J. R. and Nambiar, R. "Learning algorithms theory and applications in signal processing", control and communications, CRC Press, 1996.
- [5] Meybodi, M. R. and Kharazmi, M. R. "Application of Cellular Learning Automata to Image Processing", Journal of Amirkabir, Vol. 14, No. 56A, pp. 1101-1126, 2004.
- [6] Meybodi, M. R. and Khojasteh, M. R. "Application of Cellular Learning Automata in Modeling of Commerce Networks", in Proceedings of 6th Annual International Computer Society of Iran Computer Conference CSICC2001, Isfahan, Iran, pp. 284-295, 2001.
- [7] Lakshmivarahan, S. "Learning algorithms: Theory and Applications", New York: Springer-Verlag, 1981.
- [8] Narendra, K. S. and Thathachar, M. A. L. "Learning Automata: An introduction", Prentice Hall, 1989.
- [9] Beigy, H. and Meybodi, M. R., "Cellular Learning Automata With Multiple Learning Automata in Each Cell and Its Applications", IEEE Transaction on Systems, Man, and Cybernetics, 2009, to appear.

آزمایش گردیدند و نتایج بدست آمده از دیتاست CANCER در جدول ۳ برای ۵۰۰ و ۱۰۰۰ بار تکرار نشان داده شده است.

جدول ۳: نتایج بروی دیتاست CANCER در ۵۰۰ و ۱۰۰۰ بار تکرار با CLA-EC(automata(0.1,0.1),10,10,250)

Class	ITERATION	AVG-FIT	BEST-FIT
Malignant	500	0.830354	0.877026
	1000	0.845532	0.876667
Benign	500	0.904356	0.940562
	1000	0.914005	0.93914

۱۰- نتیجه گیری

در این مقاله تولید قوانین با استفاده از الگوریتم CLA-EC پیشنهادی تولید گردید. نتایج آزمایشات نشان می دهد این الگوریتم در تولید قوانین توانمند عمل نموده و طبقه بندی مناسبی ارائه می نماید. طبقه بندی انجام شده بر روی مجموع داده Zoo و CANCER دارای ویژگی کوتاه بودن قوانین می باشد که از مزایای آن است.

مراجع

- [1] J.Durkin, "Expert Systems: Design and Development", Mac Millan Publishing Company, 1994.
- [2] Beigy, H. and Meybodi, M. R. "A Mathematical Framework for Cellular Learning Automata", Advances