

معرفی ساختار جدیدی از درخت تصمیم و استفاده از آن به همراه کلاسترینگ (FCM)، برای تولید قوانین فازی

هادی صدوقی یزدی
h.sadoghi@um.ac.ir

محمد حسین یغمایی مقدم
yaghmaee@ieee.org

حسین خسروی رشخواری
hos.khosravi@stu-mail.um.ac.ir

دانشگاه فردوسی مشهد، دانشکده مهندسی، گروه کامپیوتر، مشهد، ایران

چکیده

بی نظمی (آنتروپی) می شود، گزینش می شود. مبنای فعالیت [۱ و ۲] بر همین اساس ایجاد شده است. تولید قوانین با استفاده از پیمایش درخت تصمیم در تحقیقات پیشین ارائه شده [۳] و در [۴] بهینه نیز می شوند. در [۴] قوانین از یک درخت C4.5 استخراج می شوند و نویسندگان بر بهینه بودن روش ذکر شده، تاکید کرده اند. در مقاله حاضر، از یک درخت تصمیم جدید استفاده می شود که بر اساس یک روش معروف کلاسترینگ - به جای مفهوم آنتروپی - یعنی FCM استفاده شده است. در روش ارائه شده، بهترین ویژگی به ازاء هر نود در حال ایجاد، بر این اساس گزینش می شود که تابع هدف FCM، مینیمایز شود. با گزینش این ویژگی، داده های یک خوشه جدا شده و در هر مرحله، (در هر نود) داده های باقیمانده، در یک فرایند بازگشتی درخت تصمیم را ایجاد می کنند. روش FCM، [۵ و ۶] همانطور که در پیش زمینه مقاله ارائه خواهد شد، یک روش کلاسترینگ است که با هدف مینیمایز کردن فاصله هر داده با مرکز خوشه ها - که بر اساس الگوریتم و به صورت دینامیک جابجا می شود - پیگیری می شود، و با توجه به اینکه مراکز هر خوشه در ابتدا به صورت تصادفی انتخاب می شوند همیشه یک پاسخ تکراری نخواهد داشت. در بخش دوم مقاله، پیش زمینه ای از درخت های تصمیم، خوشه بندی FCM و قوانین فازی ارائه شده و در بخش سوم، روش ابداع شده را مطرح خواهیم کرد. نهایتاً در بخش چهارم، نتایج شبیه سازی ارائه می شود.

۲- پیش زمینه

۲-۱- درخت های تصمیم گیری

درخت تصمیم، شیوه منحصر بفردی از ارائه یک سیستم است، که تصمیم گیری های آتی را تسهیل و سیستم را به نحو مناسبی تعریف می کند. با توجه به اینکه اکثر سیستم های مهندسی، اجرایی و محاسباتی را می توان در قالب یک سری داده (ویژگی یا ویژگی ها و خروجی منطبق با آن ها) تعریف کرد، می توان با استفاده از یک الگوریتم، (ایجاد درخت) ویژگی ها و خروجی ها را آنالیز کرد و سیستم را بر اساس این داده ها در قالب یک درخت تصمیم ارائه کرد.

قوانین فازی، مبتنی بر هر روشی که ایجاد شود، از اهمیت زیادی در پیاده سازی و بهینه سازی سیستم ها برخوردار است. استفاده از دانش بشری در تولید قوانین فازی، اگر چه مزیت خوانایی و نزدیکی به تخصص تجربی دارد، اما در همه سیستم ها قابل پیاده سازی نیست. استفاده از خروجی های یک سیستم که شامل یک سری داده است، از آنجا که بر پایه کارکرد صحیح سیستم ایجاد شده، با درصد زیادی قابل اعتماد است. در مقاله حاضر، با استفاده از خروجی های سیستم و به واسطه تولید یک درخت تصمیم گیری، قوانین فازی استخراج می شوند. در حقیقت پیمایش درخت تصمیم گیری منجر به تولید قوانین شده و این قوانین، فازی و در نهایت بهینه می - شوند. درخت تصمیم ارائه شده، در مقایسه با پیاده سازی های پیشین، جدید بوده و به عنوان یک راه حل جدید در کلاسه بندی^۱ نیز قابل ارائه است. از جمله مزیت های درخت تصمیم جدید نسبت به C4.5 (که از پرکاربردترین ساختار های تصمیم گیری است)، می توان به قابلیت تصمیم گیری همزمان بر روی بیش از یک ویژگی اشاره کرد، که در C4.5 فراهم نیست. عدم تولید یک جواب مشخص، و بهبود جواب در اثر تکرار الگوریتم نیز، از جمله موفقیت های ساختار ارائه شده است.

کلمات کلیدی

سیستم های فازی، درخت تصمیم، خوشه بندی، FCM، C4.5.

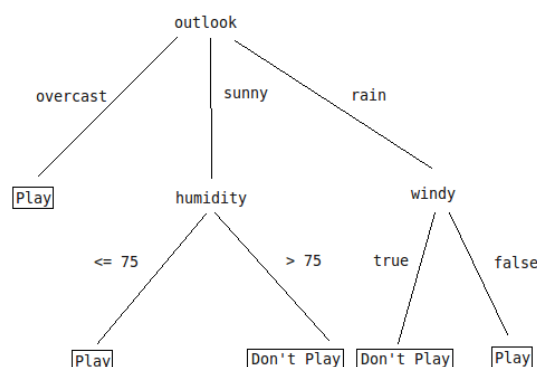
۱- مقدمه

استفاده از درخت های تصمیم به عنوان کلاسیفایر در حال حاضر کاملاً کاربردی شده و انواع مختلف پیاده سازی آن من - جمله ID3، C4.5، C5 عملاً در کاربرد های متنوع استفاده می شود. درخت تصمیم بر اساس آنالیز داده های ورودی و به منظور پیدا کردن یک ویژگی^۲ به عنوان مبنای تصمیم گیری برای هر نود بکار می رود. در هر نود ویژگی های مختلف داده ها بررسی شده و یک ویژگی که در صورت انتخاب باعث کاهش

برای مثال داده های زیر (که شامل یک سری ویژگی و خروجی متناظر است) را توسط الگوریتم ارائه شده در [۷]، به صورت درخت ارائه شده در شکل ۱ نمایش داده ایم.

```
sunny, 85, 85, false, Don't Play
sunny, 80, 90, true, Don't Play
overcast, 83, 78, false, Play
rain, 70, 96, false, Play
rain, 68, 80, false, Play
rain, 65, 70, true, Don't Play
overcast, 64, 65, true, Play
sunny, 72, 95, false, Don't Play
sunny, 69, 70, false, Play
rain, 75, 80, false, Play
sunny, 75, 70, true, Play
overcast, 72, 90, true, Play
overcast, 81, 75, false, Play
rain, 71, 80, true, Don't Play
```

جدول ۱- داده های ورودی (داده ها در دو کلاس طبقه بندی شده اند: Play و Don't Play) - ۴ فیلد اول، ویژگی ها و فیلد آخر، کلاس طبقه بندی شده است.



شکل ۱- درخت متناسب با داده های جدول ۱

روش استفاده شده در ایجاد درخت از متغیرهای معرفی شده در جدول ۲ استفاده می کند. می بینیم که روش استفاده شده، با توجه به اینکه متغیر temperature را، در تصمیم گیری بی اثر تشخیص داده است، به درستی آن را حذف کرده است.

```
outlook: sunny, overcast, rain.
temperature: continuous.
humidity: continuous.
windy: true, false.
```

جدول ۲- نوع متغیرهای شکل ۱

بعد از ایجاد درخت، می توان از ساختار ایجاد شده، برای کلاس بندی داده های تست استفاده کرد. در این صورت توسط این ساختار سریع، داده های تست به خوبی برای جدا سازی در کلاس های مختلف تصمیم گیری می شوند.

عموما داده های تعلیم که برای ایجاد درخت استفاده می شوند، از داده های تست که برای ارزیابی درخت ایجاد شده استفاده می شوند، متفاوت بوده و تعداد خطا در تشخیص کلاس داده های تست، معیار مناسب بودن الگوریتم می باشد.

۱-۲-۳: ID3، C4.5، C5

الگوریتم ID3 یک الگوریتم برای ساختن درخت تصمیم می باشد. [۱] در این الگوریتم از مفهوم بی نظمی^۳ برای دسته بندی داده ها استفاده شده است و الگوریتم در صدد آن است که میزان بی نظمی در نودهای بالای درخت حداقل باشد تا بتوان درختی با حداقل ارتفاع داشت. پس ابتدا برای تمامی ویژگی های داده های اولیه، بی نظمی را با فرمول ۱ محاسبه کرده و سپس آن ویژگی که بیشترین سودمندی را خواهد داشت، به عنوان ریشه انتخاب می کند.

$$I = - \sum_c p(c) \log_2 p(c) \quad (1)$$

برای محاسبه سودمندی هر یک از ویژگی ها از فرمول ۲ استفاده می شود.

$$Gain(A) = I - I_{res}(A) \quad (2)$$

که در آن I_{res} میزان بی نظمی باقیمانده در دسته ها به واسطه استفاده از یک ویژگی می باشد، که با کمک مجموع احتمال وقوع هر یک از تقسیمات، قابل حصول می باشد. (فرمول ۳)

$$I_{res} = - \sum_v p(v) \sum_c p(c|v) \log_2 p(c|v) \quad (3)$$

این الگوریتم فقط قادر به دسته بندی داده ها با دامنه ویژگی های گسسته و محدود می باشد و در مورد داده های نوبی و مخدوش کارایی ندارد.

الگوریتم C4.5 تکمیل شده الگوریتم ID3 می باشد. این الگوریتم قادر به دسته بندی داده های پیوسته و نوبی نیز می باشد. برای این منظور ابتدا داده ها را مرتب کرده سپس مقادیر سودمندی را برای تمامی حالت هایی که امکان جدا شدن این داده های مرتب شده از هم وجود دارد را بدست آورده و جداساز متناظر با بزرگترین مقدار سودمندی را به عنوان جدا کننده انتخاب می کنیم.

آخرین نسخه از درخت تصمیم C4.5 توسط [۸] تجاری شده و C5 نامیده شده است.

نسخه مذکور قابلیت آنالیز چند میلیون رکورد را داراست و از قابلیت های پردازشی CPU های چند هسته ای، به منظور بهبود کارکرد الگوریتم، بهره می گیرد. نتایج بهره برداری شده از نسخه آزمایشی ارائه شده در سایت اینترنتی [۸] کسب شده اند.

^۳ entropy

۲-۲- روش کلاستریگ FCM^۴

در مسائل شناسایی الگو منظور از خوشه، مجموعه ای از داده ها است که به علت شباهت زیادی که به هم دارند در یک دسته قرار گرفته اند. در خوشه بندی سعی می شود جمعی از داده ها به صورت بدون ناظر به خوشه هایی تقسیم شوند که شباهت داده های درون هر خوشه حداکثر و شباهت بین داده های درون خوشه های مختلف حداقل گردد [2]. یکی از رایجترین روشهای خوشه بندی روش خوشه بندی FCM است. روش خوشه بندی FCM مانند روش **K-Means** جزء خانواده الگوریتم های خوشه بندی دارای **تابع هدف** قرار دارد که در آنها هدف حداقل کردن یک تابع می باشد. [۱۵ و ۱۷ و ۱۸]

در این الگوریتم نمونه ها به C خوشه که از قبل تعداد مشخصی دارد تقسیم می شوند. در الگوریتم خوشه بندی C میانگین فازی، تابع هدف بصورت فرمول ۴ می باشد:

$$J = \sum_{i=1}^c \sum_{k=1}^n u_{ik}^m d_{ik}^2 = \sum_{i=1}^c \sum_{k=1}^n u_{ik}^m \|x_k - v_i\|^2 \quad (۴)$$

در فرمول ۴، m یک عدد حقیقی بزرگتر از ۱ است که در اکثر موارد ۲ انتخاب می شود. اگر در فرمول فوق m را برابر ۱ قرار دهیم تابع هدف خوشه بندی C میانگین (کلاسیک) غیر فازی بدست می آید. در فرمول فوق x_k نمونه K ام و v_i نماینده یا مرکز خوشه A ام و n تعداد نمونه ها می باشد. u_{ik} میزان تعلق نمونه A ام در خوشه K ام را نشان می دهد. علامت $\|*\|$ میزان تشابه (فاصله) نمونه با (از) مرکز خوشه می باشد که می توان از هر تابعی که بیانگر تشابه نمونه و مرکز خوشه باشد استفاده کرد. از روی u_{ik} می توان یک ماتریس U تعریف کرد که دارای C سطر و n ستون می باشد و مولفه های آن هر مقداری بین ۰ تا ۱ را می توانند اختیار کنند. اگر تمامی مولفه های ماتریس U بصورت ۰ و ۱ باشند الگوریتم مشابه C میانگین کلاسیک^۵ خواهد بود. با اینکه مولفه های ماتریس U می توانند هر مقداری بین ۰ تا ۱ را اختیار کنند اما مجموع مولفه های هر یک از ستونها باید برابر ۱ باشد و داریم:

$$\sum_{i=1}^c u_{ik} = 1, \forall k = 1, \dots, n \quad (۵)$$

معنای این شرط این است که مجموع تعلق هر نمونه به C خوشه باید برابر ۱ باشد. برای بدست آوردن فرمولهای مربوط به u_{ik} و v_i باید تابع هدف تعریف شده را می نیمم کنیم. با

استفاده از شرط فوق و برابر صفر قرار دادن مشتق تابع هدف خواهیم داشت:

$$u_{ik} = \frac{1}{\sum_{j=1}^c \left(\frac{d_{ik}}{d_{jk}}\right)^{2/(m-1)}} \quad (۶)$$

$$v_i = \frac{\sum_{k=1}^n u_{ik}^m x_k}{\sum_{k=1}^n u_{ik}^m} \quad (۷)$$

با استفاده فرمول های ۶ و ۷، الگوریتم خوشه بندی C میانگین فازی بصورت زیر می باشد.

مراحل الگوریتم:

۱. مقدار دهی اولیه برای c، m و U^0 . خوشه های اولیه حدس زده شوند.

۲. مراکز خوشه ها محاسبه شوند (محاسبه v_i ها).

۳. محاسبه ماتریس تعلق از روی خوشه های محاسبه

شده در ۲.

۴. اگر $\|U^{i+1} - U^i\| \leq \epsilon$ اگر ۱

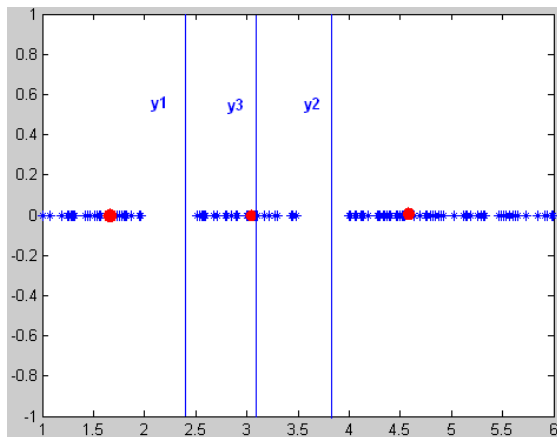
غیر اینصورت برو به مرحله ۲.

نقاط قوت الگوریتم C میانگین فازی عبارتند از اینکه همیشه همگرا می شود. و اینکه الگوریتم بدون نظارت^۶ است. و از نقاط ضعف الگوریتم C میانگین فازی می توان به زیاد بودن زمان محاسبات و حساسیت به حدس های اولیه و نویز اشاره کرد و ممکن است الگوریتم در مینیمم های محلی متوقف شود. حساسیت الگوریتم به نویز، با در نظر گرفتن یک خوشه برای داده های نویزی حل شده است. ضمناً در صورت وجود تعدادی داده دارای برچسب، حدس های اولیه برای مراکز خوشه نیز بهبود می یابند. [۱۹]

۲-۳- پایگاه قوانین فازی

منطق فازی برای اولین بار توسط پروفیسور زاده [۹] ارائه شد و تا کنون در علوم و صنایع مختلف کاربردی شده است.

اولین و موفق ترین کاربرد منطق فازی در زمینه کنترل است و مهم ترین جنبه عملی این تئوری، پیاده سازی سیستم های پیچیده در قالب متغیرها و قوانین فازی است. متغیرهایی که به صورت فازی (پیوسته و معمولاً به صورت بازه ای بین صفر و یک) تعریف می شوند و قوانینی که با دانش بشری مطابق بوده و به کاربرد های عینی نزدیک ترند.



شکل ۲- داده های خوشه بندی شده، مراکز خوشه و خطوط جداساز

علاوه بر روش کشف y_1 ، y_2 ، y_3 که بر مبنای خوشه بندی فازی C میانگین^۸ ارائه شده، تعریف معیاری مناسب، برای ترجیح یک ویژگی به منظور انتخاب، برای نود در حال ایجاد، بسیار مهم است. دو مساله مطرح شده یعنی روش کشف آستانه ها و روش انتخاب بهترین ویژگی، در ۱-۳ و ۲-۳ ارائه می شود. لازم به ذکر است با اینکه در مثال مورد بررسی، در هر نود، یک ویژگی مبنای تصمیم گیری است، تصمیم گیری بر مبنای همزمان ۲ یا چند ویژگی نیز میسر است و عدم محدودیت در تعداد ویژگی، خود یکی از نقاط قوت الگوریتم ارائه شده است.

۳-۱- انتخاب آستانه جداساز

همانطور که در پیش زمینه ارائه شد، FCM صرفاً یک روش خوشه بندی است. در FCM مراکز خوشه ها به طور تصادفی انتخاب شده، و پس از تکرار بهبود می یابند. فرض کنید، ویژگی اول (از ۴ ویژگی) را به عنوان اولین ویژگی ای که می بایست آنالیز شود، انتخاب کرده ایم (با توجه به بخش ۲-۳ این فرایند را برای ویژگی های دیگر نیز اجرا خواهیم کرد). FCM را بر روی اعداد تک بعدی مربوط به ویژگی اول اجرا کرده و پیشاپیش تعداد کلاس ها (۳) را به عنوان ورودی به FCM اعلام می کنیم. FCM پس از اجرا، سه مرکز خوشه معرفی می کند. خط جداساز کلاس ۱ از ۲ (y_1)، نقطه میانگین مراکز خوشه های مربوطه است. y_2 و y_3 نیز به همین ترتیب انتخاب می شوند.

۳-۲- انتخاب بهترین ویژگی برای جداسازی در هر نود

پس از انتخاب نقاط جداساز برای ویژگی اول، این فرایند را برای ویژگی های دیگر نیز اجرا می کنیم. مبنای انتخاب ویژگی برتر برای قرار گرفتن در یک نود، کمینه کردن تابع هدف^۹ FCM می باشد.

پایگاه قوانین فازی^۷، در بر گیرنده قوانینی است که به صورت IF-THEN بر متغیر های فازی اعمال می شود. نحوه تعریف and و or بر اساس تشخیص متخصص تعریف شده و aggregation، implication نیز می بایست به طور صحیح انتخاب شوند. در این مقاله، هدف ارائه روشی برای تولید قوانین بوده و در بخش ۳ به تفصیل مورد بحث قرار می گیرد. با توجه به گستردگی بحث تئوری فازی، پیشنهاد می شود به [۱۰-۱۳] مراجعه کنید.

۳- الگوریتم پیشنهادی

به صورت اجمالی، الگوریتم پیشنهادی را می توان به سه بخش تقسیم کرد:

الف- تولید درخت تصمیم بر مبنای داده های ورودی و با کمک FCM

ب- استخراج قوانین از درخت (پیمایش درخت تصمیم)

ج- ایجاد سیستم فازی و استفاده از قوانین ایجاد شده در بخش قبل

د- بهینه سازی قوانین سیستم فازی

بخش الف، که آن را می توان نقطه قوت الگوریتم ارائه شده نامید، با آنالیز داده های ورودی در هر نود و به واسطه بررسی تک تک ویژگی ها، درخت تصمیم را ایجاد می کند.

اصولاً استفاده از درخت تصمیم به منظور تولید قوانین پیش از این نیز در [۳ و ۴] ارائه شده بود، اما ساختار متفاوت درخت ارائه شده، و استفاده از FCM در جداسازی داده ها، قابلیت جدید است که در بهبود نتایج موثر می باشد.

برای شفاف شدن روش ارائه شده، داده هایی را به عنوان مثال در نظر می گیریم که شامل چهار ویژگی بوده و در سه کلاس طبقه بندی شده اند.

انتخاب یک ویژگی در هر نود، به این معنی است که این ویژگی، در صورت انتخاب، بیش از ویژگی های دیگر داده ها را جدا نموده و بنابراین در درخت نهایی، کاربر را سریع تر به تصمیم نهایی می رساند. برای مثال درخت C4.5 با استفاده از مفهوم آنتروپی، داده ها را به نحوی جدا می کند که بهترین ویژگی برای حداقل کردن بی نظمی انتخاب شده و در ضمن آستانه ای برای تقسیم داده ها و تصمیم گیری آن ها انتخاب می شود. در مثال ما، با توجه به ۳ کلاس بودن داده ها، مقدار آستانه، یک عدد نیست، بلکه ۳ عدد می باشد. y_1 جداساز داده های کلاس ۱ از ۲، y_2 جداساز داده های کلاس ۲ از ۳ و y_3 جداساز داده های کلاس ۱ از ۳، می باشند. (شکل ۲)

^۸ Fuzzy C-Means (FCM) objective function

^۷ Fuzzy Rule Base

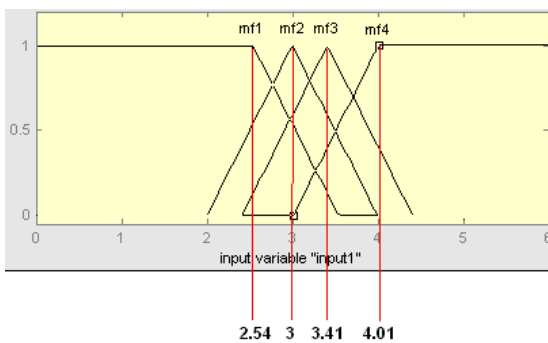
۳-۳- استخراج قوانین و فازی سازی

درخت ایجاد شده، پس از یک پیمایش به یک سری قوانین با شرطی مبتنی بر ویژگی ها، تبدیل شده، و به منظور فازی سازی قابل استفاده اند. برای مثال، قسمتی از قوانین استخراج شده از درخت شکل ۳، با پیمایش اول عمق عبارتند از:

```
if f2<1 and f3<5.1 and f4<2.3 then Class=A
if f2<1 and f3<5.1 and 2.3<f4<4.1 then Class=C
if f2<1 and f3<5.1 and f4>4.1 then Class=B
.....
```

فرایند فازی سازی را می توان به صورت زیر خلاصه کرد:

- به ازاء تک تک ویژگی ها، متغیر فازی تعریف می کنیم.
 - به ازاء هر متغیر، تمامی نود ها را مورد بررسی قرار داده تا، در صورت وجود یک نقطه در شرط، آن را در تابع عضویت وارد کنیم. (می توان از انواع مثلثی یا ذوزنقه ای یا موارد مشابه در ایجاد تابع عضویت بهره جست.)
 - به ازاء تعداد کلاس ها، متغیر خروجی ایجاد می-کنیم.
 - قوانین استخراج شده از درخت را در پایگاه قوانین فازی وارد می کنیم.
- نهایتاً، داده های تست را بر روی سیستم فازی وارد کرده و نتایج را آنالیز می کنیم.
- به عنوان مثال، تابع عضویت متغیر ورودی اول (ویژگی ۱) را می توان به فرم شکل ۴ نمایش داد:



شکل ۴- تابع عضویت متغیر F1

(مرکز توابع عضویت مثلثی، از پیمایش تمامی شاخه های درخت و جستجوی تمامی شروط اعمال شده بر آن ایجاد شده اند.)

به عبارت دیگر، بر اساس تابع هدف (فرمول ۴)، ویژگی ای به عنوان بهترین جداساز انتخاب می شود که، تابع هدف را کمینه کند. پس از انتخاب بهترین ویژگی، الگوریتم در مرحله بعدی، کلاس داده ها را مجدداً در ناحیه های تصمیم مورد بررسی قرار می دهد؛ در صورتیکه تمام داده های یک ناحیه تصمیم، در یک کلاس قرار داشته باشند، ما به یک تصمیم قطعی (به نود نهایی) برای ناحیه تصمیم مورد نظر دست یافته ایم. در این صورت، این داده ها (که به تصمیم نهایی منجر شده اند) را از داده های اولیه خارج کرده و مجدداً فرایند بازگشتی را برای نود های دیگر و داده های باقیمانده، اجرا می کنیم.

در غیر این صورت (عدم دست یافتن به تصمیم نهایی)، داده های هر خوشه را مجدداً در فرایند بازگشتی مربوطه اجرا کرده و فرایند بازگشتی را تا آنجا ادامه می دهیم که، تمامی نود ها به گره نهایی ختم شوند.

به صورت کلی الگوریتم ارائه شده را می توان به صورت زیر خلاصه کرد:

۱- شروع

۲- به ازاء هر ویژگی (X_i) :

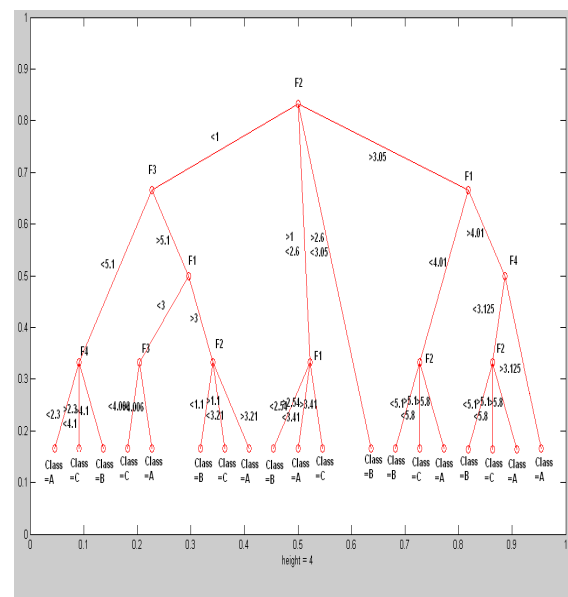
۳- خوشه ها = اجرای FCM روی دادهای ورودی.

۴- برای ویژگی منتخب (X_i) با کمترین *objective function*.

۵- در صورتیکه خوشه ها دارای داده هایی از یک کلاس بودند، یک نود نهایی ایجاد کن و داده ها را از چرخه بازگشتی حذف کن.

۶- در غیر این صورت، برای داده های هر خوشه، فرایند بازگشتی ۲ تا ۶ را اجرا کن.

درخت ایجاد شده توسط الگوریتم فوق در شکل ۳ نمایش داده شده است:



شکل ۳- درخت ایجاد شده پس از اجرای الگوریتم

های روش پیشنهادی است. چرا که می توان الگوریتم را تا آنجایی مکررا اجرا کرد که درخت مناسب با حداقل خطا روی داده های تست ایجاد شود. برای مثال، جدول ۵ نتایج اجرای الگوریتم، در ۱۵ تکرار، را نمایش می دهد).

تکرار	عمق درخت	خطا روی داده های تعلیم	خطا در تست
۱	۷	۳	۶
۲	۸	۱	۶
۳	۷	۲	۷
۴	۹	۱	۷
۵	۹	۱	۶
۶	۶	۳	۶
۷	۸	۰	۴
۸	۷	۲	۴
۹	۶	۴	۱۰
۱۰	۸	۳	۱۰
۱۱	۷	۲	۷
۱۲	۷	۲	۴
۱۳	۸	۱	۶
۱۴	۸	۰	۶
۱۵	۸	۲	۷

جدول ۵- نتایج اجرای FCMDT در ۱۵ تکرار

۵- نتیجه گیری

در این مقاله روشی جدید برای تولید قوانین فازی ارائه شده است. قوانین از پیمایش یک درخت تصمیم استخراج شده اند. درختی که توسط یک الگوریتم جدید و کارا، از الگوریتم FCM برای جداسازی داده ها استفاده کرده و با کارایی بالا، قوانین را تولید می کند. قوانین استخراج شده، به همراه متغیر های مرتبط (که همان ویژگی های داده ها هستند) فازی شده و خروجی های سیستم تحت شرایط مختلف مورد بررسی قرار می گیرد.

با توجه به کارایی مناسب درخت تصمیم جدید (FCMDT)، قوانین ایجاد شده در سیستم فازی کارا ارزیابی شده و راهکار جدیدی برای ادامه تحقیقات در این زمینه پیش بینی می شود.

۴-۳- بهینه سازی قوانین فازی ایجاد شده

به منظور بهینه سازی قوانین، همانطور که در [۴] ارائه شده، استفاده از الگوریتم ژنتیک پیشنهاد می شود. اگر چه که بررسی این مساله جزء اهداف ارائه کنندگان این مقاله نیست.

۴- نتایج شبیه سازی

به منظور شبیه سازی الگوریتم ارائه شده، از نرم افزار MATLAB استفاده شده است.

داده های تعلیم و تست از بانک اطلاعاتی معتبری [۱۴] استخراج شده و شبیه سازی، به اثبات ادعای مطرح کمک کرده است.

۶۴ داده (به مشخصات جدول ۳) را بر روی سه درخت تصمیم مطرح، یعنی C5، C4.5 و درخت ارائه شده (که نام آن را FCMDT نامیده ایم) تعلیم داده ایم. و از ۱۵۰ داده برای تست استفاده شده است.

نتایج ارزیابی در جدول ۴ ارائه شده اند.

تعداد ویژگی ها	تعداد کلاسها	تعداد نقاط تعلیم	تعداد نقاط تست
۴	۳	۶۴	۱۵۰

جدول ۳- مجموعه داده مورد استفاده در آزمایش

نوع درخت	عمق درخت	خطا روی داده های تعلیم	خطا در تست
C5	۵	۲	۴
C4.5	۴	۲	۵
FCMDT	۸	۰	۴

جدول ۴- مقایسه درخت های C5، C4.5 و FCMDT

با توجه به نتایج، بهبود پاسخ سیستم نسبت به درخت های پیشین (خصوصا C4.5) قابل توجه است، علاوه بر این، عدم پاسخ دهی یکسان الگوریتم FCMDT در اجرای مکرر آن، از نقاط قوت راه حل پیشنهادی است. با توجه به اینکه روش FCM همواره مراکز خوشه یکسانی نتیجه نمی دهد، درخت ایجاد شده نیز در تکرار متوالی الگوریتم متفاوت خواهد بود. و البته با توجه به اینکه از اجرای الگوریتم ایجاد درخت، صرفا انتظار تولید یک درخت مناسب می رود، این مساله جزء مزیت

- [1] Quinlan, J. R. Induction of Decision Trees. *Mach. Learn.* 1, 1, 81-106, 1986.
- [2] J. R. Quinlan. Improved use of continuous attributes in c4.5. *Journal of Artificial Intelligence Research*, 4:77-90, 1996.
- [3] Guillaume, S., Designing fuzzy inference systems from data: an interpretability-oriented review. *IEEE Trans. Fuzzy Systems*. v9 i3. 426-443.
- [4] Pietari Pulkkinen , Hannu Koivisto, Fuzzy classifier identification using decision tree and multiobjective evolutionary algorithms, *International Journal of Approximate Reasoning*, v.48 n.2, p.526-543, June, 2008
- [5] J. C. Bezdek: *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, 1981.
- [6] F. H'oppner, F. Klawonn, R. Kruse, T. Runkler, *Fuzzy Cluster Analysis, (Methods for Classification, Data Analysis and Image Recognition)*, John Wiley & Sons, Ltd., 1999.
- [7] Ross Qinlan Personal Academic Page. <http://www.rulequest.com/Personal>.
- [8] RuleQuest Research Co. <http://www.rulequest.com>
- [9] Zadeh, L. *Fuzzy sets. Information and Control*, 8(3):338-353. 1965.
- [10] M. Sugeno, "An introductory survey of fuzzy control," *Inf. Sci.*, vol. 36, no. 1/2, pp. 59-83, 1985.
- [11] C. C. Lee, "Fuzzy logic in control systems: Fuzzy logic controller," *IEEE Trans. Syst., Man, Cybern.*, vol. 20, pp. 404-435, 1990.
- [12] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control." *IEEE Trans. Syst., Man, Cybern.*, vol. 15, pp. 116-132, Jan./Feb. 1985.
- [13] K. M. Passino and S. Yurkovich, *Fuzzy Control Menlo Park, CA: Addison Wesley Longman*, 1998.
- [14] UCI machine Learning Repository web site: <http://archive.ics.uci.edu/ml/datasets.html>
- [15] J. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, 1981.
- [16] R. O. Duda, P. E. Hart, G. D. Stork, *Pattern Classification*. wiley-Interscience, ISBN-0-47b-0sbb9-3.
- [17] R. Krishnapuram, J. M. Keller, A Possibilistic approach to clustering, *IEEE Transactions on fuzzy systems*, Vol. 1, No. 2, May 1993.
- [18] حاج احمدی، همایون پور، احدی- بهبود خوشه بندی C میانگینه فازی، امکانی برای خوشه بندی مجموعه دادگان بزرگ و نویزی-ولین کنفرانس فازی و سیستم های هوشمند، مشهد، ۲۰۰۷.
- [19] Dr.Saeed Shiry Ghidary Academic website, <http://ceit.aut.ac.ir/~shiry>