



گزارش تکلیف پیاده‌سازی آزمایش اول مقاله

Distributed representation of fuzzy rules and its
application to pattern classification

داریوش حسن‌پور آده

۹۳۰۸۱۶۴

۱ چکیده

در این مقاله [۱] سعی بر این بوده است که مفهوم نمایش توزیع یافته قوانین فازی را ارائه دهد و در ادامه کاربردی از این مفهوم را در مسایل طبقه بندی^۱ ارائه داده است. هدف از انجام این تکلیف آشنایی با مفهوم قوانین فازی در عمل و انجام آزمایش شماره ۱ مقاله و مقایسه نتایج به دست آمده با نتایج ارائه شده در مقاله می باشد. در این مقاله ابتدا مروری بر مفاهیم اولیه مطرح شده توسط مقاله می پردازیم سپس درمورد ساختار کدهای نوشته شده صحبتی می کنیم و سپس نتایج بدست آمده از آزمایش اول مقاله را ارائه می دهیم و در نهایت به نتیجه گیری و جمع بندی می رسیم.

فهرست مطالب

۱	چکیده	۱
۲	مقدمه	۲
۳	قوانین فازی و تقسیم بندی فضای الگوها	۲
۴	طبقه بندی الگوها با استفاده از قوانین فازی	۳
۱.۴	توابع عضویت	۳
۱.۱.۴	تابع عضویت مثلثی	۳
۲.۱.۴	تابع عضویت ذوزنقه	۴
۲.۴	استخراج قوانین	۴
۳.۴	استنتاج و طبقه بندی	۵
۵	پیاده سازی مقاله	۷
۱.۵	استخراج قوانین	۷
۱.۱.۵	تابع استخراج قوانین	۷
۲.۵	استنتاج از قوانین	۸
۱.۲.۵	تابع استنتاج از قوانین و طبقه بند	۸
۳.۵	توابع ابزاری	۸
۱.۳.۵	تابع راه انداز	۹
۶	آزمایش اول	۹
۷	جمع بندی و نتیجه گیری	۱۱
۸	مراجع	۱۱

۲ مقدمه

برای بدست آوردن قوانین فازی برای دسته‌بندی الگوهای عددی شامل دو قسمت می‌شود:

۱. تقسیم فضای الگوهای آموزشی

۲. ارائه قوانینی برای هرکدام از این تقسیم‌بندی‌ها

از مسایل مطرح برای استخراج قوانین فازی برای طبقه‌بندی کردن نمونه‌ها تعیین این مهم که سائز هریک از این تقسیم‌بندی‌ها چقدر باشد تا طبقه‌بند کننده با مشکل Overfitting یا Underfitting مواجه نشود؛ می‌باشد. زیرا که اگر سائز تقسیم‌بندی فضای الگوها درشت باشد قوانین تولیدی برای هر ناحیه از دقت پایینی و در نتیجه طبقه بند بدست آمده از قدرت کمتری برخوردار خواهد بود. و از سوی دیگر اگر سائز تقسیم‌بندی فضای الگوها کم باشد به علت زیاد شدن زیر-فضاها^۲ به قوانین استخراج شده افزوده می‌شود و پیچیدگی طبقه‌بند افزایش می‌یابد و همانند اکثر طبقه‌بندها زمانی که طبقه‌بند پیچیده‌تر شود مشکل Overfitting پیش می‌آید و همچنین در زمانی که تعداد زیر-فضاها زیاد است ممکن است که زیر-فضایی نمونه‌ای دربر نداشته باشد باعث می‌شود برای برخی از نواحی قوانین تولید نشود.

برای حل مشکلات مذکور مقاله دوراه‌حل نام برده است یکی که مبتنی بر میزان چگالی داده‌ها می‌باشد و ایده به این صورت است که در نواحی‌ای که دارای چگالی داده‌ای زیاد (که معیار می‌تواند تعداد داده‌ها یا آنتروپی داده‌ای باشد) هستند را به نواحی بیشتری شکسته شوند و نواحی با چگالی کم به تعداد زیر-فضاهای کمتری تقسیم شوند. راه‌حل دیگر که راه‌حل ارائه شده توسط مقاله می‌باشد این است که نواحی به صورت تقسیم‌بندی فضای الگوها از تقسیم‌بندی درشت به ریز می‌تواند نتایج مطلوبی ارائه دهد زیرا که اگر در تقسیم‌بندی‌های ریز اگر برای نواحی‌ای قوانینی استخراج نشود حتما در تقسیم‌بندی درشت قوانین برای آن ناحیه بدست خواهد آمد. در ادامه به توضیح مختصری از مقاله و سپس به جزییات پیاده‌سازی انجام شده می‌پردازیم.

۳ قوانین فازی و تقسیم‌بندی فضای الگوها

در این مقاله فضای الگوها را $[0, 1] \times [0, 1]$ در نظر گرفته است و این فضا را به L^2 زیر-فضا تقسیم کرده است و به و هر بعد را به پارتیشن‌های $\{A_1^L, A_2^L, \dots, A_L^L\}$ تقسیم کرده است. حال طبق ایده‌ی این مقاله که تقسیم‌بندی فضای الگوهای از پارتیشن‌های درشت به پارتیشن‌های ریز بود مقدار L با مقدار K که مقداری متغیر می‌باشد و به بازه‌ی $[2, L]$ محدود می‌باشد جایگزین می‌شود و در نتیجه ما قوانینی در به ازای پارتیشن‌های $\{A_1^L, A_2^L, \dots, A_L^L\}, \{A_1^3, A_2^3, A_3^3\}, \dots, \{A_1^2, A_2^2\}$ بدست می‌آوریم که فرمت این قوانین به صورت ۱ می‌باشد.

$$\text{If } x_1 \text{ is } A_i^K \text{ and } x_2 \text{ is } A_j^K \text{ then } x_p \text{ belongs to } G_{ij}^K \text{ with CF} = CF_{ij}, \quad i, j = 2, 3, \dots, K; \quad K = 2, 3, \dots, L \quad (1)$$

Sub-Space^۲

۴ طبقه‌بندی الگوها با استفاده از قوانین فازی

در این قسمت به شرح مختصری از توابع عضویت مورد استفاده در مقاله و روند استخراج قوانین و استنتاج براساس آنها برای نواحی پارتیشن شده می‌پردازیم.

۱.۴ توابع عضویت

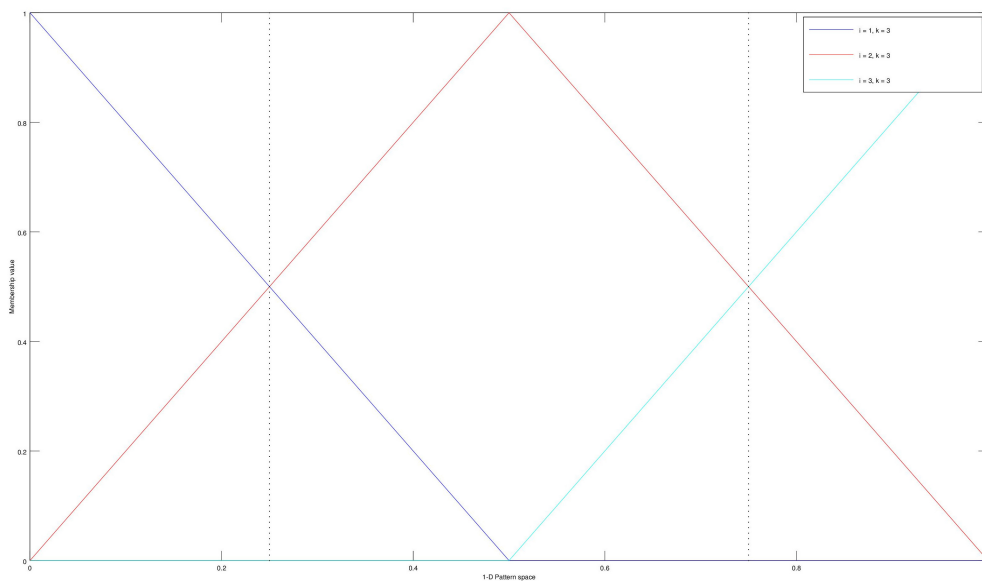
در این مقاله از ۲ نوع تابع عضویت مثلثی و ذوزنقه برای طبقه‌بندی الگوها استفاده شده است که در زیر به شرح و توضیح منطق عضویت هریک از آنها می‌پردازیم.

۱.۱.۴ تابع عضویت مثلثی

رابطه تابع عضویت مثلثی مورد استفاده در ۲ آمده است.

$$\mu_i^K(x) = \max\left\{1 - \left|x - \frac{i-1}{k-1}\right|, (k-1)\right\} \quad (2)$$

همانطور که در شکل ۱ نشان داده شده است نتیجه‌ی اجرای تابع عضویت مثلثی رابطه‌ی ۲ برای مقادیر $K=3, i=1,2,3$ در بازه‌ی $[0, 1]$ آورده شده است. منطقی که این تابع عضویت ارائه می‌دهد این است که هر الگویی با یک درجه‌ی عضویت متعلق به همه پارتیشن‌ها می‌باشد حال هرچقدر الگویی نزدیک به مرز پارتیشن‌ها باشد با درجه‌ی عضویت نزدیکی متعلق به هردو پارتیشن همسایه می‌باشد.



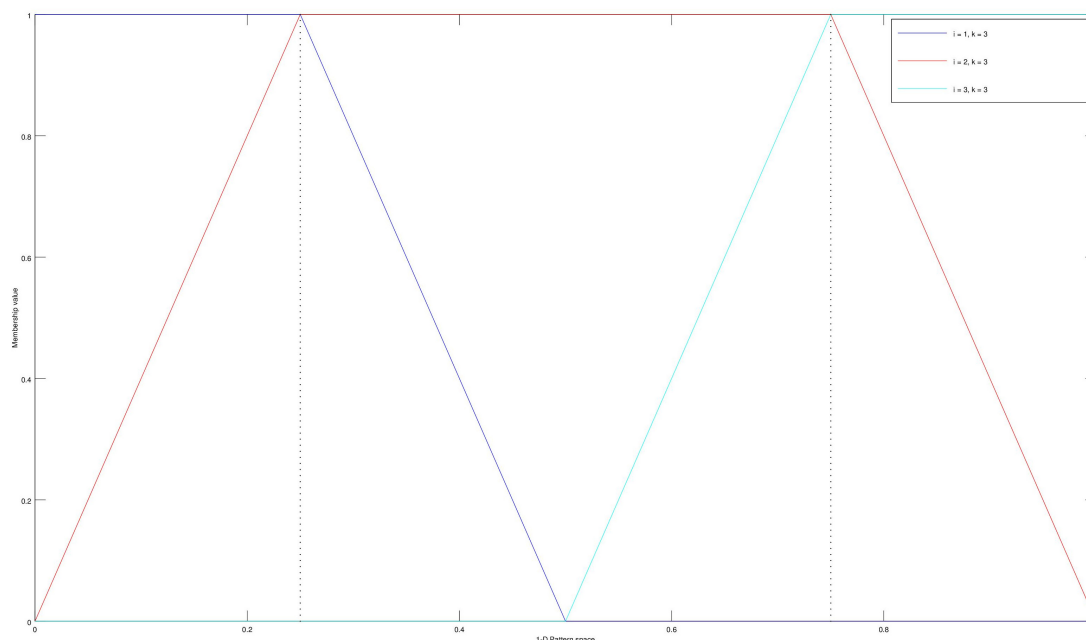
شکل ۱: اجرای تابع عضویت مثلثی برای بازه‌ی $[0, 1]$ برای $K=3$ و $i=1,2,3$

۲.۱.۴ تابع عضویت دوزنقه

رابطه تابع عضویت دوزنقه مورد استفاده در ۳ آمده است.

$$\mu_i^K(x) = \max\{\min\{2 - 2\left|x - \frac{i-1}{k-1}\right|(k-1), 1\}, 0\} \quad (3)$$

همانطور که در شکل ۲ نشان داده شده است نتیجه‌ی اجرای تابع عضویت دوزنقه رابطه‌ی ۳ برای مقادیر $K=3, i=1,2,3$ در بازه‌ی $[0, 1]$ آورده شده است. منطقی که این تابع عضویت ارائه می‌دهد این است که هر الگویی با درجه‌ی عضویت ۱ متعلق به پارتیشن‌ی که در آن هست می‌باشد و بایک درجه‌ی عضویت متعلق به دیگر پارتیشن‌ها می‌باشد، بطوری که هرچه قدر از پارتیشن‌ی که متعلق به آن است دورتر شویم درجه‌ی عضویت آن کاهش می‌یابد.



شکل ۲: اجرای تابع عضویت دوزنقه برای بازه‌ی $[0, 1]$ برای $K=3$ و $i=1,2,3$

۲.۴ استخراج قوانین

برای استخراج قوانین به ازای هریک از پارتیشن‌ها میزان سازگاری نمونه‌ها با توجه به درجه‌ی عضویت آن‌ها در آن پارتیشن برای هریک از کلاس‌ها محاسبه می‌کنیم سپس هرکدام از کلاس‌ها که میزان سازگاری بیشتری نسبت به آن پارتیشن را داشت به عنوان برچسب آن پارتیشن با یک درجه‌ی اطمینان در نظر می‌گیریم. میزان سازگاری کلاس‌ها در روابط ۴ و ۵ آمده‌اند. همان‌طور که در روابط ۴ و ۵ می‌بینیم مقادیر β_{G1}, β_{G2} وابسته به

داده‌های آموزشی و مقدار K و پارتیشن‌ای که در حال واری است (i, j) ، می‌باشد.

$$\beta_{G1} = \sum_{p \in G1} \mu_i^K(x_{1p}) \cdot \mu_j^K(x_{2p}) \quad (4)$$

$$\beta_{G2} = \sum_{p \in G2} \mu_i^K(x_{1p}) \cdot \mu_j^K(x_{2p}) \quad (5)$$

حال بعد از محاسبه‌ی مقادیر β_{G1}, β_{G2} به ازای هریک از پارتیشن‌ها پارتیشن‌هایی که مقدار β_{G1} بیشتری نسبت به مقدار β_{G2} دارند به عنوان کلاس $G1$ برچسب زده می‌شوند و آن پارتیشن‌هایی که مقدار β_{G2} بیشتری نسبت به مقدار β_{G1} دارند به عنوان کلاس $G2$ برچسب زده می‌شوند. آن پارتیشن‌هایی که مقادیر β_{G1} برابر با β_{G2} می‌باشد را نمی‌توان برچسب دهی کرد. میزان درجه‌ی اطمینان برچسب زده شده به هریک از پارتیشن‌ها نیز وابسته به مقادیر β_{G1} و β_{G2} می‌باشد که در رابطه‌ی ۶ آورده شده است.

$$CF_{ij}^K = \frac{|\beta_{G1} - \beta_{G2}|}{\beta_{G1} + \beta_{G2}} \quad (6)$$

در حالت کلی قوانین هر پارتیشن به صورت زیر بدست می‌آید.

$$\text{If } x_{1p} \text{ is } A_i^K \text{ and } x_{2p} \text{ is } A_j^K \text{ then } x_p \text{ belongs to } \begin{cases} G1 & \text{if } \beta_{G1} > \beta_{G2} \\ G2 & \text{if } \beta_{G1} < \beta_{G2} \\ Noone & \text{if } \beta_{G1} = \beta_{G2} \end{cases} \quad \text{with CF} = CF_{ij}^K.$$

در روابط ۴ و ۵ می‌توان از عملگر min استفاده کرد که آنگاه روابط ۴ و ۵ به صورت روابط ۷ و ۸ خواهند بود.

$$\beta_{G1} = \sum_{p \in G1} \mu_i^K(x_{1p}) \wedge \mu_j^K(x_{2p}) \quad (7)$$

$$\beta_{G2} = \sum_{p \in G2} \mu_i^K(x_{1p}) \wedge \mu_j^K(x_{2p}) \quad (8)$$

۳.۴ استنتاج و طبقه‌بندی

بعد از اینکه قوانین فازی که در قسمت ۲.۴ معرفی شد، استخراج شدند می‌توان با استفاده از قوانین استخراج شده، نمونه‌های جدید برچسب نشده را طبقه‌بندی کرد. بدین صورت که میزان سازگاری نمونه جدید را با قوانین هریک از کلاس‌ها سنجیده و سپس برچسب نمونه را به کلاسی نسبت می‌دهیم که قوانین حاکم بر آن کلاس دارای بیشترین سازگاری با نمونه را دارند. میزان سازگاری قوانین کلاس‌های $G1, G2$ به ترتیب در روابط ۹ و ۱۰ آمده است.

$$\alpha_{G1} = \max\{\mu_i^K(x_{1p}) \cdot \mu_j^K(x_{2p}) \cdot CF_{ij}^K \mid G_{ij}^K = G1; \quad i, j = 1, 2, \dots, K; \quad K = 2, 3, \dots, L\} \quad (9)$$

$$\alpha_{G2} = \max\{\mu_i^K(x_{1p}) \cdot \mu_j^K(x_{2p}) \cdot CF_{ij}^K \mid G_{ij}^K = G2; \quad i, j = 1, 2, \dots, K; \quad K = 2, 3, \dots, L\} \quad (10)$$

همان طور که در روابط ۹ و ۱۰ می بینیم درجه ی اطمینان بدست آمده از بخش استخراج قوانین در محاسبه ی میزان سازگاری نمونه با قوانین استفاده شده است و قانونی با بیشترین درجه ی اطمینان نقش بیشتری بر میزان سازگاری یک نمونه نسبت به بقیه بازی می کند.

بعد از اینکه مقادیر α_{G1}, α_{G2} را بدست آوردیم، نمونه با یک درجه ی اطمینان متعلق به کلاسی است که مقدار α از دیگری بیشتر است. میزان درجه ی اعتبار یا اطمینان طبقه بند نیز وابسته به مقادیر α_{G_i} می باشد. بطوری که هر قدر میزان سازگاری قوانین کلاس ها با یک نمونه دارای اختلاف بیشتری باشند، با اطمینان بیشتری می توانیم طبقه بندی کنیم. به عبارت دیگر:

$$\text{The new unseen input } x_p \text{ belongs to } \begin{cases} G1 & \text{if } \alpha_{G1} > \alpha_{G2} \\ G2 & \text{if } \alpha_{G1} < \alpha_{G2} \\ Noone & \text{if } \alpha_{G1} = \alpha_{G2} \end{cases} \quad \text{with grade of support } |\alpha_{G1} - \alpha_{G2}|.$$

برای زمانی که برای استخراج قوانین از عمگر min استفاده شده باشد باید برای محاسبه ی روابط ۹ و ۱۰ که عمگر ضرب استفاده شده بود از عمگر کمینه استفاده شود. در این صورت باید از روابط ۱۱ و ۱۲ بجای ۹ و ۱۰ استفاده کنیم.

$$\alpha_{G1} = \max\{[\mu_i^K(x_{1p}) \wedge \mu_j^K(x_{2p})].CF_{ij}^K \mid G_{ij}^K = G1; i, j = 1, 2, \dots, K; K = 2, 3, \dots, L\} \quad (11)$$

$$\alpha_{G2} = \max\{[\mu_i^K(x_{1p}) \wedge \mu_j^K(x_{2p})].CF_{ij}^K \mid G_{ij}^K = G2; i, j = 1, 2, \dots, K; K = 2, 3, \dots, L\} \quad (12)$$

نحوه ی استنتاج از مقادیر روابط ۱۱ و ۱۲ همانند روند استنتاجی که در مورد روابط ۹ و ۱۰ معرفی شد می باشد.

۵ پیاده‌سازی مقاله

در پیاده‌سازی انجام شده توابع در حالت کلی به ۳ دسته تقسیم می‌شوند - قسمتی برای استخراج قوانین فازی و قسمتی برای استنتاج از قوانین بدست آمده در بخش قبل و دسته‌ای از توابع ابزاری^۳ که در هر دو قسمت قبلی مورد استفاده قرار می‌گیرند. که هریک از این بخش‌ها به ترتیب در زیر توضیح داده می‌شود.

۱.۵ استخراج قوانین

۱.۱.۵ تابع استخراج قوانین

تابع `gen_rules` تنها تابعی که برای استخراج قوانین مورد استفاده قرار می‌گیرد تابع `gen_rules(.)` می‌باشد. که این تابع ۷ عدد ورودی می‌گیرد که در جدول ۱ شرح داده شده‌اند.

Argument	Type	Comment
<code>l_max</code>	Integer	The maximum L value to partion the patterns' space
<code>x1</code>	float	The patterns' first demintions' value
<code>x2</code>	float	The patterns' first demintions' value
<code>func</code>	function	The classify function(The problems 1...7)
<code>mf_func</code>	function	The membership function(triangular or trapezoid)
<code>mf_opt</code>	function	The operation function for membership-values arithmetic(product or min)
<code>apply_cf</code>	boolean	Indicator that if applying CF is in effect or not?(true or false)

Table1: `function rules = gen_rules(l_max,x1,x2, func, mf_func,mf_opr,apply_cf)`

خروجی تابع `function rules = gen_rules(.)` یک ماتریسی عددی^۴ از قوانین است که فرمت هر قانون این ماتریس به صورت `[k i j g cf]` می‌باشد که شرح هرکدام از المان‌های قانون در جدول ۲ آمده است.

Label	Type	Comment
<code>k</code>	integer	The <i>K</i> value that rule has been generated within
<code>i</code>	integer	The region number in first dimension
<code>j</code>	integer	The region number in second dimension
<code>g</code>	[1 or 2]	The class number
<code>cf</code>	float	The confidence level of rule

Table2: Description of generated rule's format

^۳Utility Functions
^۴Numerical

۲.۵ استنتاج از قوانین

۱.۲.۵ تابع استنتاج از قوانین و طبقه‌بند

تابع classifier تنها تابعی که مسئولیت طبقه‌بندی داده‌ها را دارد این تابع می‌باشد و دارای ۵ عدد ورودی می‌باشد که در جدول ۳ شرح داده شده‌اند.

Argument	Type	Comment
rules	Integer	The generated rules
xp	Matrix	The un-classified patterns
mf_func	function	The membership function(triangular or trapezoid)
mf_opt	function	The operation function for membership-values arithmetic(product or min)
apply_cf	boolean	Indicator that if applying CF is in effect or not?(true or false)

Table3: `function z = classifier (rules, xp, mf_func,mf_opr,apply_cf)`

خروجی تابع `(.) z = classifier` یک ماتریس کلاسی می‌باشد هر سطر این ماتریس متناظر با یک سطر از ماتریس الگوهای ورودی تابع می‌باشد و فرمت هر سطر به صورت `[g cf]` می‌باشد که شرح هرکدام از المان‌ها در جدول ۴ آمده است.

Label	Type	Comment
g	[1 or 2]	The class number
cf	float	The confidence level of classification

Table4: Description of classification's output's format

۳.۵ توابع ابزاری

توابع ابزاری بیشتر برای کارهای ابتدایی مانند تولید نمونه و توابع عضویت و ... می‌باشند که از درجه اهمیت کمتری نسبت به توابع ذکر شده برخوردارند که بجز تابع راه‌انداز^۵ `[frpc(.)]` فقط به نام‌بردن و ذکر هدف دیگر توابع ابزاری بسنده می‌کنیم.

از میان توابع جدول ۵ تابع راه‌انداز که وظیفه‌ی راه‌اندازی پروژه و انجام عملیات تولید قوانین فازی و تست قوانین و چاپ نتایج حاصل از تست دارد به عهده دارد نیاز به توضیح بیشتری راجع به فازهای اجرایی خود دارد.

^۵Bootstrapper

Function	Return Type	Comment
frpc	void	The bootstrapper (to be explained).
gen_data	Matrix	Generates random data in $[0, 1] \times [0, 1]$ pattern space with uniform distribution.
value2class	[1 or 2]	Converts its input values into 2 class-type [1 or 2].
mf_trapezoid	float	The trapezoid membership function.
mf_triangular	float	The triangular membership function.
print_rules	void	Prints the passed rules in manner of a nice verbal format.
print_classification_res	void	Prints the classification results in manner of a nice verbal format.

Table5: List of utility functions

۱.۳.۵ تابع راه انداز

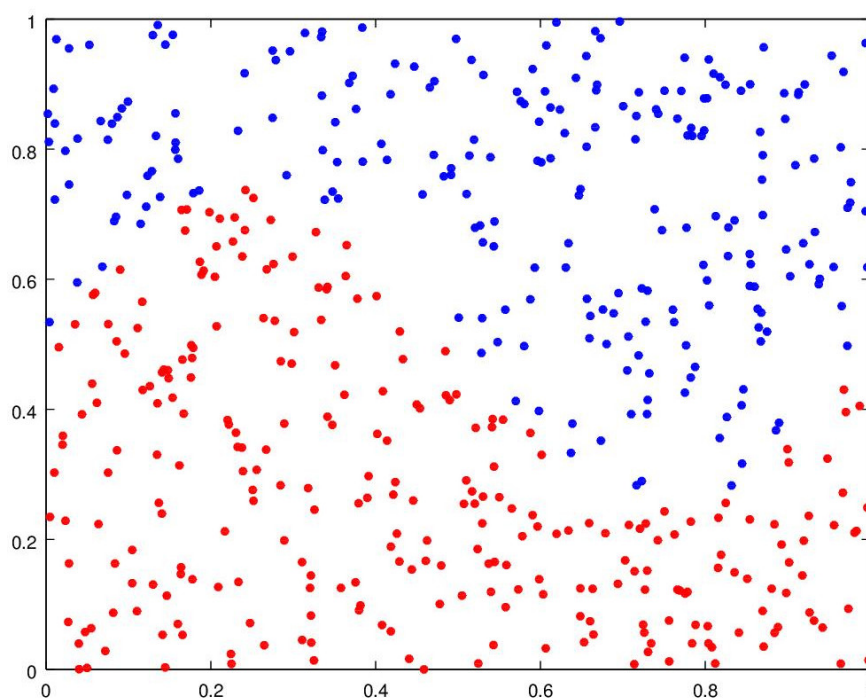
تابع frpc به عنوان ورودی یک مقدار $lmax$ دریافت می کند که مقدار پیش فرض این ورودی ۱۰ می باشد. سپس با یکی از توابع مسائل ۱...۷ که در اینجا تابع مساله ی ۱ مورد استفاده قرار گرفته شده است. سپس برای انجام آزمایشات یک داده ی تست به صورت تصادفی تولید شده است که همه آزمایشات روی یک داده تست انجام شده باشند که نتایج آزمایشات قابل مقایسه با یک دیگر باشند.

طبق جدول ۱ مقاله که از دو تیپ-تعدادی داده ی آموزشی برای تولید قوانین استفاده کرده است، که به صورت یک داده ی آموزشی ۴۰ تایی و یک داده ی آموزشی ۱۰۰ تایی و در هر دوی این ها تعداد داده های تست ۱۰۰ عدد هستند. بدین منظور بنده نیز ۲ دسته ۴۰ تایی و ۱۰۰ تایی به عنوان داده ی آموزشی در داخل یک حلقه تولید میکنم. سپس به ازای هر داده آموزشی ۶ تست مطرح در جدول شماره ی ۱ مقاله را در داخل حلقه دوم اجرا میکنم که مقادیر متغیرهای در تست از تابع get_opt که وابسته به شماره تست (که شماره تست ها متناسب با تست های جدول ۱ در نظر گرفته شده اند). می باشد. سپس با استفاده از داده آموزشی قوانین فازی استخراج شده و سپس قوانین بروی داده ی تست آزموده می شود و سپس نتایج تست چاپ می شوند.

۶ آزمایش اول

جدول ۶ میانگین مساله ی یک مقاله که در رابطه ی ۱۳ آورده شده است. فضای الگوی $[0, 1] \times [0, 1]$ را به صورت غیرخطی به دو بخش مثبت و منفی تقسیم می کند که در شکل ۳ آمده است.

$$f_1(x) = -\frac{1}{4} \sin(2\pi x_1) + x_2 - 0.5 \quad (13)$$



شکل ۳: نمایش افراز فضای الگو $[0, 1] \times [0, 1]$ به دو بخش مثبت (کلاس ۲ - نقاط آبی) و منفی (کلاس ۱ - نقاط قرمز) توسط تابع مساله‌ی ۱

جدول ۶ نتایج چندین اجرای الگوریتم بروی مساله‌ی ۱ مقاله می‌باشد.

Test#	Shape of fuzzy set	Type of operator	Grade of certainty	The number of patterns = 40		The number of patterns = 100	
				Current	Unclass	Current	Unclass
1	Triangular	Product	With CF	95.53	0.0	93.24	0.0
2	Trapezoid	Product	With CF	96.23	0.0	95.32	0.6
3	Triangular	Product	Without CF	94.32	0.0	93.0	0.0
4	Trapezoid	Product	Without CF	69.15	31.0	63.33	36.33
5	Triangular	Min	With CF	95.97	0.0	94.10	0
6	Triangular	Min	Without CF	93.45	0.0	91.0	0

Table6: Average rates of correctly classified patterns and unclassified patterns

۷ جمع‌بندی و نتیجه‌گیری

با توجه به نتایج جدول ۶ مشاهده می‌شود پیاده‌سازی بنده به جز در تست شماره‌ی ۴ (تست با تابع عضویت دوزنقه، اوپراتور ضرب و بدون درجه‌ی اطمینان) در مابقی تست‌ها نتایجی بهتر از نتایج مقاله گرفته است، یک دلیل که چرا ممکن است ظاهراً نتیجه‌ی بهتر حاصل شده است این می‌تواند باشد که این نتایج حاصل از اعمال الگوریتم فقط و فقط روی مساله شماره ۱ مقاله می‌باشد و دیگر مسائل در نظر گرفته نشده است. ولی به قطع میتوان گفت که پیاده‌سازی انجام شده هم‌ارز پیاده‌سازی مقاله می‌باشد. ولی علت اینکه چرا تست شماره‌ی ۴ با اختلاف زیاد هم نسبت به دیگر نتایج حاصل از پیاده‌سازی بنده و هم نسبت به نتایج گزارش شده مقاله کمتر است را نمیدانم، چندین بار کدها را بررسی کردم ولی علت امر برایم آشکار نشد. از آنجایی که کدها بسیار ساخت‌یافته نوشته شده‌اند و هنگام پیاده‌سازی از تکنیک DRY^۶ پیروی شده است، نمی‌توان گفت شاید الگوریتم نوشته شده دارای اشتباه بوده است؛ چون در این صورت نتایج دیگر تست‌ها نیز خراب می‌شد. به هر حال علت اینکه چرا همه تست‌ها نتایجی بهتر از مقاله دادند و فقط تست شماره ۴ نتیجه‌ای بدتر داده است برایم معما باقی ماند.

۸ مراجع

- [1] H.Ishibuchi, K.Nozaki, and H.Tanaka. Distributed representation of fuzzy rules and its application to pattern classification. *Fuzzy Sets Syst.*, 59, 1993.