به نام خدا

Fuzzy methods and systems

Assignment number 5

Masoud Fatemi

Distributed representation of fuzzy rules and its application to pattern classification

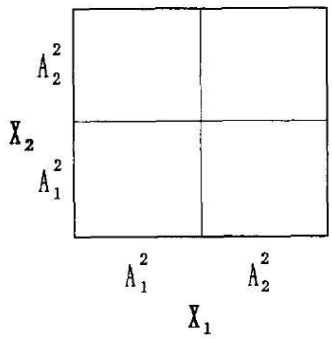
مقدمه

هدف ما در این جا کلاس بندی الگوها در یک فضای ۲ بعدی به ۲ کلاس مختلف با استفاده از از قوانین فازی می‌باشد.برای این منظور با استفاده از داده‌های آموزش که به الگوریتم داده می شوند قوانین فازی برای کلاس بندی استخراج می شوند.سپس در مرحله بعد داده‌های تست را با استفاده از قوانین فازی که در مرحله قبل ساخته ایم در ۲ کلاس مختلف کلاس بندی می‌کنیم و دقت الگوریتم را نیز محاسبه می‌کنیم.

در ابتدا روشی که توسط مقاله ارائه شده به صورت مختصر شرح داده می شود و سپس به صورت دقیق و مرحله به مرحله مراحل انجام کار همراه با پیاده سازی صورت گرفته شرح داده می شوند.

روش ارائه شده توسط مقاله

در این مقاله در ابتدا فرض می‌کنیم ما تعدادی داده ی آموزش به صورت $x_p = (x_{1p}, x_{2p})$ که هرکدام عضو کلاس G1 یا G2 هستند داریم.این داده‌ها در فضایی ۲ بعدی که ابعاد آن را [۰،۱]*[۰،۱] در نظر می گیریم توزیع شده اند.حال فرض می‌کنیم که هر بعد فضای مورد نظر را به L قسمت کوچکتر تقسیم می‌کنیم که در این صورت برای L=2 شکلی مانند شکل زیر به دست می‌آوریم که به آن rule table متناظر با L=2 گفته می شود.



ما برای L=2 چهار ناحیه به دست می‌آوریم که برای آنها قوانین فازی را به صورت زیر می‌نویسیم:

If x_1 is A_i^L and x_2 is A_j^L then ... i,j = 1, 2, ... , L.

در روش پیشنهادی در این مقاله فرض شده است که به صورت هم زمان از L-1، rule table برای کلاس بندی استفاده می‌کنیم که قوانین آن به صورت زیر نوشته می شوند:

If x_1 is A_i^L and x_2 is A_j^L then ... i,j = 1, 2, ..., K ; K = 2, 3, ... , L.

حال اگر فرض کنیم که هر محور فضای الگوهای ما به K قسمت تقسیم شده است برای هر قسمت یک membership function مثلثی یا ذوزنقه با توجه با فرمول‌های زیر در نظر می گیریم که به K و i وابسته می‌باشند. سپس معیاری به نام CF معرفی می شود که درجه اطمینان ما از اینکه هر داده به کدام کلاس G1 یا G2 تعلق دارد را بیان می‌کند

$\mu_i^k(x) = \max\{1 - |x - a_i^k|/b^k, 0\}$ $i = 1, 2, \dots, K$

$\mu_i^K(x) = \max\{\min\{2 - 2|x - a_i^K|/b^K, 1\}, 0\}$ $i = 1, 2, \dots, K$

$a_i^K = (i - 1)/(K - 1)$ $I = 1, 2, \dots, K$

$b^K = 1/(K - 1)$

در نهایت قوانین فازی به صورت زیر نوشته می شوند که با استفاده از تمام داده‌های آموزش به دست می‌آیند:

If x_{1p} is A_i^L and x_{2p} is A_j^L then then x_p belongs to G_{ij}^K with CF = CF_{ij}^K

مراحل استخراج قوانین فازی از داده‌های آموزش بر اساس روش پیشنهادی مقاله به صورت زیر می‌باشد:

۱) از K=2 تا یک L ماکسیمم برای هر K فضای الگوها را به K قسمت تقسیم می‌کنیم و در هر مرحله برای تمام داده‌های کلاس G1 و برای تمام داده‌های کلاس G2 میزان عضویت یا درجه ی تعلق به توابع عضویت متناظر با هر قسمت را حساب کرده و مقدار βی متناظر را بر اساس فرمول زیر که آن را به عنوان سازگاری تعریف می‌کنیم به دست می‌آوریم.

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

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

۲) اگر برای یک ناحیه $\beta_{G1} = \beta_{G2}$ باشد ما برای آن ناحیه هیچ قانونی نمی سازیم ولی اگر بین مقادیر βای محاسبه شده برای کلاس G1 و G2 برای یک ناحیه فازی اختلاف وجود داشته باشد از شرایط زیر برای تعیین کلاس در قسمت متناظر استفاده می‌کنیم:

$\beta_{G1} > \beta_{G2}$ then $G_{ij}^K = G1$

$\beta_{G1} < \beta_{G2}$ then $G_{ij}^K = G2$

۳) برای هر قسمت با توجه با مقادیر سازگاری حساب شده مقدار CF را بر اساس فرمول زیر محاسبه می‌کنیم:

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

در محاسبه مقادیر β برای هر ناحیه می‌توان به جای استفاده از t-norm ضرب بین درجه‌های عضویت از t-norm مینیمم استفاده کرد که در این صورت فرمول‌های محاسبه مقادیر سازگاری به صورت زیر می شوند:

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

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

با جایگزینی K=2,3,…,L بر اساس مراحلی که در بالا ذکر شده قوانین فازی را به شکل زیر از داهای عددی به دست می‌آوریم که از آنها برای کلاس بندی داده‌های تست و یا یک داده ی ناشناخته استفاده می‌کنیم.

If x_{1p} is A_i^L and x_{2p} is A_j^L then then x_p belongs to G_{ij}^K with CF = CF_{ij}^K i,j = 1, 2, …, K ; K = 2, 3, … , L

مراحل به دست آوردن کلاس برای یک داده ی جدید ناشناخته و یا داده‌های تست به صورت زیر می‌باشد:

۱) محاسبه مقادیر α_{G1} و α_{G2} براساس فرمول زیر:

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

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

۲) اگر $\alpha_{G1} > \alpha_{G2}$ نتیجه می گیریم که داده ی مورد نظر به کلاس G1 تعلق دارد با درجه ی حمایت یا اطمینان α_{G1} - α_{G2} ولی اگر $\alpha_{G2} > \alpha_{G1}$ بود نتیجه می گیرم که داده ی مورد نظر به کلاسG2 تعلق دارد با درجه ی حمایت $\alpha_{G2} - \alpha_{G1}$.

اگر در محاسبه ی مقادیر سازگاری از t-norm مینیمم استفاده شود در این صورت برای محاسبه مقادیر α_{G1} و α_{G2} از فرمول زیر استفاده می‌کنیم:

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

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

انتظار می رود که با افزایش مقدار L دقت الگوریتم برای کلاس بندی داده‌های تست بیشتر شود.اما بر اساس شبیه سازی‌های کامپوتری انجام شده مشاهده می شود که در روش پیشنهادی توسط مقاله افزایش مقدار L تاثیری در قدرت الگوریتم نخواهد داشت.

پیاده سازی:

روش پیشنهادی توسط مقاله که در بالا به صورت خلاصه توضیح داده شد در محیط متلب پیاده سازی شده است و فایل آن همراه این گزارش ارسال شده است. در این بخش به صورت مرحله به مرحله روش پشنهاد شده

همراه با پیاده سازی مربوطه شرح داده می شود. پیاده سازی صورت گرفته شامل قسمت‌های زیر می‌باشد که در متن کد نوشته شده هم با همین عناوین با کامنت مشخص شده اند.

Generate Train Data

در این قسمت ما داده‌های آموزش خود را ساخته تا با استفاده از آنها قوانین فازی مورد نیاز برای کلاس بندی را تولید کنیم.یک ماتریس با ۴۰ سطر و ۲ ستون شامل اعداد تصادفی بین ۰ تا ۱ می سازیم و سپس ستون اول این ماتریس را در متغیر x1 و ستون دوم آن را در متغیر x2 میرزیم.بعد از آن با استفاده از problem 1 که توسط مقاله تعریف شده است مقدار خروجی را به ازای هر داده ی ورودی x که شامل ۲ مقدار x1 و x2 است محاسبه کرده و در متغیر problem_number_one_value قرار می دهیم.بعد از آن برای تمام مقادیر محاسبه شده به آنهایی که مثبت هستند کلاس ۱ و به آنهایی که منفی هستند کلاس ۲ را نسبت می دهیم. تا به اینجا ما ۴۰ داده ی دو بعدی در فضای [۰،۱]*[۰،۱] تولید کرده ایم که هر کدام از آنها یا در کلاس ۱ یا در کلاس ۲ قرار دارند.

Plot Train Data

در این قسمت ما داده‌های تولید شده در مرحله ی قبل را جهت پیدا کردن دیدی مناسب و درکی بهتر از موضوع در یک فضای ۲ بعدی ترسیم نموده.وجود این بخش اختیاری می‌باشد و تاثیری در پاسخ نهایی ندارد.

Variables Definition

در این بخش متغیرها و پارامترهای مورد نیاز در ادامه ی الگوریتم تعریف می شوند، در سطر اول این بخش ماتریس داده‌های آموزش که شامل داده‌های ۲ بعدی و کلاس آنها می‌باشد تولید می شود.همچنین ۲ متغیر g1 و g2 که در انتهای این بخش تعریف شده اند شامل داده‌های کلاس g1 و داده‌های کلاس g2 به صورت جداگانه می باشند.در انتهای این قسمت با توجه به این نکته که در مقاله ذکر شده است که افزایش مقدار L تاثیری در قدرت روش کلاس بندی پیشنهادی ندارد لذا در اینجا نیز مانند مقاله میزان ماکسیمم برای L برابر ۲۰ در نظر گرفته می شود.

Create Fuzzy Rules

این بخش از ۳ حلقه ی تکرار تو در تو جهت تولید قوانین فازی استفاده می‌کند.حلقه ی بیرونی مقدار K که برابر تعداد بخش‌های فضای الگویی در هر مرحله می‌باشد را تا رسیدن به مقدار ماکسیمم می شمارد و دو حلقه دیگر تمام قسمت بندی‌های هر مرحله را برای تولید قوانین فازی جستجو می‌کنند.q اندیس شمارشگر تعداد قوانین تولید شده می‌باشد که در ابتدای حلقه سوم قرار گرفته و به ازای تولید هر قانون یکی اضافه می شود. beta_class_g1 و beta_class_g2 همان مقادیر سازگاری می باشند که با توجه به فرمول ارائه شده در

مقاله محاسبه می‌شوند.روش محاسبه مقادیر سازگاری به این صورت است که به ازای هر K وi برای تمام داده‌های کلاس ۱ میزان تعلق آنها در هر دو بعد به membership functionهای مثلثی (یا ذوزنقه ای) متناظر با هر قسمت جداگانه حساب شده و سپس برای هر داده این مقادیر در هم ضرب شده و با مقادیر دیگر داده‌ها جمع شده و به عنوان مقدار سازگاری کلاس ۱ یا کلاس ۲ برای قسمت متناظر برگشت داده می‌شود. پس از محاسبه مقادیر بتا یا سازگاری برای هر ناحیه مقدار confidence با توجه به فرمول ذکر شده در مقاله محاسبه می شود و در متغیر cf قرار می‌گیرد.حال با استفاده از بتاهای محاسبه شده در هر قسمت بندی برای هر ناحیه کلاس هر ناحیه را با توجه به این مقادیر محاسبه می‌کنیم و در واقع برای آن یک قانون فازی تولید می‌کنیم.روش انجام کار بدین صورت است که ۲ بتای محسابه شده را یکدیگر مقایسه کرده هر کدام که بیشتر بود کلاس متناظر با آن را برای قسمت مورد نظر انتخاب می کنیم.اما اگر برای یک منطقه ی خاص مقادیر بتاها با یکدیگر برابر بودند برای آن منطقه ی خاص هیچ کلاسی انتخاب نمی‌شود و در واقع برای آن منطقه هیچ قانونی تولید نمی‌شود.پس از آنکه با مقایسه مقادیر سازگاری (β) کلاس یک منطقه مشخص شد و در واقع قانون فازی آن تولید شد، باید مقادیر مربوط به این کلاس شامل ماتریس نواحی،ماتریس تعداد قوانین،ماتریس cfها و ماتریس K به روز شوند.

در این قسمت از کد نوشته شده برای تولید قوانین فازی در محیط متلب می‌توان با خارج کردن دستوراتی که با fprintf شروع می‌شوند از حالت comment تمامی قوانین فازی تولید شده ی متناظر با هر قسمت بندی را به صورت کامل مشاهده کرد.در شکل زیر جهت نمونه برای L=3 قوانین تولید شده که خروجی حاصل از پیاده سازی می باشند نشان داده شده اند.

```
for k = 2 rules are :
if "i" is 1 and "j" is 1 then class is G2 and CF = 0.770477
if "i" is 1 and "j" is 2 then class is G2 and CF = 0.147534
if "i" is 2 and "j" is 1 then class is G2 and CF = 0.541989
if "i" is 2 and "j" is 2 then class is G1 and CF = 0.264731

for k = 3 rules are :
if "i" is 1 and "j" is 1 then class is G2 and CF = 1.000000
if "i" is 1 and "j" is 2 then class is G2 and CF = 0.782518
if "i" is 1 and "j" is 3 then class is G1 and CF = 0.584817
if "i" is 2 and "j" is 1 then class is G2 and CF = 0.965455
if "i" is 2 and "j" is 2 then class is G2 and CF = 0.366138
if "i" is 2 and "j" is 3 then class is G1 and CF = 0.707941
if "i" is 3 and "j" is 1 then class is G2 and CF = 0.972590
if "i" is 3 and "j" is 2 then class is G1 and CF = 0.304309
if "i" is 3 and "j" is 3 then class is G1 and CF = 1.000000
number of generated rules are : 13
```

Generate Test Data

در این قسمت داده‌های تست ما تولید می شوند که برای کلاس بندی آنها از قوانین فازی به دست آماده در مرحله قبل استفاده می‌کنیم.در این قسمت جهت به دست آوردن دقت الگوریتم در پایان کار نیاز به محاسبه کلاس داده ها نیز داریم تا در انتها با کلاس هایی که بر اساس قوانین فازی به دست می آیند مقایسه شوند و دقت را برای ما اندازه گیری کنند،از این رو به همان روشی که برای داده‌های آموزش عمل کردیم کلاس هر داده محاسبه می شود و در پایان این بخش برداری که شامل کلاس‌ها می‌باشد در متغیر t قرار می‌گیرد.

Plot Test Data

در این قسمت نیز مانند قسمت Plot train data ما داده‌های تولید شده در مرحله ی قبل را جهت پیدا کردن دیدی مناسب و درکی بهتر از موضوع در یک فضای ۲ بعدی ترسیم نموده.وجود این بخش اختیاری می‌باشد و تاثیری در پاسخ نهایی ندارد.

Fuzzy Inference For Test Data Pattern Classification

در این قسمت از پیاده سازی، عمل کلاس بندی داده‌های تست بر اساس قوانین فازی استخراج شده از داده‌های عددی انجام می شود. حلقه ی اول به تعداد قوانین کلاس G1 اجرا می شود یا به عبارت دیگر به دلیل اینکه هر قانون فازی نوشته شده متناظر با یک بلوک خاص در یک قسمت بندی خاص می باشد به ازای تمام بلوک‌هایی که در مرحله ی آموزش از کلاس G1 تشخیص دادیم اجرا می شود.در هر تکرار حلقه مقدار α_{G1} برای تمام داده‌های تست در یک بلوک متناظر با کلاس G1 محاسبه شده و در متغیر z قرار داده می شود و مقادیر قبلی آلفا که متناظر با بلوک‌های دیگر کلاس G1 هستند مقایسه می‌شوند و در نهایت ماکسیمم آنها در ماتریس alpha_g1 قرار می‌گیرد.حلقه ی دوم نیز تمام مراحل حلقه ی اول را ولی این بار برای کلاس G2 انجام می دهد و در انتها مقادیر ماکسیمم α_{G2} را در ماتریس alpha_g2 قرار می دهد.در این بخش همانطور که مشاهده می شود ۲ فرمول برای محاسبه z با یک membership function وجود دارد که اولی مقادیر CF را برای محاسبه در نظر گرفته ولی در فرمول دوم مقدار z بدون در نظر گرفتن CF محاسبه می شود.تفاوت بین این ۲ حالت را در قسمت نتایج مشاهده خواهیم نمود.

در ادامه یک دستور شرطی داریم که ۲ ماتریس alpha_g1 و alpha_g2 را به یکدیگر مقایسه کرده و برای هر داده ی آموزش اگر مقدار α_{G1} بیشتر از مقدار α_{G2} باشد آن داده از کلاس G1 و در غیر این صورت از کلاس G2 خواهد بود.سپس نتیجه ی حاصل از کلاس بندی داده‌های تست به همراه اختلاف مقادیر آلفا برای هر داده ی آموزش که از آن به عنوان grade of support یاد می‌کنیم در ماتریس results قرار می‌گیرد.

Rate Computation

در این قسمت دقت کلاس بندی داده‌های تست یا به عبارت دیگر دقت الگویتم محاسبه می شود. به این صورت که کلاس هایی که برای داده‌های تست به دست آماده اند که در واقع همان ستون اول ماتریس results می باشند در ماتریس u قرار می گیرند،حال اختلاف بین ماتریس u و ماتریس t که در قسمت generate test data تولید کردیم و شامل کلاس واقعی داده‌های تست می‌باشد را محاسبه کرده و دقت روش را به دست می‌آوریم.

Plot Result

در این قسمت داده‌های تست و کلاسی که برای هر کدام از آنها توسط الگوریتم بدست آماده در یک فضای ۲ بعدی ترسیم می شوند.این قسمت نیز تاثیری در پاسخ نهایی ندارد و جهت مقایسه ی ظاهری داده‌های تست به همراه کلاس واقعی در نمودارهای قبلی و کلاس هایی که توسط الگوریتم به دست آماده اند و به دست آوردن دیدی نسبی و ظاهری از دقت اگوریتم پیاده سازی شده است.

Membership Functions

در این بخش ۲ membership function مثلثی و ذوزنقه ای ارائه شده در مقاله جهت محاسبه ی درجه ی عضویت داده‌ها پیاده سازی شده اند،که هر کدام از آنها ۳ مقدار به عنوان ورودی دریافت می‌کنند و مقدار درجه عضویت را بر می گردانند.با توجه به فرمول ارائه شده برای این membership functionها در مقاله و همانطور که در پیاده سازی مشاهده می شود مقدار خروجی یا به عبارت دیگر همان درجه عضویت در هر ۲ تابع به مقادیر K وi بستگی دارد لذا در تقسیم بندی‌های مختلف و نواحی متفاوت که مقادیر K وi مرتبا دچار تغییر می شوند این توابع نیز دائماً دست خوش تغیرات می شوند.

نتیجه گیری:

تعدادی از نتایج به دست آماده به همراه خروجی حاصل از پیاده سازی نشان داده می شوند و مابقی نتایج بدون خروجی مورد برسی قرار می گیرند.

در شکل زیر خروجی حاصل از membership functionهای مثلثی برای ۱۰ تکرار همراه با ۴۰ نمونه ی train و100داده ی تست برایL=20 در ۲ حالت وجود CF شکل a و عدم وجود CF شکل b برای problem 1 که توسط مقاله تعریف شده نشان داده شده است. همان طور که مشاهده می‌شود برای حالت

```
rate_per_each_iterations =

    97    94    94    96    96    95    94    95    90    90

total_rate =

   94.1000
```

a

```
rate_per_each_iterations =

    92    93    90    88    93    92    83    93    93    86

total_rate =

   90.3000
```

b

وجود پارامتر CF الگوریتم عملکرد بهتری داشته است و با دقت بیشتری داده‌های تست را کلاس بندی کرده است.

در ۲ تصویر زیر نیز تمام شرایط مانند مرحله ی قبل می‌باشد جز اینکه در این حالت از ۱۰۰ داده برای آموزش استفاده شده است.مشاهده می شود که مانند حالت قبلی اگر در محاسبات از پارامتر CF استفاده کنیم دقت الگوریتم در کلاس بندی افزایش پیدا می‌کند(تصویر بالایی).نتیجه ی دیگری که از مقایسه‌ی این حالت با حالت قبلی گرفته می شود این است که با افزایش تعداد داده‌های آموزش الگوریتم عمل کرده بهتری داشته و به دقت بهتری دست پیدا کرده است.

```
rate_per_each_iterations =

    95    97    95    95   100    95    95    98    97    97

total_rate =

   96.4000
```

```
rate_per_each_iterations =

    91    89    96    92    95    93    96    91    95    95

total_rate =

   93.3000
```

با مشـــاهده ی خروجی در ۲ حالت مختلف اســـتفاده از membershipهای مثلثی و ذوزنقه ای این نتیجه حاصل می شود که شکل تابع مورد استفاده تاثیر چندانی در خروجی ندارد.در حالت توزیع شده،یعنی حالتی که ما در اینجا برســـی کردیم که در آن یک مقدار ماکسیمم ثابت برای L در نظر گرفته می شود و تا رسیدن K به آن به ازای مقادیر مختلف K قسمت بندی را انجام می دهیم،مقدار L انتخابی تاثیری در نتیجه ی جواب نخواهد داشت.همچنین در حالت‌های مختلف استفاده از t-norm ضرب و مینیمم برای محاسبه سازگاری‌ها مشاهده می شود که t-norm ضرب منجر به دقت بیشتری برای الگوریتم نسبت به مینیمم می شود.