به نام خدا

مسعود فاطمی

بهرام سلامت روندی

گزارش پروژه پایانی درس داده کاوی

Fuzzy Association Rule Mining

1. مقدمه

تحلیل همبستگی در واقع به معنی مطالعه و شناخته ویژگی ها، صفت‌ها و مشخصه هایی می‌باشد که با یکدیگر رخ می دهند.روش‌ها و متدهایی که برای تحلیل همبستگی وجود دارند به عنوان روش‌های تحلیل سبد بازار نیز شناخته می‌شوند. این روش‌ها و متدها به دنبال آشکار کردن و پرده برداشتن از همبستگی و ارتباط بین ویژگی‌های مختلف می‌باشند که در نهایت بتوانند با توجه به این همبستگی‌ها قوانینی را تولید کنند که رابطه ی بین مقدار دو متغیر و یا بیشتر را بیان کنند. قوانین همبستگی که در انتها استخراج می شوند و روابط موجود بین ویژگی‌های مختلف را برای ما توضیح می دهند به فرم تالی → مقدم می باشند که همچنین همراه هر قانون دو مقدار confidence و support نیز به دست می آیند که در ادامه توضیح داده می شوند.

قوانین همبستگی یک کلاس بسیار مهم از قواعد در داده‌ها می‌باشند که توسط جامعه و افراد خبره در زمینه-ی داده کاوی به شدت و به صورت گسترده مورد مطالعه قرار گرفته اند.هدف کلی در این زمینه پیدا کردن وقایع و آیتم‌های می باشد که به صورت مکرر و به صورت هم زمان در یک مجموعه انتقالات یا حالت‌ها رخ می هند.پیدا کردن مواردی که به صورت همزمان اتفاق می‌افتند و با یکدیگر همکاری دارند همبستگی (association) نام دارد.

الگوهای مکرر الگوهایی هستند که در یک مجموعه داده به فراوانی ظاهر می شوند. به عنوان مثال، مجموعه-ای از آیتم ها،مانند شیر و نان که به فراوانی با همدیگر در یک مجموعه داده ی تراکنشی ظاهر می شوند،یا یک توالی مانند در ابتدا خریدن یک کامپیوتر و سپس یک دوربین دیجیتالی و بعد از آن یک کارت حافظه، اگر مکرر در تارخچه پایگاه داده فروشگاه رخ داده باشند،یک الگوی ترتیبی (مکرر) می باشند.

اما روابط هم بستگی که تا اینجا به آن اشاره کردیم به صورت روابط باینری مطرح می شوند که در آن ها یک ویژگی یا یک مشخصه یا حضور دارد یا خیر.برای غلبه بر این مشکل روش تحلیل قوانین همبستگی به صورت فازی یا Fuzzy Association rule mining به وجود آمده است که در آن به آیتم ها و ویژگی ها اجازه داده می-شود متعلق به بازه هایی باشند که این بازه ها می توانند با یکدیگر نیز هم پوشانی داشته باشند که موضوع بحث ما در این پروژه می‌باشند. در این روش آیتم‌ها می توانند در بیش از یک مجموعه عضویت داشته باشند و مشکل حالات باینری را بر طرف کنند.در Fuzzy Association rule میزان درجه ی تعلق یک آیتم به وسیله ی توابع عضویت و همچنین معیارهای اندازه گیری کیفیت قوانین تولید شده به کمک تئوری مجموعه‌های فازی تعریف می شوند.با استفاده از این روش می‌توان قوانینی تولید کرد که در حالت غیر فازی ممکن است اصلا این قوانین وجود نداشته باشند. کشف قوانین همبستگی به منزله‌ی یک مرحله ی بسیار مهم در فرآیند داده کاوی به شمار می روند.

2. تولید قوانین همبستگی فازی

در قسمت قبل مفهوم قوانین همبستگی و قوانین همبستگی فازی به صورت مختصر توضیح داده شد،در این بخش قوانین همبستگی فازی را در یک دیتاست استاندارد که از سایت UCI گرفته شده با روشی که در ادامه

شرح داده می شود به دست می‌آوریم و بعد از آن پیاده سازی روش مورد نظر که در واقع یه روش Fuzzy Association Rule Mining است را مشاهده می‌کنیم.

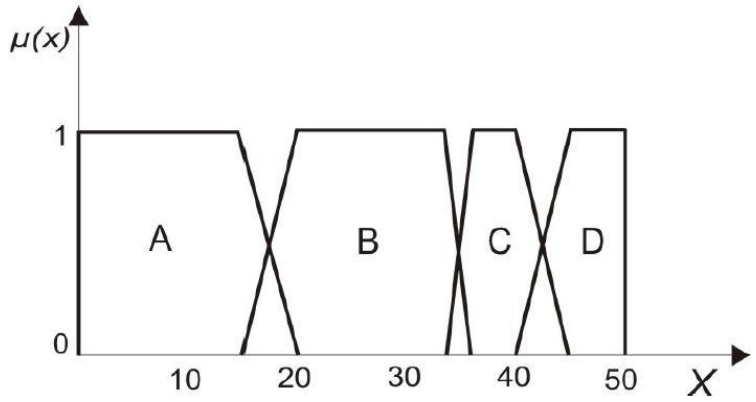
مراحل انجام کار در یک نگاه کلی به این صورت می‌باشند در ابتدا از روی داده‌های کمی و عددی مجموعه‌های فازی را به دست می آوریم سپس یک دیتاست جهت انجام داده کاوی تولید کرده و بعد از آن عملیات فازی مورد نیاز و محاسبات فازی لازم را انجام می دهیم، در ادامه به کمک یک الگوریتم مجموعه آیتم‌های مکرر را تولید و به دست می‌آوریم و در انتها قوانین همبستگی که واقع قوانین همبستگی فازی می‌باشند را به دست می‌آوریم.

پیاده سازی مورد نظر در محیط R صورت گرفته که یک محیط رایگان و بسیار مناسب برای محاسبات آماری و عملیات داده کاوی می‌باشد.پیاده سازی صورت گرفته قابل اجرا بر روی هر پلت فرمی که زبان R بر روی آن نصب شده باشد می‌باشد.

1.2. ساخت مجموعه های فازی

در بیشتر روش‌های بدست آوردن قوانین همبستگی فازی که تا کنون به وجود آماده است از یک فرد خبره در زمینه ی مورد نظر برای تعریف و مشخص کردن مجموعه‌های فازی استفاده شده است که در آنها فرد خبره متناسب با هر ویژگی کمی و عددی مجموعه فازی آن را تعریف کرده است.در تعدادی از روش‌های موجود نیز تلاش شده تا با استفاده از برخی‌روش‌ها مانند خوشه بندی مجموعه‌های فازی متناسب با هر ویژگی به صورت اتوماتیک کشف و ساخته شوند.در پیاده سازی صورت گرفته ما به دنبال این هستیم که هر دو روش را برای کاربر ممکن سازیم تا کاربر تصمیم بگیرد که آیا به صورت دستی خواهان ساخت مجموعه‌های فازی می‌باشد و یا می‌خواهد ساخت این مجموعه‌ها را به برنامه واگذار کند.در حالتی که خود کاربر تصمیم به تعریف مجموعه‌های فازی بگیرد او باید برای هر مجموعه مرزهایی مناسب تعریف کند.علاوه بر تعیین دقیقه محل مرزها کاربر باید تعداد مجموعه های فازی برای هر ویژگی را به دقت مشخص کند،چرا که این مجموعه فازی‌ها و مرزهای آنها باید به گونه‌ای انتخاب شود که تمام مقادیر یک ویژگی را پوشش دهند.در زیر یک نمونه از تعریف ۴ مجموعه فازی برای یک ویژگی را مشاهده می کنیم:

$A=\{0,20\}$, $B=\{15,36\}$, $C=\{34,45\}$, $D=\{40,50\}$



اگر کاربر تصـمیم به تعریف مجموعه فازی‌ها نداشته باشد سیستم با روش پیشنهادی که در در ادامه توضیح داده می شــود آنها را به دســت می‌آورد،اما این روش تضــمینی برای صــحیح عمل کردن به ما نمی دهد.روش پیشنهادی تنها برای به بدست آوردن ۳ مجموع فازی به صورت زیر می‌باشد:

مجموعه Low: مرز پایین برابر با مینیم مقدار ویژگی می باشد و مرز بالا به صورت زیر محاسبه می شود:

$$hb = mean\ -\frac{sd}{2} + mean * overlap$$

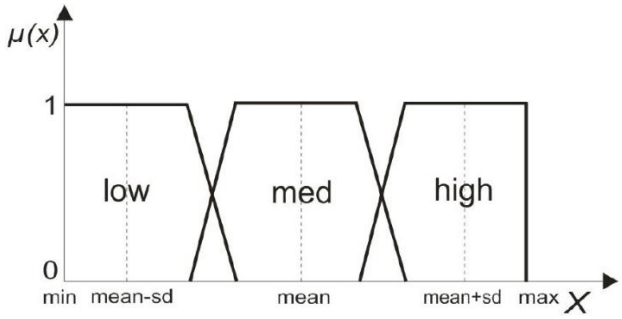
مجموعه Medium: مرز پایین و بالا صورت زیر محاسبه می شوند:

$$lb = mean\ -\frac{sd}{2} - mean * overlap$$

$$hb = mean + \frac{sd}{2} + mean * overlap$$

مجموعه High: مرز بالا برابر ماکزیمم مقدار ویژگی می‌باشد و مرز پایین آن به صورت زیر محاسبه می شود:

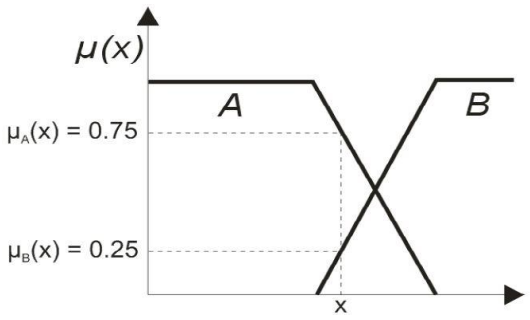
$$lb = \left(mean + \frac{sd}{2}\right) - mean * overlap$$



مقدار همپوشانی (overlap) برابر هر مقداری بین ۰ تا ۱ می تواند باشد.روش ذکر شده بسیار ساده می‌باشد و از دقت پائینی برخوردار می‌باشــد و تنها جهت مشــاهده ی نتایج ارائه شــده اســت و برای پروژه‌های واقعی نیاز به استفاده از الگوریتم‌های بسیار مناسب تری می‌باشد.

2.2. ساخت یک دیتاست جدید برای داده کاوی

پس از تعریف مجموعه‌های فازی متناظر با هر ویژگی نوبت به ساخت یک دیتاست جدید برای اجرای فرآیند Association Rule Mining از روی داده‌های اصلی می‌باشد.این مرحله به سادگی قابل اجرا و محاسبه می‌باشد،به این صورت که مقادیر هر ویژگی را به مجموعه‌های فازی متناظر با آن فیت کرده و درجه عضویت هر داده را به مجموعه‌ها محاسبه می‌کنیم،در زیر یک مثال از این مرحله را مشاهده می‌کنیم:



T={5.27.12.17,22,16,9,14}

A=[0,15] B=[11,23] C=[20,30]

A	B	C
1	0	0
0	0	1
0.75	0.25	0
0	1	0
0	0.33	0.67
0	1	0
1	0	0
0.25	0.75	0

3.2. محاسبه ی عملگرهای فازی

در این مرحله در ابتدا باید تصـمیم گرفت که از کدام t-norm برای محاسـبه معیارهای فازی Confidence و Support می‌خواهیم اسـتفاده کنیم.در روش پیشنهادی از Lukasiewicz t-norm برای این مقادیراستفاده می-شود که فرمول آن را در زیر ارائه می کنیم:

$$supp(A \to B) = \sum_{(x,y)\in D} \min(A(x), B(x))$$

$$conf(A \to B) = \frac{\sum_{(x,y)\in D} \min(A(x), B(x))}{\sum_{(x,y)\in D} A(x)}$$

4.2. تولید مجموعه آیتم های مکرر

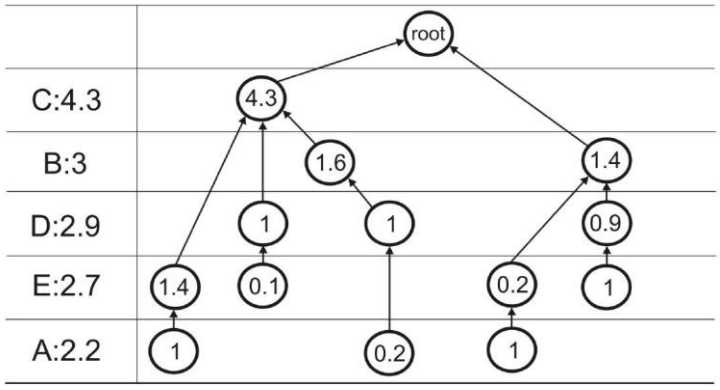
برای تولید مجموعه آیتم‌های مکرر از پایگاه داده، الگوریتم FP-Growth انتخاب شـده است. تنها مشکلی که در اینجا ظاهر می‌شـود تغییر الگوریتم به نحوی می‌باشـد که توانایی عمل کردن بر روی پایگاه داده ی فازی را داشته باشد.خوش بختانه این تغییر برای الگوریتم مورد نظر به سادگی قابل اجرا می‌باشد چرا که داده‌های فازی

به آسـانی توانایی تشـکیل یک FP-Tree را دارند و کاوش روی درخت سـاخته شـده تفاوت چندانی با الگوریتم اصـلی ندارد. سـاخت یک FP-Tree از روی یک پایگاه داده ی فازی کار سر راستی به حساب می‌آید.تنها تفاوت این قسمت با الگوریتم اصلی این می‌باشد که علاو بر به حساب آوردن آیتم هایی که به صورت هم زمان رخ می-دهند باید مقادیر درجه‌های عضـویت نیز مد نظر قرار بگیرند.این مقادیر درجه ی عضـویت در نتیجه به اسانی به مقدار کلی نودهای درخت اضافه می‌شوند.

برای مثل پایگاه داده ی فازی زیر را در نظر بگیرید که از آن برای ساخت یک FP-Tree استفاده می شود:

A	B	C	D	E
1	0	0.3	0	1
0	0.6	1	0	0
0.2	1	1	1	0
0	0.4	0	0.9	1
0	0	1	0	0.4
0	0	1	1	0.1
1	1	0	0	0.2

ساخت یک FP-Tree از این پایگاه داده منجر به درخت زیر می شود که شامل مجموع مقادیر فازی در ندهایش می‌باشد.در شکل زیر درخت سخته شده را مشاهده می‌کنید.حال از این درخت برای اجرای الگوریتم FP-Groth استفاده می شود که در زیر به صورت مختصر آن را توضیح می دهیم.



FP-growth که در اینجا اسـتفاده می شـود بسیار مشابه مدل حالت باینری آن می‌باشد.تفاوت آنها در این است که در اینجا برای محاسبه معیار Support، ساخت الگوهای شرطی پایه و FP-Tree شرطی جدید از مقادیر فازی اسـتفاده می‌کند.مقدار مینیمم Support می تواند با مقایسه مینیمم یک مسیر در درخت با مینیمم Supportی

که توسط کاربر تعریف شده است چک شود.در شکل زیر الگوهای شرطی پای برای آیتم‌های درخت بالا نشان داده شده اند.

item	conditional pattern base
A	$\{\langle C, E:1\rangle, \langle C, B, D:0.2\rangle, \langle B, E:0.2\rangle\}$
E	$\{\langle C:1.4\rangle, \langle C, D:0.1\rangle, \langle B:0.2\rangle, \langle B, D:0.9\rangle\}$
D	$\{\langle C:1\rangle, \langle C, B:1\rangle, \langle B:0.9\rangle\}$
B	$\{\langle C:1.6\rangle\}$
C	$\emptyset$

توجه به این نکته مهم است که بر خلاف روش کلاسیک مقادیر کمتر از نودهای برگ می توانند در یک مسیر در درخت ظاهر شوند و این دلیل این است که ما نمی توانیم از مقادیر برگ‌ها برای ساخت الگوهای شرطی استفاده کنیم و به جای آن باید از مقدار مینیمم یک مسیر استفاده کنیم.

5.2. تولید قوانین همبستگی

تولید قوانین همبستگی در این مرحله همان روش معمول تولید این قوانین است که به صورت کامل در کلاس شرح داده شده است.قسمت مقدم هر قانون می تواند به تعداد دلخواه آیتم داشته باشد اما در قسمت تالی تنها یک آیتم باید وجود داشته باشد.برای هر مجموعه آیتم تک تک آیتم‌های آن باید در قسمت تالی قرار گیرند و میزان Confidence قانون مورد نظر با مینیمم Confidence که از پیش تعیین شده مقایسه شود و در صورت بیشتر بودن Confidence قانون مورد نظر آن را در سیستم ذخیره می‌کنیم.

3. پیاده سازی

در این بخش عملکرد پیاده سازی صورت گرفته به صورت مرحله به مرحله همراه با جزیات شرح داده می-شود.پایگاه داده ای که در این بخش از آن استفاده می‌کنیم یکی از پایگاه داده‌های سایت UCI تحت عنوانه Adultuci می‌باشد که شامل سه فایل عددی می‌باشد.قبل از اجرای پیاده سازی صورت گرفته بر روی این پایگاه داده نیاز به مقداری پیش پردازش می‌باشد چرا که بسیار مهم است که پایگاه داده‌ای که از آن برای بد- دست آوردن قوانین همبستگی فازی استفاده می‌کنیم فقط شامل ویژگی‌های عددی باشد.

اولین تابعی که در اینجا از آن استفاده می‌کنیم و آن را معرفی می‌کنیم تابع getFuzzySets می‌باشد که شامل چند زیر تابع تحت عناوین getCenters، getLow، getMedium و getHigh می‌باشد، که همانطور که در

بخش قبل ذکر شد برای هر ویژگی سه مجموعه فازی به همراه بازه ی مورد نظر برای هر مجموعه را حساب می‌کند و نتایج حاصل را نشان می دهد.اما برای اینکه برای تمام ویژگی‌ها این مجموعه فازی‌ها را حساب کنیم از getAllSets استفاده می کنیم.

computeIndividualMem: این تابع مقدار درجه ی عضویت برای یک داده از دیتاست اصلی به مجموعه فازی‌های به دست آماده در مرحله قبل را برای ما محاسبه می‌کند.این تابع مقدار داده ی مورد نظر و مجموعه فازی‌ها را به عنوان ورودی دریافت می‌کند.تا اینجا ما درجه عضویت برای یک داده را محاسبه کردیم،سپس با استفاده از createMembership تمام این مقادیر درجه های عضویت محاسبه شده در مرحله ی قبل را در کنار یکدیگر می‌گذاریم.

تابع بعدی تابع generateMineableMatrix می باشد که تمام توابع قبل را با یکدیگر ترکیب می کند و ماتریس نهایی که داده کاوی را برای ما ممکن می سازد را تولید می‌کند. مرحله ی مهم بعدی ذخیره ی دیتاست جدید و مجموعه‌های اصلی می‌باشد که با استفاده از تابع getMatrixIndex انجام می‌شود.

در این مرحله برای محاسبه ی support ما به یک مجموعه از آیتم‌ها احتیاج داریم. تابع generateSupport مقادیر مینیمم مجموعه آیتم‌های مشخص شده در هر ردیف از پایگاه داده ی فازی را برای ما حساب می‌کند و همانطور که پیش از این توضیح داده شد این مقدار محاسبه شده برابر support این مجموعه آیتم می‌باشد.

حال برای محاسبه ی confidence دیگر تنها وجود یک مجموعه داده کافی نیست و ما به یک قانون نیز نیاز داریم که شامل مقدم و تالی باشد.برای تولید یک از تابع makeRule استفاده می شود. قانونی که در این مرحله تولید می شود به عنوان ورودی برای تابع generateConfidence فرستاده می شود تا میزان confidence آن محاسبه شود.مقدار confidence با توجه به فرمولی که در بخش قبل ذکر شد برای هر قانون محاسبه می شود.

مراحل به دست آوردن مجموعه آیتم‌های مکرر و توابع وابسته به آن به دلیل حجم زیاد و محدودیت نوشتاری در اینجا توضیح داده نمی شوند و توضیح آن ها را به صورت کامل و جامعه و شفاف به ارایه ی شفاهی واگذار می‌کنیم.برای به دست آوردن تمام قانون های اولیه و کاندید از مجموعه آیتم‌های مکرر از تابع generateRules استفاده می‌کنیم. این تابع برای هر مجموعه آیتم مکرر تمام قوانین ممکن با یک آیتم تنها در قسمت تالی را تولید کرده و یک لیست برای ما برمی‌گرداند.لیست به دست آماده در این مرحله را به عنوان ورودی برای تابع evaluateRules در نظر می گیریم و این تابع تمام قوانینی که مقدار confidence آنها از مینیمم مقدار مشخص شده توسط کاربر بیشتر است را برای ما برمی‌گرداند.

و در نهایت نوبت به این می‌رسد که مشخص شود کدام ستون‌ها در پایگاه داده ی اصلی متعلق به قوانین کشف شده می‌باشند.این کار به وسیله ی تابع traceBack انجام می شود.این مرحله قوانینی تولید می‌کند که توسط کاربر به راحتی قابل فهم می باشتد و درواقع قوانین نهایی و خروجی سیسیتم می باشند.