مجله تحقیق در عملیات و کاربردهای آن، سال هشتم، شماره ۴ (پیاپی ۳۱)، زمستان ۹۰، ص ص ۴۸-۴۱
شاپا ۷۲۸۶-۲۲۵۱

پیشنهاد شیوه‌ای مبتنی بر الگوریتم PSO چند هدفه جهت استخراج قوانین انجمنی در داده کاوی

مهدی نصیری*، احمد اسمعیلی، بهروز مینایی، ناصر مزینی
دانشکده مهندسی کامپیوتر، دانشگاه علم و صنعت ایران، تهران، ایران

رسید مقاله: سیزدهم تیرماه ۱۳۹۰
پذیرش مقاله: بیست و پنجم آبان ماه ۱۳۹۰

چکیده

استخراج قوانین انجمنی یا همان استخراج روابط ممکن مابین داده های یک پایگاه داده بزرگ، یکی از وظایف مهم و بسیار زمان بری در داده کاوی به شمار می‌آید. بدین منظور، الگوریتم‌های متنوعی پیشنهاد شده‌اند که از نظر سرعت اجرا و کیفیت جواب‌ها با هم متفاوت می باشند. در این مقاله سعی شده است با استفاده از الگوریتم تکاملی ***PSO*** چند هدفه یک روش سریع و با کیفیتی مناسب جهت استخراج این قوانین ارایه شود. این الگوریتم پیشنهادی روی یک مجموعه داده از نوع تراکنشی و با اندازه چهل هزار رکورد آزمایش شده است که نتایج حاصل حاکی از کارآمدی این روش پیشنهادی است.

کلمات کلیدی: قوانین انجمنی، داده کاوی، الگوریتم PSO. چند هدفه.

۱ مقدمه

با رشد سریع در اندازه و تعداد پایگاه داده‌ها، کاوش دانش، قواعد یا اطلاعات سطح بالا از داده‌ها به منظور پشتیبانی از تصمیم‌گیری‌ها و پیش بینی رفتارهای آتی، ضروری به نظر می‌رسد. روش‌های داده کاوی در جهت دستیابی به اهداف بالا به چندین گروه دسته بندی می‌شوند که عبارتند از: رده‌بندی، خوشه بندی، کاوش قواعد انجمنی، تحلیل الگوهای ترتیبی، پیش بینی، به تصویر کشیدن داده‌ها و غیره [۲-۱].

کاوش قوانین انجمنی یکی از وظایف مهم در داده کاوی، روند یافتن روابطی ما بین خصیصه‌ها یا ما بین مقادیر آن ها در یک پایگاه داده بزرگ است که در جهت امر تصمیم گیری کمک ساز باشند. یافتن چنین روابطی داخل یک مجموعه وسیعی از داده‌ها به علت ماهیت نمایی آن کار ساده ای نیست.

* عهده دار مکاتبات
آدرس الکترونیکی: Nasiri_m@iust.ac.ir

Archive of SID
www.SID.ir

این روابط را می‌توان به صورت عبارات *IF-THEN* نمایش داد. به شرط بررسی شده در قسمت IF، مقدم و به آنچه که در قسمت THEN می‌آید، نتیجه گفته می‌شود که در این مقاله آن ها را به ترتیب با A و C نشان می‌دهیم.

بنابراین می توان یک رابطه را به صورت $A \to C$ نشان داد و یک چنین رابطه‌ای که ما بین خصیصه‌های داخل رکوردهای یک پایگاه داده با پیروی از ملاک‌های خاصی برقرار باشد قانون انجمنی گوییم.

نوع دیگری از قوانین، که قوانین رده بندی نامیده می‌شوند، را نیز می‌توان با ساختار مشابهی نمایش داد ولی کاملا با قوانین انجمنی متفاوت هستند. در چنین قوانینی، قسمت نتیجه تنها شامل مقادیری از یک خصیصه از پیش تعیین شده به نام رده است، حال آنکه چنین محدودیتی در مورد قوانین انجمنی برقرار نیست. یعنی هر خصیصه و هر تعداد خصیصه‌ای می‌توان در هر دو قسمت قانون داشت. تنها محدودیتی که در مورد قوانین انجمنی باید برقرار باشد این است که دو طرف رابطه نباید دارای خصیصه مشترکی باشد یعنی $A \cap C = \phi$.

الگوریتم‌های مختلفی به منظور جستجوی این قوانین پیشنهاد شده‌اند [۵-۳] . در بخش بعدی، مقدمه مختصری در رابطه با این الگوریتم ها ارئه می کنیم. همانطور که خواهیم دید، این الگوریتم‌ها محدودیت‌های مربوط به خود را دارند. در این مقاله سعی گردیده است با ارایه روشی بر مبنای یک الگوریتم تکاملی سریع و کارآمدی به نام PSO برخی از این محدودیت‌ها مرتفع شوند.

در ادامه این مقاله ابتدا در بخش ۲ کلیتی از روش‌های موجود کاوش قوانین انجمنی ارایه می‌کنیم. در بخش ۳ توضیحی در مورد الگوریتم تکاملی PSO می‌دهیم. در بخش ۴ به جزئیات روش پیشنهادی خواهیم پرداخت. سپس تحلیل نتایج بدست آمده از اجرای این روش را در بخش ۵ ارایه کرده و در نهایت در بخش ۶ نتیجه‌گیری این مقاله ارایه خواهد شد.

۲ الگوریتم‌های کاوش قوانین انجمنی

الگوریتم‌های موجود برای جستجوی قوانین انجمنی به طور کلی بر مبنای رویکرد پیشنهادی Agrawal می‌باشند[۳,۶]. به عنوان نمونه، Apriori، SETM ، AIS، جستجوی Pincer ، و DIC تعدادی از الگوریتم‌های متداول بر مبنای این رویکرد می‌باشند[۹-۶]. این الگوریتم‌ها روی یک پایگاه داده باینری که پایگاه داده سبد خرید (market basket) نامیده می‌شود بکار می‌روند. به منظور آماده سازی پایگاه داده سبد خرید، هر رکورد از پایگاه داده اصلی به صورت یک رکورد باینری نشان داده می‌شود که در آن فیلدها،که آیتم نامیده می‌شوند، با استفاده از مقداری یکتا از هر پایگاه داده اصلی تعریف می‌شوند. برای یک پایگاه داده با تعداد بسیار زیادی از خصیصه‌ها که هر کدام دارای مقادیر متمایزی می‌باشند، تعداد کل این آیتم‌ها بسیار بزرگ می‌شود. در نتیجه ذخیره این پایگاه داده باینری مورد استفاده در الگوریتم‌های قانون کاوی، یکی از محدودیت‌های الگوریتم‌های حاضر است.

این الگوریتم‌ها در دو مرحله کار می کنند [۳]. مرحله اول برای تولید مجموعه آیتم‌های مکرر است که از تمامی مجموعه آیتم‌های ممکن، با استفاده از سنجشی با نام تعداد پشتیبان (support count) و پارامتری به نام پشتیبان

Archive of SID

www.SID.ir

کمینه (minimum support) که توسط کاربر تعیین می‌شود، به دست می‌آید. تعداد پشتیبان (SUP) یک مجموعه آیتم عبارت است از تعداد رکوردهایی از پایگاه داده که تمامی عناصر آن مجموعه باشد. اگر ارزش پشتیبان کمینه خیلی بالا باشد، تعداد مجموعه‌های آیتم مکرر تولید شده کمتر خواهد بود و در نتیجه منجر به تولید قوانین کمتری می‌شود. به طریق مشابه اگر این مقدار خیلی پایین باشد، تقریبا تمامی مجموعه آیتم‌های ممکن، مکرر شده و در نتیجه تعداد قوانین نهایی تولید شده زیاد خواهد گردید. حال انتخاب قوانین بهتر از میان آن ها مشکل دیگری خواهد بود.

بعد از یافتن مجموعه آیتم‌های مکرر در مرحله اول، شروع به تولید قوانین با استفاده از یک پارامتر دیگری به نام اطمینان کمینه (minimum confidence) که توسط کاربر تعریف می‌شود می‌کنیم. ضریب اطمینان یا دقت پیش بینی یک قانون $A \Rightarrow C$ به صورت رابطه (۱) تعریف می‌شود.

$$Confidence = \frac{SUP(A \cup C)}{SUP(A)} \qquad (۱)$$

محدودیت دیگر این الگوریتم‌ها در نحوه‌ی کد کردن است که در آن ها نمادهای مجزایی برای هر مقدار ممکن خصیصه‌ها استفاده می‌شود. این شمای کد کردن ممکن است برای خصیصه‌های با مقادیر اسمی مفید باشد، ولی برای خصیصه‌های عددی مناسب نیست. زیرا می‌توانند مقادیر مختلفی داخل هر رکورد بگیرند. غیر از این محدودیت‌ها، مشکل دیگری که این الگوریتم‌ها دارند این است که در تولید قوانین، ترتیب آیتم‌ها نقش مهمی در آن ها بازی می‌کند[۳]. حال آنکه روش پیشنهادی در این مقاله دارای این محدودیت نیست.

در روشی که در این مقاله معرفی می‌شود، سعی می‌کنیم تا قوانین انجمنی را بر اساس درجه اطمینان، پشتیبان آن ها با استفاده از الگوریتم بهینه سازی جمعی ذرات (PSO) و با یک کدینگ بسیار ساده و سریع استخراج کنیم به گونه‌ای که قوانین تولید شده دارای طول مناسبی نیز باشند. به همین منظور در بخش بعدی، مقدمه‌ای کوتاه از الگوریتم PSO ارایه می‌کنیم.

۳ الگوریتم بهینه‌سازی جمعی ذرات (PSO)

الگوریتم PSO که توسط Kennedy و Eberhart پیشنهاد شده است[۱۰]، یک تکنیک بهینه سازی تصادفی بر مبنای جمعیت می‌باشد که از رفتارهای اجتماعی دسته پرندگان و ماهی‌ها الهام می‌گیرد. در الگوریتم PSO ابتدا سیستم با یک جمعیتی از جواب‌های تصادفی مقدار دهی اولیه می‌شود و سپس با به روز رسانی نسل‌ها جواب بهینه جستجو می‌شود. بر خلاف روش مشابه، الگوریتم ژنتیک، این الگوریتم هیچ عملگر تکاملی مانند ادغام و جهش ندارد و از سرعت همگرایی بالاتری نسبت به آن برخوردار است. در PSO جواب‌های بالقوه که ذره نامیده می‌شوند، در کل فضای مسئله با دنبال کردن بهینه‌ترین ذره کنونی به حرکت در می‌آیند. اگر یکی از ذرات مسیر خوبی را بیابد, سایر ذرات به دنبال آن ذره حرکت می‌کنند، هرچند که از آن خیلی دور باشند. رفتار جمعی با استفاده از ذرات داخل فضای چندبعدی که دارای دو مشخصه‌ی مکان و سرعت هستند مدل می‌شود.

Archive of SID

www.SID.ir

این ذرات در کل این فضا حرکت می‌کنند و بهترین مکانی را که تاکنون ملاقات کرده‌اند را به خاطر می‌سپارند. آن ها این موقعیت‌های خوب را به اطلاع یکدیگر رسانده و موقعیت و سرعت حرکت خود را براساس این موقعیت‌های خوب تنظیم می‌کنند. اگر بخواهیم این روند را به صورت دقیق‌تر بیان کنیم، مراحل زیر را خواهیم داشت: جمعیت ذرات در ابتدا با جمعیتی تصادفی از جواب‌ها مقداردهی اولیه می‌شوند. این جمعیت اولیه به صورت تکراری در کل فضای جستجو d- بعدی حرکت می‌کنند و به دنبال جواب‌های جدید می‌گردند. برای هر ذره تابع برازندگی f به منظور اندازه گیری کیفیت جواب محاسبه می‌شود تا بهترین ذره مشخص گردد. هر ذره دارای یک مکان و یک سرعت است که به ترتیب توسط بردارهای مکان $\mathbf{x}_i$ (که i اندیس ذره می‌باشد) و سرعت $\mathbf{v}_i$ نشان داده می‌شوند. هر ذره بهترین مکان خود تا لحظه کنونی را در بردار $P_i=(p_{i1}, p_{i2}, \ldots, p_{id})$ و مقدار j-امین بعد خود را در p_{ij} نگهداری می‌کند. بهترین بردار در میان همه‌ی ذرات در بردار P_g ذخیره می‌شود که مقدار j-امین بعد آن در p_{gj} قرار دارد. در طول زمان تکرار t، بروزرسانی سرعت از سرعت قبلی به مقدار جدید توسط رابطه‌ی(۲) و مقدار جدید برای مکان از رابطه‌ی(۳) محاسبه می‌شوند.

$$v_{ij}(t+1) = wv_{ij}(t) + c_1 r_1 (p_{ij}(t) - x_{ij}(t)) + c_2 r_2 (p_{gj}(t) - x_{ij}(t)) \quad (۲)$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \quad (۳)$$

که در آن w ضریب اینرسی نامیده می‌شود. r_1 و r_2 اعداد تصادفی هستند که برای حفظ چگالی جمعیت بوده و به صورت یکنواخت در بازه‌ی [۲-۱] توزیع شده‌اند. c_1 و c_2 ثابت‌هایی مثبت می‌باشند که به ترتیب ضرایب اجزای خود تشخیصی و اجتماعی نامیده می‌شوند. شبه کد زیر مراحل فوق را به صورت خلاصه نشان می‌دهد:

```
Initialize();
for t =1 to the limit of iterations
  for i =1:N
    Fitness_i^(t) = EvaluationFitness (X_i^(t)) ;
    UpdateVelocity (V_i^(t+1)) according to formula (2);
    LimitVelocity (V_i^(t+1)) ;
    UpdatePosition (X_i^(t+1)) according to formula(3);
    if needed, update P_i and P_g;
  End
  Terminate if P_g meets problem requirements;
End
```

٤ روش پیشنهادی

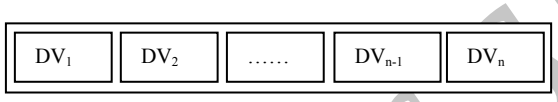
٤-١ پیکربندی یک ذره

در این مقاله ما سعی می‌کنیم قوانین انجمنی را با استفاده از الگوریتم PSO بیابیم. بدین منظور اولین کاری که باید انجام گیرد نمایش دادن یا به عبارت بهتر کد کردن قوانین انجمنی داخل یک ذره است. در کدینگ به کار رفته در این مقاله، هر ذره شامل متغییرهای تصمیمی (DV) است که نماینده آیتم‌ها می‌باشند. به این ترتیب

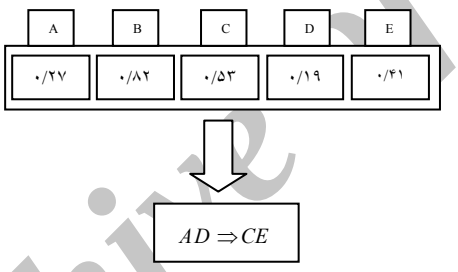
Archive of SID

www.SID.ir

که آیتم i–ام در مکان i–ام ذره قرار دارد. حال متغییر تصمیم i–ام در این ذره مشخص خواهد کرد که آیا خصیصه i–ام جزء خصیصه های قانون نمایش داده شده توسط این ذره است یا خیر و اینکه این خصیصه جزء بخش مقدم قانون است یا جزء نتیجه قانون. به این صورت که، اگر این متغییر مقداری در بازه‌ی $0.00 \le DV_i \le 0.33$ بگیرد به این معنی است که متغیر i–ام در بخش مقدم قانون خواهد آمد. و اگر مقداری در بازه‌ی $0.33 < DV_i \le 0.66$ بگیرد، یعنی متغییر i–ام جزء قسمت نتیجه است و در نهایت اگر $0.66 < DV_i \le 1.00$ باشد، آن متغییر در آن قانون ظاهر نخواهد شد. در نتیجه تمامی متغیرهای تصمیمی که در بازه مابین ۰/۰۰ و ۰/۳۳ مقدار بگیرند قسمت مقدم قانون را خواهند ساخت در حالی که متغییرهای دارای مقداری بین ۰/۳۳ و ۰/۶۶، قسمت نتیجه را می‌سازند. در شکل (۱) پیکربندی یک ذره جهت نمایش یک قانون را می‌بینیم و شکل (۲) مثالی از نحوه تبدیل کدینگ یک ذره به یک قانون را نشان می‌دهد.



شکل ۱. پیکربندی یک ذره جهت نمایش یک قانون



شکل ۲. مثالی از نحوه تبدیل یک ذره به یک قانون

همان طور که مشاهده می‌شود، در این روش نمایش طول ذره برابر خواهد بود با تعداد خصیصه‌های موجود در پایگاه داده.

٤-۲ طراحی تابع برازندگی

هدایت‌گر ذرات داخل جمعیت به سمت یک جواب بهینه با استفاده از تابع برازنگی می‌باشد. در اینجا به طور معمول فرض می‌شود که ذره با بزرگترین مقدار برازندگی، بهترین ذره می‌باشد. در روش پیشنهادی اگر فرض کنیم A و C به ترتیب مجموعه خصیصه‌های شرکت کننده در قسمت مقدم و نتیجه قانون باشند که با استفاده از دیکد کردن ذره متناظر، طبق روش بخش قبل به دست آمده‌اند تابع برازندگی را به صورت زیر تعریف می‌کنیم:

$$Fitness = \alpha_1 \times \frac{Num(AUC)}{DS} + \alpha_2 \times \frac{Num(AUC)}{Num(A)} - \alpha_3 \times NA \qquad (۴)$$

Archive of SID
www.SID.ir

که در این رابطه Num(X) تابعی است که تعداد رکوردهای شامل تمامی خصیصه‌های مجموعه X را بر می‌گرداند، DS اندازه مجموعه داده است، NA تعداد خصیصه های شرکت کننده در قانون تولید شده بوده و ضرایب α به منظور کنترل تاثیر هریک از پارامترهای داخل تابع برازندگی می‌باشند و بنابر نیاز کاربر می‌توانند به دلخواه تنظیم شوند.

همانگونه که دیده می‌شود، قسمت‌های اول و دوم این تابع به ترتیب مربوط است به محاسبه‌ی مقدار پشتیبان و درجه اطمینان قانون تولید شده. در نظر گیری این دو مولفه با هم در محاسبه برازندگی قانون تولید شده ضروری به نظر می‌رسد زیرا درجه اطمینان و یا پشتیبان به تنهایی نمی توانند معیاری برای قضاوت روی کیفیت قانون تولید شده باشند. بدیهی است قانونی از درجه کیفیت بالایی برخوردار است که این دو فاکتور باهم در آن از مقدار بالایی برخوردار باشند. از سویی دیگر می‌دانیم که در قوانین با طول زیاد احتمال وجود خصیصه‌های زائد که باعث کاهش کیفیت جواب تولید شده می‌شوند زیاد است. در نتیجه در قسمت سوم سعی می کنیم قوانین با طول نسبتا کوتاه و در نتیجه با قابلیت خوانایی، درک و کیفیت بالاتر که در داده کاوی از اهمیت ویژه‌ای برخوردار است، تولید نماییم.

۵ نتایج شبیه سازی

ما روش پیشنهادی خود را روی چند مجموعه داده پیاده سازی و آزمایش کردیم که نتایج حاصل از آن ها بسیار راضی کننده بود. در این مقاله برای نمونه نتیجه کار روی یک مجموعه داده با ۴۰۰۰۰ رکورد از نوع تراکنشی که دارای ۲۹ خصیصه می‌باشد، نشان می‌دهیم. در پیاده سازی روش پیشنهادی، پارامترهای الگوریتم PSO را به صورت جدول (۱) می گیریم.

جدول ۱. پارامترهای الگوریتم PSO

V_{max}	تابع اینرسی	نرخ‌های یادگیری c2,c1	تعداد تکرارها	اندازه جمعیت	پارامترها
۰/۵	۰/۹–۰/۴	۲	۵۰۰	۲۵	مقادیر

ضرایب α_1، α_2 و α_3 مربوط به تابع برازندگی در این آزمایش نیز به ترتیب مقادیر ۰/۸، ۰/۸ و ۰/۰۵ گرفته‌اند. به علت بزرگی مجموعه داده برای بارگذاری در حافظه، ما در هر مرحله نمونه‌ای با اندازه مشخص را به صورت تصادفی انتخاب کرده و الگوریتم پیشنهاد شده در این مقاله را روی آن اجرا می کنیم. در هر اجرا ما شرط خاتمه را بر اساس بیشینه مقدار تعداد قوانین انجمنی تولید شده کلی در حین اجرا می‌گیریم. از میان قوانین تولید شده، قوانین تکراری و قوانین با درجه اطمینان زیر ۸۵٪ و پشتیبان زیر ۲۰٪ را حذف کرده و باقی مانده‌ها را به عنوان قوانین نهایی معرفی می‌کنیم. نتایج حاصل در جدول (۲) آمده است. در دو ستون آخر این جدول علاوه بر پارامترها و نتایج گفته شده، میانگین درجه اطمینان و مقدار پشتیبان قوانین نهایی داخل هر زیر گروه را نیز آورده‌ایم.

از روی مجموعه قوانین تولید شده برای نمونه‌های مختلف و با بیشینه قوانین مختلف، می‌توان مشاهده کرد که بعد

www.SID.ir

از حداکثر ۲۰۰ قانون بدون محدودیت تولید شده، الگوریتم قوانین چندان زیادی تولید نمی‌کند. به عبارت دیگر بعد از این تعداد الگوریتم تقریبا همگرا می‌شود. به عنوان مثال در مورد نمونه سوم تنها ۱ قانون بیشتر با قیمت سنگین ۱۰۰ بار اجرای اضافی تولید می‌شود. قابل ذکر است که به علت نرمال شدن داده‌های این مجموعه داده و وجود مقادیر زیاد صفر در آن، مقدار پشتیبان نسبتا پایینی بدست آمده است زیرا همانگونه که می‌دانیم این مولفه شدیدا وابسته به نوع مجموعه داده‌ها می‌باشد. در حالی که روی یکسری داده‌های دست ساز این الگوریتم قوانینی با تقریبا حداکثر مقدار پشتیبان ممکن تولید کرده است که نشانگر کارایی بسیار بالای آن در استخراج این نوع قوانین می‌باشد.

جدول ۲. نتایج حاصل از آزمایش

میانگین طول قوانین	میانگین مقدار پشتیبان	میانگین درجه اطمینان	تعداد قوانین تولید شده	حداکثر تعداد قوانین	اندازه نمونه
۱۴/۲	٪۲۳	٪۹۱	۳۹	۱۰۰	
۱۳/۶	٪۲۳	٪۹۳	۴۸	۲۰۰	۱۰۰۰
۱۱/۷	٪۲۶	٪۹۷	۵۱	۳۰۰	
۱۵/۱	٪۲۲	٪۹۴	۴۱	۱۰۰	
۱۵/۵	٪۲۴	٪۹۴	۵۲	۲۰۰	۱۰۰۰
۱۴/۶	٪۲۵	٪۹۶	۵۴	۳۰۰	
۱۲/۹	٪۲۲	٪۹۴	۴۶	۱۰۰	
۱۳/۳	٪۲۵	٪۹۵	۶۷	۲۰۰	۱۰۰۰
۱۱/۸	٪۲۷	٪۹۸	۶۸	۳۰۰	
۱۲/۳	٪۳۰	٪۹۶	۶۱	۱۰۰	
۱۱/۸	٪۳۳	٪۹۷	۷۲	۲۰۰	۲۰۰۰
۱۰/۱	٪۳۵	٪۹۹	۷۴	۳۰۰	
۱۲/۸	٪۲۹	٪۹۶	۶۵	۱۰۰	
۱۱/۹	٪۳۴	٪۹۹	۷۱	۲۰۰	۲۰۰۰
۱۱	٪۳۶	٪۹۹	۷۱	۳۰۰	

٦ نتیجه‌گیری

کشف قوانین انجمنی در یک مجموعه داده بزرگ یک عمل زمان بر و در عین حال بسیار مهم با کاربردهای متنوع و وسیع است. ما در این مقاله سعی کردیم از ویژگی‌های الگوریتم‌های تکاملی به خصوص سرعت همگرایی بالای الگوریتم PSO برای کاوش قوانین انجمنی استفاده کنیم و راهکاری را بر این اساس پیشنهاد دادیم. الگوریتم پیشنهادی را روی چند مجموعه داده آزمایش کردیم که نتایج حاصل حاکی از موفقیت خوب روش پیشنهادی در عمل بود. ما در این مقاله از درجه اطمینان، مقدار پشتیبان و طول قانون تولید شده به عنوان برازندگی الگوریتم PSO استفاده کردیم، حال آنکه استفاده از پارامترهای دیگر مانند درجه جالبی و استفاده از

Archive of SID

www.SID.ir

الگوریتم PSO از نوع پارتو می‌تواند بهتر عمل کرده و قوانین قوی‌تری تولید نماید. از سوی دیگر ما در قسمت آزمایش از نمونه‌های به تصادف انتخاب شده از مجموعه داده اصلی استفاده کردیم ولی از آنجایی که استفاده از نمونه خوب صحت قوانین تولید شده را افزایش می‌دهد، می‌توان از روش‌هایی مانند خوشه‌بندی جهت انتخاب نمونه استفاده کرد. ما این ایده ها را به عنوان موضوعی برای کارهای آتی در این زمینه پیشنهاد می‌کنیم.

منابع

[1] Tan, P. N., Steinbach, M., Kumar, V., (2005). Introduction to Data Mining.Addison Wesley.
[2] Adamo, J. M., (2001). Data Mining for Association Rules and Sequential Patterns, Springer-Verlag, New York.
[3] Agrawal, R., Srikant, R., (1994). Fast algorithms for mining association rules, in: Proceedings of the 20th International Conference on Very Large Databases, Santiago, Chile.
[4] Ayad, A. M., (2000). A new algorithm for incremental mining of constrained association rules. Master Thesis, Department of Computer Sciences and Automatic Control, Alexandria University.
[5] Hipp. J., et al., (2000). Algorithms for association rule mining-a general survey and comparison, SIGKDD Explorations 2.
[6] Agrawal, R. T., Imielinski, Swami, A., (1993). Mining association rules between sets of items in large databases," Proc. of ACM SIGMOD Conf. on Management of Data, pp.207–216.
[7] Freitas, A. A., (2001). Understanding the crucial role of attribute interaction in data mining, Artificial Intelligence Review 16 pp.177–199.
[8] Lin D. I., Kedem, Z. M., (1998). Pincer-search: an efficient algorithm for discovering the maximal frequent set, Proc. of 6th European Conf. on Extending Database Technology.
[9] Brin, S., et al., (1997). Dynamic itemset counting and implication rules for market basket data, Proc. of ACM SIGMOD International Conf. on Management of Data. Tucson, AZ.
[10] Kennedy, J., Eberhart, R. C., (1995). Particle Swarm Optimization, Proc. of IEEE In Int. Conf. on Neural Networks, pp. 1942-1948, Piscataway, NJ.

Archive of SID

www.SID.ir