



روش‌های داده‌کاوی برای یافتن داده‌های تکراری در پایگاه داده‌های بزرگ

آزاده پیرحیاتی^۱

۱ دپارتمان مهندسی کامپیوتر، دانشکده فنی دختران بروجرد، دانشگاه فنی و حرفه‌ای استان لرستان،
ایران. پست الکترونیکی: pirhayati.azadeh@gmail.com

چکیده

داده‌کاوی، پایگاه‌ها و مجموعه‌های حجیم داده را برای کشف و استخراج دانش، مورد تحلیل قرار می‌دهد. در سال‌های اخیر داده‌کاوی، با توجه به دسترسی گسترده به مقادیر بسیار زیاد داده و نیاز به تبدیل چنین داده‌هایی به اطلاعات و دانش مفید، در صنعت اطلاعات و در کل جامعه، توجه زیادی را به خود جلب کرده است. اطلاعات و دانش حاصل می‌تواند برای برنامه‌های کاربردی نظیر تجزیه و تحلیل بازار، تشخیص تقلب، حفظ مشتری، کنترل تولید و اکتشاف علمی و ... استفاده شود. یکی از مشهورترین تکنیک‌های داده‌کاوی برای تصمیم‌گیرندگان تجاری در پایگاه داده‌های بزرگ استخراج قوانین وابستگی^۱ است. کشف قوانین وابستگی در قلب داده‌کاوی قرار دارد. پیدا کردن مجموعه داده‌های تکراری، فرآیند اصلی در یافتن قانون وابستگی است.

الگوریتم‌های بسیار زیادی برای پیدا کردن الگوهای تکراری وجود دارند که در این مقاله تعدادی از آن‌ها ارائه شده است. Apriori و FP-tree متداول‌ترین روش‌ها برای یافتن آیتم‌های مکرر هستند. روش‌های دیگر از جمله Apriori TID، AIS، SETM و TR-FCTM^۲ می‌باشند. Apriori با استفاده از تولید کاندیدا با تعداد بیشتری از اسکن‌های پایگاه داده، تعداد قابل ملاحظه‌ای آیتم تکراری پیدا می‌کند. FP-tree برای پیدا کردن آیتم‌های تکراری از دو اسکن پایگاه داده بدون استفاده از تولید کاندید استفاده می‌کند. روش TR-FCTM با ایجاد یکباره‌ی کاندیدها به منظور تشکیل جدول شمارش تکرار با یک اسکن پایگاه داده، آیتم‌های تکراری را پیدا می‌کند. بررسی‌ها نشان می‌دهد که الگوریتم TR-FCTM عملکرد بهتری نسبت به Apriori و FP-tree دارد.

واژه‌های کلیدی

داده‌کاوی، قانون وابستگی، اقلام تکراری، پایگاه داده بزرگ، تولید کاندید.

۱- مقدمه

در سال‌های اخیر با بیشتر شدن کارهای کامپیوتری، پیشرفتی نمایی در تولید و کنترل اطلاعات الکترونیک به وجود آمده است. هر سازمان یا شرکتی به این نتیجه رسیده است که اطلاعات به دست آمده در طی سالیان متوالی یک مزیت استراتژیک با اهمیت است و نتایج مفید بالقوه‌ی بسیاری در این مقادیر بسیار زیاد داده وجود دارند. از این رو نیازمند تکنیک‌هایی برای استخراج اطلاعات ارزشمند از این مجموعه داده‌ها هستیم [۱].

داده‌کاوی تکنیک‌هایی را برای پیدا کردن اطلاعات پنهان مفید فراهم می‌نماید. داده‌کاوی گروهی از روش‌ها برای یافتن خودکار و کارآمد الگوهای معتبر، جدید، ارزشمند و قابل فهم در پایگاه‌های داده‌ی بزرگ است که پیش از این ناشناخته بودند [۱] و [۲]. اخیراً به خاطر کاربرد آن در چندین زمینه مانند تصمیم‌گیری، استراتژی بازار و پیش‌بینی‌های اقتصادی، بسیار مورد توجه متخصصین و محققین قرار گرفته است [۳].

¹ Association rule

² Transaction Reduction-Frequency Count Table Method



تکنیک‌های داده‌کاوی را می‌توان به صورت کلی به تکنیک‌های توصیفی^۱ و پیشگویانه^۲ تقسیم کرد. داده‌کاوی توصیفی به روش‌هایی اطلاق می‌شود که در آن‌ها مشخصات ضروری یا خواص مشترک داده‌ها در یک پایگاه داده نشان داده می‌شوند. تکنیک‌های توصیفی شامل وظایفی مانند استخراج داده‌های طبقه بندی شده، وابسته و تکراری است [۴]. روش‌های استخراج داده‌ی پیشگویانه آن‌هایی هستند که از استنتاج داده‌های ورودی برای دستیابی به دانش پنهان و پیشگویی‌های جالب و مفید استفاده می‌نمایند [۵]. تکنیک‌های پیشگویانه استخراج داده شامل وظایفی مانند دسته‌بندی، رگرسیون و انحراف معیار هستند [۴]. موضوع‌ها یا چالش‌های تحقیقاتی کلیدی در استخراج داده عبارتند از: کارآیی، روش‌شناسی استخراج داده، واکنش کاربر و تنوع داده. از این رو الگوریتم‌های استخراج داده و اسلوب‌شناسی‌ها باید کارآمد بوده و قابل تعمیم به یک پایگاه داده و زمان اجرای مربوط به آن باشند [۵].

تحقیقات بسیاری در راستای توسعه دادن روشی کارآمد برای پیدا کردن الگوهای تکراری انجام شده است. [۳]، [۶]، [۷]، [۸]، [۹] و [۱۰]. در میان این روش‌ها، Apriori [۶] و FP-tree [۷] مشهورترین روش‌ها برای پیدا کردن آیتم‌های مکرر هستند. FP-tree از استراتژی و ساختار الگوریتم متفاوت و زمان محاسباتی کمتری نسبت به روش Apriori برای پیدا کردن آیتم‌های تکراری استفاده می‌کند. روش TR-FCTM [۱۱] نیز به منظور پیدا کردن آیتم‌های مکرر با زمان محاسباتی کمتری نسبت به FP-Tree معرفی شده است. تعدادی از این الگوریتم‌ها در بخش‌های بعد معرفی شده است.

۲- استخراج قانون وابستگی

استخراج قانون وابستگی یکی از مشهورترین تکنیک‌های توصیفی استخراج داده است [۴]. از زبان ارائه شدن این مفهوم [۶]، استخراج قانون وابستگی به یکی از وظایف اصلی استخراج داده مبدل شده است و توجه قابل ملاحظه‌ای را در میان محققین و متخصصین به خود جلب کرده است [۸]. این روش، روشی مناسب برای پیدا کردن قانون وابستگی میان متغیرها در پایگاه‌های داده بزرگ است. برای مثال رابطه (۱):

$$\text{For each } x:\text{persons}, \text{ buys } (x, \text{"bread"}) \rightarrow \text{ buys } (x, \text{"butter"}) \quad (1)$$

که در آن، x یک متغیر و $\text{buys}(x, y)$ یک پیش‌بینی است که بیان می‌کند شخص x آیتم y را خریداری کرده است. این قانون مشخص می‌کند که درصد بالایی از افرادی که نان می‌خرند، کره نیز می‌خرند [۹]. بیشتر الگوریتم‌ها بر اساس الگوریتم متداول استخراج قانون وابستگی ساخته شده‌اند [۶] و [۱۰]. استخراج قانون وابستگی را می‌توان به صورت زیر تعریف نمود:

فرض کنید $I = \{i_1, i_2, \dots, i_n\}$ مجموعه‌ای از آیتم‌ها است. هر زیرمجموعه از I مجموعه آیتم نامیده می‌شود. فرض کنید که D مجموعه‌ای از تراکنش‌ها باشد. هر تراکنش T مجموعه‌ای از آیتم‌ها و زیرمجموعه‌ای از I است. هر تراکنش یک شناسه‌ی منحصر به فرد TID دارد. با استفاده از دو روش آماری، فعالیت قانون استخراج وابستگی کنترل می‌شود که عبارتند از پشتیبانی^۳ و اطمینان^۴ [۲]. در ابتدا، مجموعه داده‌های تکراری را بر اساس کمترین تعداد پشتیبان مشخص می‌کند. سپس، کمترین اطمینان برای پیدا کردن قوانین وابستگی میان آیتم‌های تکراری

¹ Descriptive
² Predictive
³ Support
⁴ Confidence



استفاده می‌شود. به‌طور کلی استخراج قوانین وابستگی شامل دو مرحله است: مرحله اول، یافتن مجموعه آیت‌های تکراری و مرحله دوم، تولید قوانین وابستگی قوی از مجموعه داده‌های تکراری [۱۲]. استخراج مجموعه‌های تکراری مرحله‌ی اصلی در استخراج وابستگی‌ها است که مجموعه داده‌های تکراری را در تراکنش‌های پایگاه داده مشخص می‌کند. این یک مرحله‌ی ضروری در بسیاری از روش‌های داده‌کاوی است. الگوریتم‌های بسیاری به منظور شناسایی مجموعه داده‌های تکراری پیشنهاد شده‌اند.

۳- الگوریتم AIS

الگوریتم AIS چند گذر در کل پایگاه داده ایجاد می‌کند. در طول هر گذر، تمام تراکنش‌ها را اسکن می‌کند. در گذر اول، مجموعه آیت‌های مجزا را شمارش می‌کند و تعیین می‌کند کدام یک از آن‌ها در پایگاه داده، بزرگ یا تکراری هستند. مجموعه آیت‌های بزرگ از هر گذر برای تولید مجموعه آیت‌های کاندید گسترش می‌یابند. پس از اسکن یک تراکنش، آیت‌های مشترک بین آیت‌های بزرگ گذر قبلی و آیت‌های این تراکنش تعیین می‌شوند. الگوریتم AIS اولین الگوریتم منتشر شده برای تولید تمام مجموعه آیت‌های بزرگ در تراکنش‌های یک پایگاه داده بود [۱۰]. تمرکز آن بر بهبود پایگاه‌های داده با قابلیت‌های لازم برای پردازش و پشتیبانی درخواست‌ها بود. هدف این الگوریتم، کشف قوانین کیفی است و در یافتن ارتباط بین بخش‌ها در رفتار خرید مشتری، استفاده می‌شود. این تکنیک، در نتیجه، تنها به یک مورد محدود می‌شود. مشکل اصلی الگوریتم AIS این است که کاندیدهای زیادی را تولید می‌کند که بعداً کوچک می‌شوند. یکی دیگر از مشکلات این الگوریتم، این است که در آن ساختارهای داده مورد نیاز برای حفظ مجموعه آیت‌های بزرگ و کاندید مشخص نشده است [۱۲].

۴- الگوریتم SETM

الگوریتم SETM مشابه الگوریتم AIS، گذرهای چندگانه را بر روی پایگاه داده انجام می‌دهد. در گذر اول، مجموعه آیت‌های مجزا را شمارش می‌کند و تعیین می‌کند کدام یک از آن‌ها در پایگاه داده، بزرگ یا تکراری هستند. سپس با گسترش مجموعه آیت‌های بزرگ گذرهای قبلی، مجموعه آیت‌های کاندید را تولید می‌کند. علاوه بر این، این الگوریتم، TIDهای تراکنش‌های تولیدی را با مجموعه آیت‌های کاندید به یاد می‌آورد. عملیات رابطه‌ای ادغام و پیوند جدوال می‌تواند برای تولید اقلام کاندید استفاده شود.

مزیت آن این است که با تولید مجموعه‌های کاندید، الگوریتم SETM کپی اقلام کاندید همراه با TID تراکنش تولیدی را به صورت پیوسته ذخیره می‌کند.

مشکل الگوریتم این است که چون برای هر مجموعه کاندید، یک TID مرتبط با آن وجود دارد، برای ذخیره تعداد زیادی از TIDها نیاز به فضای بیشتری است. SETM کارآمد نیست و هیچ نتیجه‌ای در مورد اجرای آن در برابر یک DBMS^۱ رابطه‌ای گزارش نشده است [۱۲].

۵- الگوریتم Apriori

این الگوریتم معروف‌ترین الگوریتم قانون وابستگی است. تفاوت‌های اساسی این الگوریتم با الگوریتم‌های AIS و SETM، روش‌های تولید و انتخاب مجموعه‌های کاندید برای شمارش است. Apriori مجموعه‌های کاندید را با پیوستن به مجموعه‌های بزرگ از گذر قبلی و حذف زیر مجموعه‌های آن، که در گذر قبلی کوچک شده‌اند، بدون در نظر گرفتن تراکنش‌ها در پایگاه داده، تولید می‌کند. تنها با توجه به مقادیر بزرگ موجود در گذر قبلی، تعداد کاندیدهای مجموعه‌های بزرگ به‌طور قابل توجهی کاهش می‌یابد [۱۲]. نتایج تجربی نشان می‌دهد که الگوریتم Apriori همیشه بهتر از AIS و SETM است [۱۰].

اگرچه پس از معرفی Apriori تعدادی الگوریتم جایگزین معرفی شدند که اغلب آن‌ها دارای ساختار کلی الگوریتم Apriori بودند ولی، تنها یک یا چند مرحله از آن را بهینه‌سازی می‌کردند. عیب الگوریتم Apriori این

^۱ Data Base Management System



است که به منظور پیدا کردن مجموعه داده‌های تکراری نیازمند تعداد اسکن‌های زیادی از پایگاه داده است و اگر اندازه‌ی پایگاه داده افزایش یابد زمان بیشتری لازم خواهد داشت [۱۳].

۶- الگوریتم Apriori TID

در این الگوریتم برای شمارش مجموعه آیت‌های کاندید، پس از اولین گذر، از پایگاه داده استفاده نمی‌شود. فرآیند تولید آیت‌های کاندید مشابه الگوریتم Apriori است. مجموعه دیگری مثل C تولید می‌شود که در آن هر عضو TID هر تراکنش و مجموعه آیت‌های بزرگ موجود در آن تراکنش را دارد. مجموعه تولید شده C برای شمارش هر مجموعه آیت کاندید استفاده می‌شود. مزیت این الگوریتم این است که در گذرهای بیشتر عملکرد Apriori TID بهتر از Apriori است و Apriori تنها در گذرهای اولیه بهتر از Apriori TID عمل می‌کند [۱۴].

۷- الگوریتم FP-tree

FP-tree در سال ۲۰۰۰ توسط هان و همکارانش ارائه شد که از روش رشد الگو و یک ساختار داده‌ی خاص استفاده می‌کرد. این الگوریتم از طریق ایجاد ساختار FP-tree برای یک پایگاه داده‌ی بزرگ، تعداد اسکن‌های پایگاه داده را کاهش می‌دهد. افزایش اندازه‌ی FP-tree با توجه به رشد پایگاه داده منجر به دشوار شدن ایجاد، جستجو و عملیات درج در یک FP-tree بزرگ می‌شود [۱۱].

این الگوریتم برای پیدا کردن آیت‌های تکراری از دو اسکن پایگاه داده بدون استفاده از تولید کاندید استفاده می‌کند. برای اولین بار و در اولین اسکن پایگاه داده خود، لیستی از آیت‌های تکراری را محاسبه نموده و آن‌ها را به ترتیب نزولی بر اساس فرکانس مرتب می‌کند. در اسکن دوم، پایگاه داده به یک FP-tree فشرده می‌شود. این الگوریتم استخراج را روی FP-tree به‌طور بازگشتی انجام می‌دهد. یک مشکل این الگوریتم، پیدا کردن مجموعه آیت‌های تکراری است که به جستجو و ساختن درختان بازگشتی تبدیل شده است [۱۴]. با وجود اینکه قدرت محاسباتی در طی سال‌های اخیر به میزان بسیار زیادی افزایش یافته است، الگوریتم‌های کارآمد با ساختارهای داده‌ی سفارشی هنوز برای به دست آوردن نتایج در زمان مناسب ضروری هستند. این امر خصوصاً در استخراج داده‌ی مهم است زیرا داده‌ی کلوی روشی بسیار زمان‌بر است [۷].

۸- الگوریتم TR-FCTM

این الگوریتم از ادغام تراکنش‌ها و جدول شمارش تکرار، برای پیدا کردن آیت‌هایی که تکرار قابل ملاحظه‌ای دارند استفاده می‌کند تا بتواند زمان محاسبات را نسبت به الگوریتم FP-tree کاهش دهد. در ابتدا تراکنش‌های مشابه در پایگاه داده یکی شده و تراکنش‌های یکپارچه شده در حافظه‌ی اصلی ذخیره می‌شوند. سپس، تراکنش‌های ادغام شده یکی یکی از حافظه‌ی اصلی خوانده شده و جدول شمارش تکرار به‌روز رسانی می‌شود [۱۱] به-طور کلی در این الگوریتم دو عمل زیر انجام می‌شود:

۸-۱- کاهش تعداد اسکن‌های پایگاه داده با استفاده از ادغام تراکنش‌ها

این عمل بر این اساس است که تراکنش پایگاه داده‌ها مکرراً در بردارنده‌ی تراکنش‌هایی مشابه هستند. این تکنیک شامل پیدا کردن این تراکنش‌ها و جایگزین کردن آن‌ها با یک تراکنش است. این عمل را می‌توان به صورت ریاضی اینگونه بیان کرد که: اگر یک مجموعه از تراکنش‌های یکسان $(Tx_1, Tx_2, \dots, Tx_m)$ در یک پایگاه داده D باشند آنگاه با یک تراکنش تنها به صورت (TM, x) جایگزین می‌شوند که TM مجموعه داده‌ی یکسان است و m تعداد کل مجموعه داده‌ی تکراری در پایگاه داده است [۱۵]. این امر باعث کاهش تعداد کل تراکنش‌ها در پایگاه داده می‌شود. از این رو این عمل به میزان قابل توجهی باعث کاهش یافتن زمان محاسبه در عملیات یافتن مجموعه داده‌های تکراری می‌شود.

۸-۲- کاهش تراکنش‌ها (FCTM)

داده D باشد. این بیان می‌کند که یک مجموعه داده‌ی X



از تراکنش T زیرمجموعه‌ای از I است و مجموعه‌ای از چنین تراکنش‌هایی، پایگاه داده‌ی D را تشکیل می‌دهند. از این رو در این الگوریتم از یک جدول به نام جدول شمارش تکرار استفاده می‌شود. این جدول دارای دو فیلد به نام‌های مجموعه داده و مقدار شمارش تکرار است. این جدول ورودی‌هایی از تعداد تکرار هر مجموعه داده‌ای که در تراکنش‌های پایگاه داده یافت شود ایجاد می‌کند. تعداد تکرار هر مجموعه داده برابر تعداد حضور آن در تراکنش پایگاه داده‌ی D است. این جدول تشکیل شده و می‌تواند تا زمانی که مجموعه داده‌ی تکراری دیگری یافت نشده، در حافظه نگهداری شود [۱۶].

در این الگوریتم برای یافتن مجموعه داده‌ی تکراری برای هر مقدار آستانه، الگوریتم به جای پایگاه داده، FCT را اسکن می‌کند. FCT دارای ورودی‌هایی از تعداد تکرار تمامی مجموعه داده‌ها است، نه تعداد کل پشتیبان آن مجموعه داده. تعداد تکرار هر مجموعه داده برابر با تعداد حضور آن مجموعه داده در تراکنش پایگاه داده D است. تعداد کل مجموعه داده‌ی به خصوص X، با بررسی اینکه آیا زیرمجموعه‌ی یک مجموعه داده‌ی بزرگ‌تر در FCT هست یا نه محاسبه می‌شود. اگر تعداد تکرار کل مجموعه داده‌ی خاص مساوی یا بزرگ‌تر از آستانه موردنظر باشد آنگاه آن مجموعه داده در مجموعه داده‌های مکرر قرار می‌گیرد [۱۱].

نتایج آزمایش زمان اجرا نشان می‌دهد که TR-FCTM زمان کمتری را برای اسکن کردن تراکنش‌ها و تولید کاندید نسبت به Apriori و FP-tree صرف می‌کند [۱۱] و [۲۰].

۹- نتیجه‌گیری و جمع‌بندی

داده‌کاوی به استخراج اطلاعات نهان یا الگوها و روابط مشخص در حجم زیادی از داده‌ها در پایگاه داده‌های بزرگ گفته می‌شود. الگوریتم‌های یافتن الگوهای تکراری به منظور پیدا کردن الگوهای حالت توجه در پایگاه‌های داده‌ی بزرگ بسیار مهم هستند. استخراج قوانین وابستگی یکی از مشهورترین تکنیک‌های داده‌کاوی در پایگاه داده‌های بزرگ است. پیدا کردن مجموعه داده‌های تکراری، فرایند اصلی در یافتن قانون وابستگی است.

الگوریتم‌های بسیار زیادی برای پیدا کردن الگوهای تکراری و قوانین وابستگی وجود دارند که در این مقاله تعدادی از آن‌ها از جمله Apriori، FP-tree، Apriori TID، TR-FCTM، SETM و AIS ارائه شدند. هر الگوریتم مزایا و معایب خودش را دارد. اولین الگوریتم منتشر شده برای تولید تمام مجموعه آیت‌های بزرگ در تراکنش‌های یک پایگاه داده الگوریتم AIS بود. الگوریتم SETM مشابه الگوریتم AIS، چندین گذر روی پایگاه داده انجام می‌دهد. Apriori با استفاده از تولید کاندید با تعداد زیادی اسکن روی پایگاه داده، آیت‌های تکراری زیادی پیدا می‌کند. FP-tree برای پیدا کردن آیت‌های تکراری از دو اسکن پایگاه داده استفاده می‌کند. الگوریتم TR-FCTM با استفاده از ادغام تراکنش‌ها، یافتن تعداد تکرارها و شمارش کل تکرارها برای یک مجموعه داده طراحی شده است. بررسی‌ها نشان می‌دهد که الگوریتم TR-FCTM عملکرد بهتری نسبت به الگوریتم‌های Apriori و FP-tree دارد.

امروزه استخراج قوانین وابستگی بین آیت‌ها در تراکنش‌های فروش پایگاه داده‌های بزرگ به عنوان یک موضوع مهم تحقیقاتی در پایگاه داده‌ها شناخته شده است.

مراجع

- [۱] G.K. Gupta, ۲۰۰۶. "Introduction to Data Mining with Case Studies". Prentice-Hall of India Pvt.
- [۲] Saravanan Suba and Christopher T, ۲۰۱۲. "A Study on Milestones of Association Rule Mining Algorithms in Large Databases". *International Journal of Computer Applications*, Vol. ۴۷, No. ۳, pp. ۱۲-۱۹.
- [۳] Ya-Han Hu and Yen-Liang Chen, ۲۰۰۶. "Mining Association Rules with Multiple Minimum Supports: A New Mining Algorithm and A Support Tuning Mechanism". *Decision Support Systems*, Vol. ۴۲, No. ۱, pp. ۱-۲۴.
- [۴] S. Shankar and T. Purusothaman, ۲۰۰۹. "Utility Se



- Mining: A Literature Survey and Comparative Study". *International Journal of Soft Computing Applications*, No. ۴, pp. ۸۱-۹۵.
- [۵] N.P. Gopalan and B. Sivaselvan, ۲۰۰۹. "Data Mining Techniques and Trends". PHI Learning.
- [۶] R. Agrawal, T. Imielinski and A. Swami, ۱۹۹۳. "Mining Association Rules Between Sets of Items in Large Databases". *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. ۲۰۷-۲۱۶.
- [۷] Jiawei Han, Jian Pei and Yiwen Yin, ۲۰۰۰. "Mining Frequent Patterns without Candidate Generation". *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. ۱-۱۲.
- [۸] M.J. Zaki and C.J. Hsiao, ۱۹۹۹. "CHARM: An Efficient Algorithm for Closed Association Rule Mining". *Department of Computer Science, Rensselaer Polytechnic Institute, Technical Report ۹۹-۱۰*.
- [۹] Yin-Ling Cheung and A.W.-C. Fu, ۲۰۰۴. "Mining Frequent Itemsets without Support Threshold: with and without Item Constraints". *IEEE Transactions on Knowledge and Data Engineering*, Vol. ۱۶, No. ۹, pp. ۱۰۵۲-۱۰۶۹.
- [۱۰] R. Agrawal and R. Srikant, ۱۹۹۴. "Fast Algorithms for Mining Association Rules". *Proceedings of ۲۰th International Conference on Very Large Data Bases*, pp. ۴۸۷-۴۹۹.
- [۱۱] Saravanan Suba and T. Christopher, 2112. "AN EFFICIENT DATA MINING METHOD TO FIND FREQUENT ITEM SETS IN LARGE DATABASE USING TR- FCTM". *ICTACT JOURNAL ON SOFT COMPUTING*, JANUARY, VOL. 2, ISSUE. 2.
- [۱۲] Himani Bathla, Ms. Kavita Kathuria, ۲۰۱۵. "Association Rule Mining: Algorithms Used". *International Journal of Computer Science and Mobile Computing (IJCSMC)*, June, Vol. ۴, Issue. ۶, pp. ۲۷۱-۲۷۷.
- [۱۳] S. Sunil Kumar, S. Shyam Karanth, K.C. Akshay, Ananth Prabhu and M. Bharathraj Kumar, ۲۰۱۲. "Improved Apriori Algorithm based on Bottom Up Approach using Probability and Matrix". *International Journal of Computer Science Issues*, Vol. ۹, No. ۲, pp. ۲۳۲-۲۴۶.
- [۱۴] Trupti A. Kumbhar and Prof. Santosh V. Chobe, 2114. "An Overview of Association Rule Mining Algorithms". *(IJCSIT) International Journal of Computer Science and Information Technologies*, Vol. 5 (1), pp. 729-731.
- [۱۵] Z. Souleymane, P.F. Viger, J.C.-W. Lin, C.-W. Wu and V.S. Tseng, ۲۰۱۵. "EFIM: A Highly Efficient Algorithm for High-Utility Itemset Mining". *Proceedings of 14th Mexican International Conference on Artificial Intelligence*, pp. ۵۳۰-۵۴۶.
- [۱۶] R. Ahirwal, N.K. Kori and Y.K. Jain, ۲۰۱۲. "Improved Data Mining Approach to Find Frequent Itemset using Support Count Table". *International Journal of Emerging Trends & Technology in Computer Science*, Vol. ۱, No. ۲, pp. ۱۹۵-۲۰۱.