

رویکردی جدید جهت کاوش الگوهای مکرر فازی با آیت‌های چندگانه و چندین حداقل آستانه پشتیبانی

منیژه نیکو فراز^۱، الهام قنبری*^۲، فاطمه نصیری^۳

۱- گروه کامپیوتر، واحد یادگار امام خمینی (ره) شهری، دانشگاه آزاد اسلامی، تهران،

ایران

nikoofaraz.m@gmail.com

۲- عضو هیات علمی گروه کامپیوتر، واحد یادگار امام خمینی (ره) شهری، دانشگاه آزاد

اسلامی، تهران، ایران

el.ghanbari@iausr.ac.ir

۳- عضو هیات علمی گروه کامپیوتر، واحد یادگار امام خمینی (ره) شهری، دانشگاه آزاد

اسلامی، تهران، ایران

fnasiri@iausr.ac.ir

چکیده

کاوش قوانین انجمنی یکی از موضوعات مهم جهت استخراج داده‌ها است تا بتوان از روی داده‌های استخراج شده، اطلاعات و روابط بالقوه بین آن‌ها را کشف کرد. استفاده از یک مقدار حداقل آستانه پشتیبانی واحد برای سنجش مطلوب بودن روابط بین آیت‌ها، سبب می‌شود تا آیت‌ها با تعداد تکرار کمتر ولی با درجه اهمیت بالا در قوانین انجمنی استخراج شده، یافت نشوند. لذا در این مقاله برای حل این مشکل، الگوریتمی پیشنهاد می‌شود که برای هر آیت‌م موجود در مجموعه داده، با توجه به درجه اهمیتی که دارد، یک مقدار حداقل آستانه پشتیبانی مجزا در نظر می‌گیرد. در ادامه در الگوریتم پیشنهادی برای حل مشکل لبه‌های تیز که از بخش بندی داده‌های کمی به منظور استخراج قوانین انجمنی به وجود می‌آید از منطق فازی استفاده می‌شود. بدین منظور روش پیشنهادی به کاوش قوانین انجمنی فازی و کشف روابط بین آیت‌ها با استفاده از مجموعه‌های فازی و عبارات زبان شناختی می‌پردازد. در روش پیشنهادی جهت استخراج تمامی الگوهای مکرر موجود، تمامی آیت‌های فازی که شرط مکرر بودن را دارند به‌جای فقط آیت‌هایی که دارای بالاترین مقدار پشتیبان هستند، در عملیات کاوش مشارکت داده می‌شوند و از ساختار درخت شماری و کوچک‌ترین آستانه پشتیبانی برای حذف داده‌های غیر مکرر استفاده خواهد شد. استفاده از ساختارهای معرفی شده موجب کاهش زمان اجرای الگوریتم و مقیاس‌پذیری آن خواهد شد.

کلمات کلیدی: قوانین انجمنی، حداقل آستانه پشتیبانی، لیست فازی، الگوهای مکرر فازی، کوچک‌ترین آستانه پشتیبانی

۱- مقدمه

استخراج قوانین انجمنی یکی از روش‌های داده‌کاوی است که با کمک الگوهای به فرم " $X \rightarrow Y$ " (X, Y مجموعه آیت‌ها هستند) نمایش داده می‌شوند [۱][۲]. تاکنون الگوریتم‌های زیادی برای استخراج قوانین انجمنی، ارائه شده است که هر یک دارای مزایا و محدودیت‌هایی هستند [۳][۴]. همواره سعی بر آن بوده تا الگوریتم‌هایی ارائه شوند که تعداد پیمایش پایگاه داده جهت کاوش قوانین را کاهش داده و قوانینی با دقت و صحت بالا در کمترین زمان ممکن استخراج نمایند و بر مسائلی مانند حجم بالای داده‌ها، محدودیت حافظه و تنوع داده‌های موجود بر روی انواع پایگاه داده فائق آیند [۵]. از آنجا که هنوز مشکلات بسیاری در این حوزه از دانش وجود دارد در این مقاله سعی بر آن است تا الگوریتمی ارائه شود که بر مشکلات موجود فائق

آید و الگوریتم‌های پیشین را بهبود بخشد. الگوریتم‌های بسیاری وجود دارند که از داده‌هایی که فقط شامل مقادیر ۰ و ۱ هستند، جهت استخراج مجموعه آیت‌های مکرر و کاوش قوانین انجمنی استفاده می‌کنند در حالی که داده‌های موجود در واقعیت شامل داده‌های کمی نیز هستند. برای استخراج قوانین انجمنی از مجموعه داده‌های کمی لازم است تا مجموعه داده‌ها به داده‌هایی از نوع دودویی تبدیل شوند که این امر مسئله لبه‌های حساس و مرزهای تیز را به وجود می‌آورد. لذا در راهکار پیشنهادی از مجموعه‌های فازی استفاده می‌شود تا مقادیر عددی نسبت به روش‌های سنتی بهتر به کار گرفته شوند؛ زیرا مجموعه‌های فازی، اثر مرزهای تیز را نرم می‌کنند. در مجموعه‌های فازی با استفاده از توابع عضویت می‌توان دقیقاً مشخص کرد که تا چه مقدار، یک آیت به یک مجموعه فازی تعلق دارد. مجموعه‌های فازی که به منظور بیان مفاهیم با عبارات زبان‌شناختی بیان می‌شوند [۶] می‌توانند با دادن مقادیر عضویت به مجموعه‌های مختلف بر مشکل مذکور غلبه کنند. قوانین انجمنی فازی نسخه‌های خوش‌ساخت قوانین انجمنی کمی هستند که در آن از مجموعه‌های فازی استفاده می‌شود [۷]. از آنجاکه در قوانین انجمنی فازی، آیت‌ها با عبارات زبان‌شناختی که برای انسان بسیار مفهوم‌تر است بیان می‌شوند، در روش پیشنهادی برای کاوش تمامی الگوهای مکرر، همه‌ی عبارات زبان‌شناختی یک آیت فازی که دارای مقدار پشتیبان بیشتر از حداقل آستانه پشتیبانی از پیش تعریف‌شده باشند در ساخت الگوهای مکرر استفاده خواهند شد.

در بسیاری از روش‌های سنتی مانند [۸] [۹] فقط یک عبارت زبان‌شناختی از یک آیت که دارای بیشترین مقدار پشتیبان است به کار گرفته می‌شوند که این موجب می‌شود تعداد بسیار زیادی از مجموعه آیت‌هایی که قابلیت مکرر بودن را دارند حذف شوند و در الگوهای کاوش شده حضور نداشته باشند. در الگوریتم‌های کاوش قوانین انجمنی سنتی از یک مقدار حداقل آستانه پشتیبانی واحد به عنوان معیاری جهت بررسی مکرر بودن یک مجموعه آیت استفاده می‌شود که این مقدار توسط کاربر تعیین می‌شود. انتخاب یک مقدار برای تمامی آیت‌های مجموعه داده برای ارزیابی مکرر بودن آن‌ها امری غیر معقولانه است. در روش‌های سنتی مانند [۱۰] [۱۱] تمامی آیت‌ها به یک شکل دیده می‌شوند و میزان اهمیت آن‌ها که می‌تواند متأثر از عواملی همچون قیمت و یا سود باشد در نظر گرفته نمی‌شود. در روش‌های مذکور چنانچه مقدار حداقل آستانه پشتیبانی عددی کوچک در نظر گرفته شود مسئله انفجار الگوهای کاوش شده وجود خواهد داشت و با مقادیر بزرگ حداقل آستانه پشتیبانی، بسیاری از آیت‌های بااهمیت کاوش نخواهند شد و مسئله آیت‌های نادر و کمیاب رخ خواهد داد. جهت حل این مشکل در الگوریتم پیشنهادی برای هر آیت از یک مقدار حداقل آستانه پشتیبانی مجزا استفاده می‌شود و از یک رابطه جهت محاسبه این مقادیر استفاده می‌شود. در ادامه، در بخش ۲ کارهای پیشین موردبررسی قرار می‌گیرد. در بخش ۳ و ۴ الگوریتم پیشنهادی معرفی و ارزیابی می‌شود و در انتها در بخش ۵ نتیجه‌گیری و کارهای آینده بیان می‌شود.

۲- کارهای پیشین

الگوریتم‌هایی که جهت کاوش قوانین انجمنی تاکنون ارائه شده‌اند را می‌توان به دودسته تقسیم‌بندی کرد: الگوریتم‌های مبتنی بر Apriori [۱۲] و الگوریتم‌های مبتنی بر درخت. در میان الگوریتم‌هایی که از چندین آستانه پشتیبانی استفاده می‌کنند، برای نخستین بار الگوریتم MSApriori [۱۳]، اولین ساختار را برای حل مسئله الگوهای مکرر با چندین آستانه پشتیبانی معرفی کرد، به این صورت که هر آیت دارای یک آستانه پشتیبانی واحد و هر گویی دارای حداقل آستانه پشتیبانی است و مجموعه آیت‌های نادر بدون نیاز به تولید تعداد زیادی قوانین بدون معنا، می‌توانند کاوش شوند. از دیگر الگوریتم‌ها MMSCumulate و MMS_Stratify هستند که به کاوش قوانین انجمنی به صورت طبقه‌بندی می‌پردازند و از هر فرم حداقل آستانه پشتیبانی چندگانه که توسط کاربر مشخص می‌شود، پشتیبانی می‌کنند [۱۴]. از دیگر الگوریتم‌ها یک مدل چند سطحی قوانین انجمنی تحت چندین آستانه پشتیبانی [۱۵] و الگوریتم FQSP-MMS جهت کاوش الگوهای ترتیبی کمی فازی با چندین آستانه پشتیبانی [۱۶] هستند.

در میان الگوریتم‌هایی که بر پایه درخت هستند می‌توان الگوریتم بهبودیافته CFP-Growth [۱۷] را نام برد که الگوهای مکرر با چندین آستانه پشتیبانی را با استفاده از روش رشد الگو و بر اساس یک ساختار جدید MIS-Tree کاوش

می کند. الگوریتم مذکور برای تولید تمامی الگوهای مکرر، به صورت بازگشتی یک سری درخت تولید می کند. پس از آن، الگوریتم پیشرفته ++CFP-growth [۱۸] ارائه شد که از کوچک ترین آستانه پشتیبانی به جای حداقل آستانه پشتیبانی و سه استراتژی بهبود یافته برای کاهش فضای جستجو استفاده می کند. تمامی الگوریتم های فوق که بر پایه درخت طراحی شده اند همگی فقط از داده های دودویی پشتیبانی می کنند. لذا در این مقاله رویکردی ارائه می شود که بتواند داده های کمی را پشتیبانی کند و به کاوش مجموعه بسیار کاملی از الگوهای مکرر بپردازد. در انتها می توان الگوریتم MFFI-Miner [۱۹] را نام برد که برای کاوش الگوهای مکرر فازی با استفاده از حداقل آستانه پشتیبانی ارائه شده است. این رویکرد جزء بهترین رویکردهایی است که توانسته است با استفاده از مفهوم لیست فازی نیاز به پیمایش چندباره پایگاه داده را کاهش دهد، با استفاده از استراتژی های هرس حجم فضای نمونه را کاهش داده و بدون نیاز به تولید مجموعه آیت های کاندید، موجب بهبود زمان اجرای الگوریتم و مصرف حافظه شود.

۳- الگوریتم پیشنهادی

در الگوریتم پیشنهادی با هدف استخراج قوانین انجمنی از مجموعه داده های کمی لازم است تا در ابتدا داده ها پیش پردازش شوند که شامل فازی سازی داده های تراکنشی است. خروجی عملیات پیش پردازش، مجموعه ای از آیت های فازی با عبارات زبان شناختی به همراه مقادیر عضویشان هستند. برای تعریف مجموعه آیت های فازی لازم است تا داده های کمی با استفاده از توابع عضویت از پیش تعریف شده به مقدار فازی نگاشت پیدا کنند. در ادامه مراحل الگوریتم پیشنهادی، که شبه کد آن در شکل (۱) نمایش داده شده است، به طور دقیق بیان می شود.

به این منظور در ابتدا برای کاهش تعداد دفعات پیمایش پایگاه داده، برای هر آیت فازی یک لیست فازی ساخته و در مجموعه لیست های فازی چندگانه MFLS ذخیره می شود (خط ۱). لیست های فازی که تشکیل دهنده گره های درخت شمارشی می باشند شامل سه فیلد هستند: نام (مجموعه آیت فازی)، مجموعه ای از شماره تراکنش ها و میزان عضویت هر مجموعه آیت فازی در هر تراکنش است. جهت ترکیب لیست های فازی $I_x.FL$ و $I_y.FL$ و ساخت لیست فازی $I_{xy}.FL$ با مجموعه آیت های بیشتر، در ابتدا به ازای تمامی رکوردهای موجود در دو لیست فازی $(E_y \in I_y.FL \text{ and } E_x \in I_x.FL)$ شرط وجود شماره تراکنش های یکسان بررسی می شود $(E_x.tid == E_y.tid)$. در صورت وجود شماره تراکنش های یکسان، لیست فازی $I_{xy}.FL$ طبق رابطه ی (۱) تکمیل می گردد.

$$\begin{aligned} E_{xy}.tid &\leftarrow E_x.tid \\ E_{xy}.value &\leftarrow \min(E_x.value, E_y.value) \\ E_{xy} &\leftarrow \langle E_{xy}.tid, E_{xy}.value \rangle \end{aligned} \quad (1)$$

در انتها مقدار رکورد E_{xy} به لیست فازی $I_{xy}.FL$ اضافه می شود. بعد از ساخت لیست های فازی، زمان حذف اطلاعات غیر مفید بر اساس کوچک ترین مقدار آستانه پشتیبانی آیت ها است که طبق رابطه ی (۲) محاسبه می شود (خط ۲).

$$LMS = \min\{ms(i1), ms(i2), \dots, ms(ij), \dots, ms(im)\} \quad (2)$$

در این رابطه $ms(ij)$ مقدار آستانه پشتیبانی هر آیت ij موجود در پایگاه داده تراکنشی D است. ویژگی کوچک ترین آستانه پشتیبانی به عنوان یک عامل برای کاهش فضای جستجو استفاده می شود. از این ویژگی برای حذف کردن داده هایی استفاده می شود که به صورت بالقوه مکرر نمی باشند. اگر یک گره درخت مانند گره R_{ij} دارای مقدار پشتیبان کمتر از کوچک ترین آستانه پشتیبانی باشد، آنگاه هر گره ای که حاوی R_{ij} است یا به عبارتی تمام زیرمجموعه های گره R_{ij} مستقیماً از روی درخت فضای نمونه ای حذف می شوند تا دیگر کاوش نشوند و در صورتی که دارای مقدار پشتیبان بیشتر از کوچک ترین مقدار آستانه پشتیبانی باشد آنگاه در مجموعه آیت های اول سطح FirstLevel-MFFI ذخیره می شوند (خط ۶). مقدار پشتیبان مجموعه آیت فازی R_{ij} در پایگاه داده D' برابر با مجموع مقادیر فازی است که در رابطه ی (۳) نشان داده شده است.

$$\text{Supp}(R_{il}) = \sum_{R_{il} \in T_q \wedge T_q \in D'} \text{value}(R_{il}, T_q) \quad (3)$$

در این رابطه، D' پایگاه داده فازی و T_q شماره تراکنش است. در ادامه طبق ویژگی درخت شمارشی، آیتمها بر اساس مقدار آستانه پشتیبانی شان به صورت صعودی مرتب می شوند (خط ۹).

Input: A Fuzzy database D' with the transactions T ; A set of Minimum Support threshold values MSs , each MS value is assigned to a single item.

Output: The complete set of Multiple Fuzzy Frequent Itemsets (MFFIs) with MMISs.

1. Scan the D' to construct MFLs←The fuzzy-list of each fuzzy linguistic term contains: Name of fuzzy-list(R_{il}), Transaction Numbers($tids$), fuzzy values;
2. $LMS = \text{Find_Min}(MSs)$; // calculate the Least Minimum Support
3. **for each** fuzzy-list R_{il} in MFLs **do**
4. $\text{sup}(R_{il}) = \sum_{R_{il} \in T_q \wedge T_q \in D'} f_{v_{iql}}$
5. **if** $\text{sup}(R_{il}) \geq LMS$ **then**
6. FirstLevel-MFFI ← FirstLevel-MFFI $\cup R_{il}$
7. **end if**
8. **end for**
9. Sort FirstLevel-MFFI in MS ASC. Order;
10. childrenOfRa ← FirstLevel-MFFI;
11. Ra ← \emptyset ;
12. **while** childrenOfRa is not Null
13. **for each** fuzzy-list Ra ∈ childrenOfRa **do**
14. calculate the $\text{Supp}(Ra)$ and $\text{MIS}(Ra)$ from the built structure of Ra;
15. **If** $\text{Supp}(Ra) \geq \text{MIS}(Ra) * |D'|$ **then**
16. MFFIs ← MFFIs \cup Ra; // update the final derived MFFIs
17. childrenOfRa ← \emptyset ; // initialize the set of all extensions of Ra
18. **for each** fuzzy-list Rb ∈ childrenOfR **do** // initialize the set of all extensions of Ra such b after
19. Rab ← Ra \cup Rb; // obtain the extension of Ra
20. Fuzzy-List-Construction(Ra, Rb);
21. childrenOfRa ← childrenOfRa \cup Rb // update the derived set
22. **end for**
23. Ra ← Rab;
24. **end if**
25. Go to Line 13
26. **end for**
27. **end while**
28. return MFFIs with MMISs

شکل (۲): شبه کد الگوریتم پیشنهادی

سپس هر شاخه از درخت به صورت بازگشتی پیمایش می شود. در این پیمایش، آیتمهای فازی که مقدار پشتیبان آنها کمتر از حداقل آستانه پشتیبانی شان باشد حذف می شوند و الگوهای مکرر از روی گره های درخت شمارشی که حذف نشده اند، استخراج می شوند (خطوط ۱۲-۲۷). حداقل آستانه پشتیبانی یک مجموعه فازی K تایی $X = \{i_1, i_2, \dots, i_k\}$ در پایگاه داده فازی D' به صورت $\text{MIS}(X)$ نشان داده می شود و برابر با کوچک ترین مقدار m_S آیتمها است. برای محاسبه حداقل آستانه مجموعه آیتمهای چندگانه از رابطه ی (۴) استفاده می شود که در آن مقدار آستانه پشتیبانی هر آیتم i_j موجود در پایگاه داده تراکنشی D که به صورت $ms(i_j)$ نشان داده می شود مطابق با رابطه ی (۵) محاسبه می شود.

$$\text{MIS}(X) = \min\{ms(i_1), ms(i_2), \dots, ms(i_k)\} \quad (4)$$

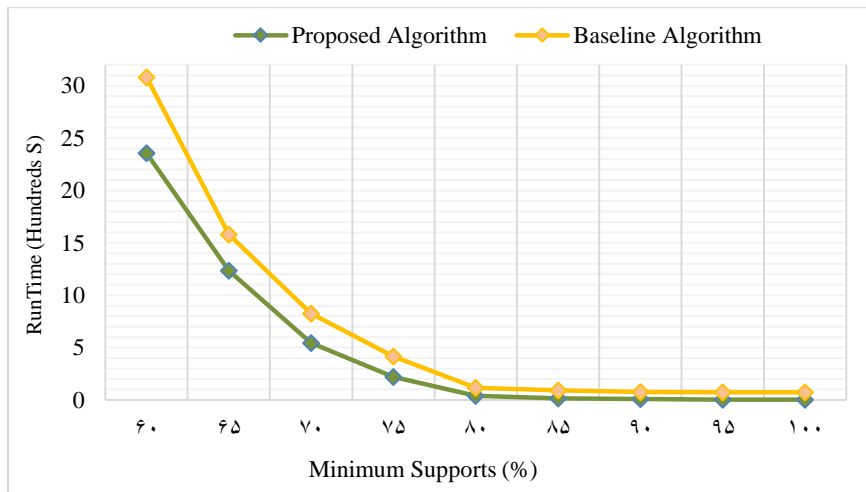
$$ms(i_j) = \begin{cases} LS, & m(i_j) \leq LS \\ m(i_j), & T > m(i_j) > LS, \\ T, & m(i_j) \geq T \end{cases} \quad m(i_j) = \beta f(i_j) \quad (5)$$

سومین همایش ملی مهندسی کامپیوتر، داده کاوی و داده های حجیم

در این رابطه $f(i_j)$ تابعی از تعداد دفعات وقوع آیت i_j در تراکنشها است. T تعداد تراکنشها و LS کوچکترین آستانه پشتیبانی است که توسط کاربر تعیین می شود. β نیز پارامتری در بازه (۱ و ۰) است که رابطه بین مقدار $m(i_j)$ و تعداد وقوع i_j را کنترل می کند.

۴- ارزیابی الگوریتم پیشنهادی

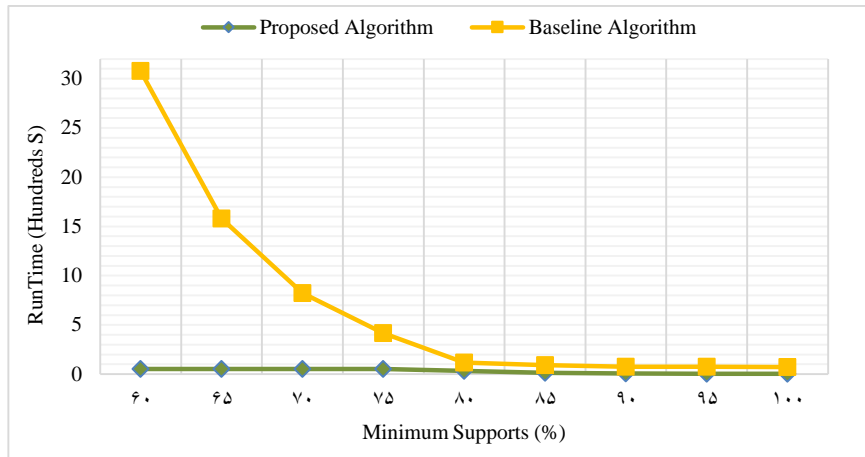
برای ارزیابی الگوریتم پیشنهادی از مجموعه داده استاندارد Mushroom که دارای ۸۱۲۴ تراکنش، ۲۳ آیت و اعداد کمی در بازه [۱،۱۱۹] است، استفاده شده است. به منظور مقایسه کارایی الگوریتم پیشنهادی از منظر معیارهایی مانند زمان اجرای برنامه و مقیاس پذیری از الگوریتم پایه MFFI-Miner [۸] که جز جدیدترین رویکردها در این حوزه است استفاده می شود. آزمایشها بر روی یک سیستم با حافظه ۸ گیگابایت و پردازنده Core i5 با سیستم عامل ویندوز ۱۰، ۶۴ بیتی اجرا شده اند. در ابتدا مقایسه ی بین الگوریتم پیشنهادی با الگوریتم پایه از لحاظ زمان اجرای برنامه انجام می شود و سپس الگوریتمها بر اساس مقیاس پذیری مورد ارزیابی قرار می گیرند. به هنگام ارزیابی الگوریتم پیشنهادی از لحاظ زمان اجرا، ابتدا مقایسه بر اساس یک مقدار حداقل آستانه پشتیبانی واحد انجام و سپس از چندین آستانه پشتیبانی استفاده می شود. از آنجاکه الگوریتم پایه از یک مقدار حداقل آستانه پشتیبانی استفاده می کند، به هنگام مقایسه در حالت چندین آستانه پشتیبانی، از مقدار LMS که در رابطه ی (۲) برای محاسبه مقادیر آستانه پشتیبانی چندگانه ارائه شده است استفاده می شود. در شکل (۴)، روند تغییرات زمان اجرای الگوریتم پیشنهادی و الگوریتم پایه مبتنی بر استفاده از حداقل آستانه پشتیبانی واحد مشاهده می شود. بر طبق این شکل، الگوریتم پیشنهادی زمان اجرای کمتری نسبت به الگوریتم MFFI-Miner دارد. علت این امر استفاده از روشهایی جهت حذف مجموعه آیت هایی که قادر به استخراج الگوهای مکرر نمی باشند و در نتیجه کاهش فضای جستجو در الگوریتم پیشنهادی است. هرچقدر مقدار حداقل آستانه پشتیبانی کاهش یابد، الگوریتم پیشنهادی کارآمدی خود را بیشتر نشان می دهد.



شکل ۴: زمان اجرای الگوریتمها با مقادیر مختلف حداقل آستانه پشتیبانی و مقدار حداقل آستانه پشتیبانی واحد

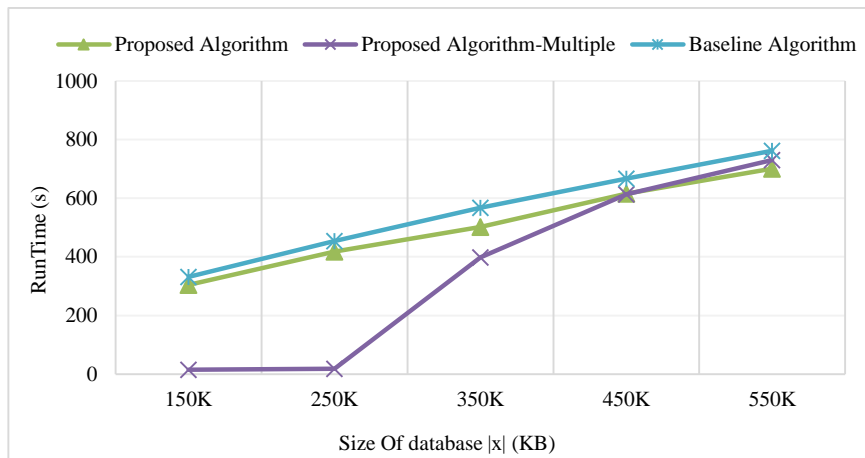
در شکل (۵)، روند تغییرات زمان اجرای الگوریتم پیشنهادی و الگوریتم پایه با حداقل آستانه پشتیبانی چندگانه برای الگوریتم پیشنهاد شده نشان داده شده است. در این حالت مقدار $\beta = 0.5$ قرار داده شده است. همان طور که قابل مشاهده است الگوریتم پیشنهادی از لحاظ زمان اجرای الگوریتم در این حالت نیز همچون زمانی که از مقدار حداقل آستانه پشتیبانی واحد استفاده می شود، وضعیت مناسب تری نسبت به الگوریتم پایه قرار دارد.

سومین همایش ملی مهندسی کامپیوتر، داده کاوی و داده های حجیم



شکل ۵: زمان اجرای الگوریتم‌ها با مقادیر مختلف حداقل آستانه پشتیبانی و مقدار حداقل آستانه پشتیبانی چندگانه

در ادامه، ارزیابی از لحاظ مقیاس پذیری انجام می‌شود. در این ارزیابی، زمان اجرای الگوریتم پیشنهادی با تعداد تراکنش‌های مختلف مجموعه داده Mushroom با الگوریتم پایه مورد مقایسه قرار گرفته‌اند. الگوریتم پیشنهادی در هر دو حالت حداقل آستانه پشتیبانی واحد Proposed Algorithm و حداقل آستانه پشتیبانی چندگانه Proposed Algorithm-Multiple مورد بررسی قرار گرفته است. مطابق شکل (۶)، همان‌طور که نشان داده شده است، زمان اجرای الگوریتم‌ها همراه با افزایش اندازه مجموعه داده، تقریباً به صورت خطی افزایش می‌یابند ولی این افزایش در الگوریتم پیشنهادی و در هر دو حالت نسبت به الگوریتم پایه کمتر می‌باشد.



شکل ۶: مقیاس پذیری الگوریتم‌های پیشنهادی از لحاظ زمان اجرای الگوریتم

۵- نتیجه گیری

در این پژوهش به منظور هرس داده‌هایی که پتانسیل تولید مجموعه‌های فازی مکرر را ندارند و همین‌طور جهت کاهش فضای جستجو، الگوریتمی پیشنهاد شده است که از کوچک‌ترین آستانه پشتیبانی بر روی درخت شمارشی استفاده می‌کند. الگوریتم پیشنهادی می‌تواند بدون تولید مجموعه آیت‌های فازی کاندید که از مشخصه‌های تمامی الگوریتم‌های بر پایه الگوریتم Apriori است، به صورت مستقیم، با کاوش درخت شمارشی و استفاده از روش‌هایی جهت حذف اطلاعات غیر جذاب و بدون نیاز به کاوش چندباره کل پایگاه داده، به استخراج الگوهای مکرر فازی چندگانه بپردازد و زمان اجرا را بهبود بخشد. اثربخشی و کارایی الگوریتم پیشنهادی برای استخراج مجموعه آیت‌های مکرر فازی با چندین آستانه پشتیبانی مورد ارزیابی قرار گرفت. از نتایج، می‌توان یافت که الگوریتم پیشنهادی به طور قابل ملاحظه‌ای از الگوریتم پایه MFFI-Miner

از لحاظ زمان اجرا با مقادیر مختلف حداقل آستانه پشتیبانی و در مقیاس های مختلف داده، بهتر عمل می کند. بر اساس الگوریتم پیشنهادی مشکل نادیده گرفتن آیت هایی که به ندرت در تراکنش ها رخ می دهند، رفع می گردد. الگوهای مکرر کمتر، اما مفیدتری تولید می شوند که می تواند برای کمک به مدیران یا بهره برداری در سیستم های خبره و هوشمند استفاده شوند. در کارهای آتی، چگونگی گسترش روش پیشنهادی و کاوش الگوهای مکرر از روی داده های توالی و همین طور از روی پایگاه های داده ی پویا که حاوی اطلاعات سری زمانی هستند می تواند یک موضوع جالب و چالش برانگیز باشد.

۶- مراجع

1. J. Han 1, M. Kamber 2 and J. Pei 3, *Data Mining Concepts and Techniques Preface and Introduction*. The Morgan Kaufmann, pp.248-262, 1999.
2. C. Zhang 1 and Shichao Zhang 2, "Association Rule Mining, Models and Algorithms" Springer-Verlag Berlin Heidelberg NewYork, vol. 2307, pp. 199-223, 2002.
3. M. Heikki 1, T. Hannu 2 and a. I. Verkamo 3, "Efficient Algorithms for Discovering Association Rules" *AAAI Workshop on Knowledge Discovery in Databases KDD94*, vol. 118, NO. 1-4, pp. 181-192, 1994.
4. H. Toivonen 1, "Sampling Large Databases for Association Rules," *Proceedings of the 22nd VLDB Conference, Mumbai, India*, pp. 1-12, 1996.
5. R. Agrawal 1 and R. Srikant 2, "Mining quantitative association rules in large relational tables," *ACM SIGMOD Record*, pp. 1-12, 1996.
6. Z. Qureshi 1, J. Bansal 2 and S. Bansal 3, "A Survey on Association Rule Mining in Cloud Computing" *International Journal of Emerging Technology and Advanced Engineering*, vol. 3, NO. 4, pp. 318-321, 2013.
7. P. Santhi Thilagam 1 and V. S. Ananthanarayana 2, "Extraction and optimization of fuzzy association rules using multi-objective genetic algorithm" *Pattern Analysis and Applications*, vol. 11, NO. 2, pp. 159-168, 2008.
8. C. W. Lin 1, T. P. Hong 2 and W. H. Lu 3, "Linguistic data mining with fuzzy FP-trees" *Expert Systems with Applications*, vol. 37, NO. 6, pp. 4560-4567, 2010.
9. C. W. Lin 1 and T. P. Hong 2, "Mining fuzzy frequent itemsets based on UBFFP trees" *Journal of Intelligent and Fuzzy Systems*, vol. 27, NO. 1, pp. 535-548, 2014.
10. B. Wang 1, X. Liu 2 and L. Wang 3, "Mining Fuzzy Association Rules in the Framework of AFS Theory" *Annals of Data Science*, vol. 2, NO. 3, pp. 261-270, 2015.
11. L. Wang 1, J. Y. Dong 2 and S. L. Li 3, "Fuzzy Inference Algorithm based on Quantitative Association Rules" *Procedia Computer Science*, vol. 61, pp. 388-394, 2015.
12. R. Agrawal 1 and R. Srikant 2, "Fast Algorithms for Mining Association Rules in Large Databases" *Journal of Computer Science and Technology*, vol. 15, NO. 6, pp. 487-499, 1994.
13. Y. M. Bing Liu 1 and W. Hsu 2, "Mining Association Rules with Multiple Minimum Supports," *In Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 337-341, 1999.
14. M. C. Tseng 1 and W. Y. Lin 2, "Efficient Mining of Generalized Association Rules with Non-uniform Minimum Support" *Data & Knowledge Engineering*, vol. 62, NO. 1, pp. 41-64, 2007.
15. Y. Lee 1, T. Hong 2, and T. Wang 3, "Mining Fuzzy Multiple-level Association Rules under Multiple Minimum Supports," *2006 IEEE International Conference on Systems, Man and Cybernetics, Taipei*, pp. 4112-4117, 2006.
16. T. C. Huang 1, "Discovery of fuzzy quantitative sequential patterns with multiple minimum supports and adjustable membership functions," *Information Sciences*, vol. 222, pp. 126-146, 2013.
17. Y. H. Hu 1 and Y. L. Chen 2, "Mining association rules with multiple minimum supports: a new mining algorithm and a support tuning mechanism" *Decision Support Systems*, vol. 42, NO. 1, pp. 1-24, 2006.
18. R. U. Kiran 1 and P. K. Reddy 2, "Novel Techniques to Reduce Search Space in Multiple Minimum Supports-Based Frequent Pattern Mining Algorithms," *14th International Conference on Extending Database Technology*, pp. 11-20, 2011.
19. J. C. W. Lin 1, T. Li 2, P. Fournier-Viger 3, T. P. Hong 4, J. M. T. Wu 5, and J. Zhan 6, "Efficient Mining of Multiple Fuzzy Frequent Itemsets" *International Journal of Fuzzy Systems*, vol. 19, NO. 4, pp. 1032-1040, 2017.