سمینار دوره کارشناسی ارشد

مهندسی کامپیوتر- نرم‌افزار

عنوان سمینار:

کاوش مجموعه‌داده‌های سودمند

دانشجو:

پروین تقوی

استاد راهنما:

دکتر نگین دانشپور

تیر ۹۹


بسم الله الرحمن الرحیم

تقدیر و تشکر

با سپاس فراوان از راهنمایی‌ها و زحمات استاد محترم خانم دکتر دانشپور که مرا در این مسیر یاری نمودند و خانم دکتر ترابی که با مشورت خود مرا راهنمایی کردند.

چکیده

امروزه با وجود حجم بالای داده، یافتن رابطه‌ی با معنی بین داده‌ها و استخراج دانش مفید امری ضروری و کارآمد برای تصمیم‌گیری است. در اغلب روش‌های گذشته که به استخراج این روابط با معنی و یا داده‌های مفید پرداخته‌اند، به سود آیتم‌ها توجهی نشده‌است. حال آن که داشتن اطلاعات درباره داده‌های سودمند می‌تواند در تصمیم‌گیری‌ها کارآمدتر باشد. در این تحقیق سعی در بررسی روش‌هایی داریم که با توجه به میزان سودمندی هر آیتم، مجموعه‌ای از داده‌هایی را که نسبت به یک آستانه سودمندی از پیش تعیین‌شده سود بیشتری دارند کشف کنند.

کلیدواژه‌ها

داده‌کاوی، مجموعه‌داده‌ی سودمند، مجموعه‌داده‌ی تکراری، آستانه‌ی سودمندی

فهرست عناوین

فصل اول

مقدمه و شرح مسئله

1-1- مقدمه

از سال ۱۹۵۱ به بعد که رایانه، در تحلیل و ذخیره‌سازی داده‌ها[1] به کار رفت، حجم اطلاعات ذخیره شده در آن پس از حدود ۱۰ سال دو برابر شد و هم‌زمان با پیشرفت فناوری اطلاعات، حجم داده‌ها در پایگاه داده‌ها[2] هر دو سال یک‌بار، دو برابر شد[۱] و همچنان با سرعت بیش‌تری نسبت به گذشته حجم اطلاعات ذخیره شده بیش‌تر و بیش‌تر می‌شود. باوجود منابع اطلاعاتی مانند شبکه جهانی وب، سیستم‌های یکپارچه اطلاعاتی، سیستم‌های یکپارچه بانکی و تجارت الکترونیک لحظه‌به‌لحظه به حجم داده‌ها در پایگاه داده‌ها اضافه شده و باعث به وجود آمدن انبارهای عظیمی از داده‌ها شده‌است، به‌طوری‌که ضرورت کشف و استخراج سریع و دقیق دانش از این پایگاه‌های داده را بیش‌ازپیش نمایان کرده است. شدت رقابت‌ها در عرصه‌های علمی، اجتماعی، اقتصادی، سیاسی و نظامی نیز اهمیت سرعت یا زمان دسترسی به اطلاعات را دوچندان کرده است. بنابراین نیاز به طراحی سیستم‌هایی که قادر به اکتشاف سریع اطلاعات مورد علاقه کاربران با تأکید بر حداقل مداخله انسانی باشند از یک‌سو و روی آوردن به روش‌های تحلیل متناسب با حجم داده‌های حجیم از سوی دیگر، به‌خوبی احساس می‌شود. در حال حاضر، داده‌کاوی[3] مهم‌ترین فناوری برای بهره‌وری مؤثر، صحیح و سریع از داده‌های حجیم است و اهمیت آن رو به فزونی است. داده‌کاوی فرآیندی پیچیده جهت شناسایی الگوها و مدل‌های صحیح، جدید و به‌صورت بالقوه مفید، در حجم وسیعی از داده می‌باشد، به طریقی که این الگوها و مدل‌ها برای انسان‌ها قابل‌درک باشند. داده‌ها اغلب حجیم می‌باشند و به تنهایی قابل استفاده نیستند، اما دانش نهفته در داده‌ها قابل استفاده می‌باشد. بنابراین بهره‌گیری از قدرت فرآیند داده‌کاوی جهت شناسایی الگوها و مدل‌ها و نیز ارتباط عناصر مختلف در پایگاه داده جهت کشف دانش نهفته در داده‌ها و نهایتاً تبدیل

---

[1] Data

[2] Database

[3] Datamining

داده به اطلاعات، روزبه‌روز ضروری‌تر می‌شود. در داده‌کاوی معمولاً به کشف الگوهای مفید از میان داده‌ها اشاره می‌شود. منظور از الگوی مفید مدلی در داده‌ها است که ارتباط میان یک زیرمجموعه از داده‌ها را توصیف می‌کند و معتبر، ساده، قابل‌فهم و جدید است. روش‌های مختلفی برای کشف این الگوها ارائه شده است که در ادامه به شرح مختصری از دو روشی که پایه‌ی مسئله‌ی استخراج مجموعه-داده‌های پرسود هستند، می‌پردازیم.

2-1- روش‌های پایه‌ی مسئله‌ی استخراج مجموعه‌داده‌های سودمند

مزیت این روش‌ها که مبتنی بر الگویابی هستند بر چندین رویکرد داده کاوی دیگر این است که کشف الگوها نوعی یادگیری بدون نظارت است که نیازی به داده‌های برچسب‌دار ندارد. برای حل مسئله-ی استخراج مجموعه‌داده‌های سودمند[1] از بسط روش کشف مجموعه‌داده‌های پرتکرار استفاده شده‌است و از روش کشف قواعد وابستگی نیز ایده گرفته شده‌است.

۱-۲-۱- روش کشف مجموعه‌داده‌های پرتکرار

یکی از مسائل مشهور در زمینه‌ی داده‌کاوی مسئله‌ی استخراج مجموعه‌داده‌های پرتکرار (FIM[2]) بوده است. هدف آن پیدا کردن گروهی از آیتم‌هاست که به صورت تکراری در پایگاه داده وجود دارند. از ویژگی‌های این مسئله بسته‌شدن رو به پایین است و بیان می‌کند که ابرمجموعه‌های یک مجموعه‌داده‌ی غیرتکراری، غیرتکراری هستند و زیرمجموعه‌های یک مجموعه‌داده‌ی مکرر، پرتکرار هستند[۲]. اگر فرض کنیم پایگاه داده‌ی ما شامل تراکنش‌های خرید باشد در این صورت یکی از مشکلات این روش این است که به تعداد اقلام خریداری‌شده در هر تراکنش توجه نمی‌شود. در واقع

[1] High Utility Itemset Mining

[2] Frequent Itemsets Mining

تنها وجود و یا عدم وجود اقلام مهم است. مسئله‌ی دیگر نیز این است که در این روش تمام آیتم‌ها از ارزش یکسانی برخوردار هستند.

۲-۲-۱- روش کشف قواعد وابستگی

کاوش قواعد وابستگی[1] آن دسته از مسائل داده‌کاوی را شامل می شود که در آن به دنبال استخراج و تعریف قواعد و الگوهایی هستیم که توصیف دقیق‌تری را از فضای حاکم بر داده‌ها ارائه می‌دهند[۳]. کاربردهای گسترده‌ی این روش‌ها در هوش تجاری، شبکه های اجتماعی و مجازی، تجارت الکترونیک، صنعت بانکداری، وب‌کاوی و همچنین کاربرد آن در پیش‌بینی باعث می‌شود که این مسئله اهمیت زیادی داشته‌باشد. برای مثال با استفاده از این روش و با در نظر گرفتن یک پایگاه داده‌ی تراکنشی، رابطه‌ای را کشف می‌کنیم که در آن با خرید پنیر به احتمال ۷۰ درصد گردو و نان هم به فروش می‌رسد. با وجود تمام این‌ها در این روش نیز تمام اقلام از ارزش یکسانی برخوردار هستند، درصورتی که در واقعیت هر جنس و یا رویدادی از ارزش خاص خود برخوردار است.

3-1- تاریخچه و تعریف موضوع

مسئله‌ی استخراج مجموعه‌داده‌های پرسود زمانی مورد توجه قرار گرفت که یافتن داده‌های پرتکرار برای کاربردهای واقعی خیلی کارآمد نبودند و سود یکی از فاکتورهای مهم برای نتایج بهتر و موثرتر بود. اصطلاح مجموعه‌داده‌های پرسود در ابتدا در سال ۲۰۰۳ مطرح شد و در سال ۲۰۰۵ تعریف اولیه‌ای از آن ارائه شد[۲]. در این مسئله یک پایگاه داده که شامل آیتم‌ها و تعداد آن‌ها است، یک جدول که شامل ارزش هر آیتم است و آستانه سودمندی که توسط کاربر تعیین می‌شود به عنوان ورودی به الگوریتم داده می‌شود. در خروجی اگر سود مجموعه‌داده‌ای از این آستانه بیشتر بود به عنوان یک داده‌ی پرسود استخراج می‌شود. در واقع راه حل‌های این مسئله به طور کلی به دو دسته تقسیم می‌شوند که عبارتند

[1] Association Rule Mining

از روش‌هایی که در دوفاز اجرا می‌شوند و روش‌های تک‌فاز[۲]. اولین الگوریتمی که برای این مسئله ارائه شد الگوریتم two phase [۴] نام داشت که در دوفاز اجرا می‌شد. در روش دوفاز یک معیار TWU[1] تعریف می‌شود. در فاز اول تمام فضای جست‌وجو بررسی می‌شود و آن دسته از مجموعه‌داده‌هایی که مقدار سود آن‌ها از TWUشان بیشتر بود به عنوان مجموعه‌داده‌های کاندید در نظر گرفته می‌شوند. در فاز دوم سود دقیق اعضای مجموعه‌ی کاندید را با اسکن پایگاه داده محاسبه می‌کنند.

در روش تک فاز که در یک مرحله انجام می‌شود ابتدا با روش‌های هرس مناسب فضای جست‌وجو را کاهش می‌دهند و سپس سود آیتم‌هایی که باقی مانده‌است را با یک بار اسکن پایگاه داده محاسبه می‌کنند و با مقدار آستانه سودمندی مقایسه می‌کنند[۲].

فاکتورهایی که باعث می‌شود روش‌های ارائه‌شده با هم متفاوت باشند عبارتند از:

- تعداد مراحل اجرای الگوریتم
- نوع ساختمان داده‌ی به‌کاررفته
- روش جست‌وجو
- نمایش پایگاه داده
- معیارهای هرس فضای جست‌وجو

4-1- جمع‌بندی

در این فصل به چرایی ضرورت الگویابی در داده‌های حجیم پرداختیم و همچنین پیشینه و ویژگی-های کلی مسئله‌ی HUIM را بیان کردیم.

[1] Transaction Weighted Utility

در فصل بعدی به شرح راهکارهای ارائه‌شده برای مسئله‌ی استخراج مجموعه‌داده‌های پرسود پرداخته می‌شود و فاکتورهای بالا به طور کامل توضیح داده‌می‌شوند.

فصل دوم

مجموعه‌داده‌های پرسود؛ شرح راهکارها

2-1- مقدمه

در علم داده‌کاوی، کاوش الگوهایی با کارایی و سود بالا در حال ظهور است، که شامل کشف الگوهایی است که دارای اهمیت بالایی در پایگاه‌های اطلاعاتی هستند. کارایی یک الگو را می توان از نظر معیارهای عینی مختلف مانند سود، فرکانس و وزن اندازه گرفت. در میان انواع مختلف الگوهای کاربردی که می‌توانند در پایگاه‌های اطلاعاتی کشف شوند، داده‌هایی با سود بالا بیشتر مورد مطالعه قرار می‌گیرند. یک مجموعه‌داده با سود بالا مجموعه‌ای از مقادیر است که در پایگاه داده ظاهر می‌شود، سود آن توسط یک تابع منفعت اندازه‌گیری می‌شود و برای کاربر دارای اهمیت بالایی است. استخراج مجموعه‌داده‌های سودمند مشکل استخراج داده‌های مکرر را با در نظر گرفتن مقادیر و ارزش آیتم‌ها حل می‌کند.

مسئله‌ی استخراج مجموعه‌داده‌های سودمند کاربردهای مختلفی در زمینه‌ی تجارت الکترونیک، پزشکی، وب‌کاوی و بررسی جریان کلیک دارد. برای مثال در زمینه پزشکی می‌توان بررسی نمود که کدام روش درمان و یا داروها برای یک بیماری خاص مفیدتر است. همچنین با بررسی جریان کلیک کاربر در صفحات وب می‌توان دریافت که به چه موضوعاتی علاقه‌ی بیشتری دارد و آن‌ها را در ورود بعدی به کاربر معرفی نمود و یا محصولات مورد علاقه‌ی وی را پیدا کرد که سودآور نیز هستند. یک کاربرد معروف استخراج مجموعه‌داده‌های سودمند کشف همه مجموعه موارد خریداری‌شده توسط مشتریان است که سود بالایی را به دست می‌آورد. در این فصل به توضیح کامل راهکارهای ارائه‌شده برای حل مسئله‌ی کشف مجموعه‌داده‌های پرسود پرداخته می‌شود. در بخش تعریف مسئله، نحوه‌ی کلی محاسبه‌ی سود اقلام، کران‌های بالا برای کاهش فضای جست‌وجو در هر دو روش تک‌فاز و دوفاز توضیح داده خواهد شد. در ادامه به صورت مبسوط به بررسی روش‌های تک‌فاز و دوفاز می‌پردازیم و در بخش پایانی بسط‌های مختلفی از مسئله‌ی پایه را توضیح خواهیم داد.

2-2- تعریف اولیه مسئله

تعریف‌هایی که در این بخش به آن پرداخته می‌شود تعاریف کلی هستند که در همه‌ی کارهای انجام شده یکسان است بنابراین برای فهم بیشتر و یکپارچگی موضوع از تعاریف موجود در مقاله‌ی [۵] استفاده شده‌است.

فرض کنید I مجموعه‌ای شامل تمام اقلام موجود در تراکنش‌ها باشد. و $D = \{T_1, T_2, ..., T_n\}$ نیز پایگاه داده‌ای شامل تمام تراکنش‌ها است. هر آیتم $i \in I$ نیز به یک مقدار مثبت p(i) که سود خارجی حاصل از فروش یک قلم از آن است، مرتبط می‌شود و در هر تراکنش مانند T_c تعدادی از هر آیتم مانند i وجود دارد که این مقدار مثبت را با $q(i, Tc)$ نشان می‌دهند. توجه داشته باشید که در مسئله‌ی استخراج مجموعه‌داده‌های پرسود پایه، سود هر آیتم مثبت در نظر گرفته می‌شود.

پایگاه داده‌ی تراکنشی مورد استفاده در این نوشته در جدول(۲-۱) و سود خارجی هر آیتم در جدول(۲-۲) نشان داده‌شده‌است.

جدول۲-۱: پایگاه داده‌ی تراکنشی

TID	Transaction
T_1	$(a, 1)(c, 1)(d, 1)$
T_2	$(a, 2)(c, 6)(e, 2)(g, 5)$
T_3	$(a, 1)(b, 2)(c, 1)(d, 6)(e, 1)(f, 5)$
T_4	$(b, 4)(c, 3)(d, 3)(e, 1)$
T_5	$(b, 2)(c, 2)(e, 1)(g, 2)$

جدول۲-۲: سود خارجی اقلام

Item	*a*	*b*	*c*	*d*	*e*	*f*	*g*
Profit	5	2	1	2	3	1	1

تعریف۱: (سود یک آیتم و سود یک مجموعه‌داده). سود آیتمی مانند i در یک تراکنشی مانند T_c به روش زیر به‌دست می‌آید.

$$U(i,T_c)=q(i,T_c)\times p(i) \qquad (۱-۲)$$

سود مجموعه‌داده‌ای مانند $X \subseteq I$ که در تراکنش T_c حضور دارد، به صورت زیر به‌دست می‌آید. که در این‌جا g(X) مجموعه‌ای از تراکنش‌هاست که X در آن‌ها حضور دارد.

$$u(X,T_c)=\sum_{i\in X\wedge T_c\in g(X)} U(i,T_c) \qquad (۲-۲)$$

تعریف۲: (سود کلی یک مجموعه‌داده در کل پایگاه داده). سود کلی یک مجموعه‌داده از جمع سود آن در هر تراکنشی که حضور دارد به وجود می‌آید.

$$U(X)=\sum_{X\subseteq T_c\wedge T_c\in D} u(X,T_c) \qquad (۳-۲)$$

تعریف۳: (استخراج مجموعه‌داده‌های پرسود) زمانی مجموعه‌داده‌ای مانند X را پرسود نامیم که سود آن از مقدار آستانه‌ی سودمندی تعریف‌شده توسط کاربر (minutil[1]) کمتر نباشد.

تعریف۴: سود کلی یک تراکنش مانند T_c از جمع سود حاصل از هر آیتم موجود در آن به‌دست می‌آید.

[1]Minimum utility

$$TU(T_c) = \sum_{x \in T_c} U(x, T_c) \qquad (۴-۲)$$

تعاریفی که در بالا آورده شده‌است برای تمام روش‌های تک‌فار و دوفاز مورد استفاده قرار می‌گیرد.

2-3- روش حل دوفاز

همان‌طور که قبلا هم گفته‌شد الگوریتم‌های دوفاز از راه حل‌های اولیه‌ی ارائه‌شده در این زمینه بوده‌اند. عواملی که باعث می‌شوند روش‌های دوفاز با هم متفاوت باشند بستگی به این دارد که از چه روش جست‌وجویی استفاده شود و این‌که روش ارائه‌شده بسطی از چه الگوریتمی باشد. به طور کلی الگوریتم‌های دوفاز یا بسطی از الگوریتم Apriori هستند و یا بسطی از روش FP-Growth می‌باشند[۲]. هر دوی این الگوریتم‌ها روش‌هایی برای حل مسئله‌ی FIM هستند. در ابتدا از بسط روش Apriori استفاده شد که در این روش از جست‌وجوی سطحی برای کاوش فضای جست‌وجو بهره برده‌شد. مشکل این روش این بود که در جست‌وجوی سطحی حافظه‌ی زیادی مورد نیاز بود. بنابراین از بسط روش مبتنی بر رشد استفاده کردند. در این روش از جست‌وجوی عمقی استفاده شد که میزان مصرف حافظه در آن کمتر بوده است. بنا به تعریفی که در فصل گذشته برای روش‌های دوفاز بیان شد، روش‌های محاسبه‌ی سودی که برای اولین بار در این دسته از روش‌ها مورد استفاده قرار گرفت در زیر آورده شده‌است.

تعریف۴: سودمندی وزنی تراکنش برای یک مجموعه‌داده مانند X از جمع سود تمام تراکنش‌هایی که X در آن‌ها حضور داشته است به‌دست می‌آید.

$$TWU(X) = \sum_{T_c \in g(X)} TU(T_c) \qquad (۵-۲)$$

در جدول (۲-۳) مقدار TWU محاسبه‌شده برای تک‌آیتم‌ها آورده شده‌است.

جدول۲-۳: مقدار TWU برای تک‌آیتم‌ها

Item	*a*	*b*	*c*	*d*	*e*	*f*	*g*
TWU	65	41	96	58	88	30	38

ویژگی۱: اگر مقدار TWU برای یک مجموعه‌داده X از مقدار minutil کمتر باشد، آنگاه X و ابرمجموعه‌های آن جز موارد پرسود نخواهند بود.

در فصل قبل گفته شد که مقدار TWU معیاری است که بتوان با استفاده از آن و ویژگی۱ اعضای مجموعه‌ی کاندید را در روش‌های دوفاز مشخص کرد. سپس با اسکن پایگاه داده و بر اساس رابطه‌ی۲-۳ مقدار سود هر عضو کاندید محاسبه می‌شود. به دلیل این‌که روش‌های دوفاز جز راه‌حل‌های اولیه برای حل این مسئله هستند، قدیمی بوده و از توضیح و بسط آن اجتناب می‌کنیم.

2-4- روش حل تک‌فاز

به طور معمول در روش‌های تک‌فاز از دو ساختمان داده استفاده می‌شود. یکی از ساختمان‌های داده لیست سودمندی است و دیگری پایگاه داده‌ی واکشی‌شده می‌باشد. الگوریتم HUI-Miner که در [۶] آمده‌است، اولین روشی است که به صورت تک فاز ارائه شد و کاستی‌های روش دوفاز را پوشش داد. در این روش از لیست سودمندی استفاده شده‌است. الگوریتم EFIM که در [۵] آمده‌است نیز از روش‌های پایه در زمینه تک‌فاز بودن است که در آن از پایگاه‌داده‌ی واکشی‌شده استفاده شده‌است. درادامه به توضیح لیست سودمندی و پایگاه داده‌ی واکشی‌شده می‌پردازیم.

۱-۴-۲- روش‌های مبتنی بر لیست سودمندی

در مقالات [۷]، [۸]، و [۹] که از این روش استفاده کرده‌اند سعی شده‌است که با ارائه‌ی راه حلی هزینه ادغام لیست سودمندی را کاهش دهند. برای مثال در [۷] با ارائه‌ی روشی هر آیتم در هر تراکنش

با آیتم‌های قبل خود در ارتباط است و مقایسه‌ی دو لیست سودمندی را به‌طور هوشمندانه‌ای انجام می‌دهد تا در مصرف زمان و حافظه صرفه‌جویی شود. یا در [۸]و [۹] با ارائه‌ی روشی سعی در کمتر کردن حافظه‌ی مصرفی توسط لیست سودمندی می‌شود. در مقالات دیگر مانند [۱۰]، [۱۱]، [۱۲]، [۱۳]، [۱۷] و [۱۵] هر یک با استفاده از این روش سعی در حل بسطی از مسئله‌ی پایه HUIM دارند. تعریف‌هایی که در این بخش به آن پرداخته می‌شود تعاریف کلی هستند که در همه‌ی کارهای مبتنی بر لیست سودمندی یکسان است. بنابراین برای فهم بیشتر و یکپارچگی موضوع از تعاریف موجود در مقاله‌ی [۶] استفاده شده‌است.

در این روش هر مجموعه‌داده با یک لیست سودمندی مرتبط است. لیست سودمندی برای تک‌آیتم‌ها لیست سودمندی اولیه است که می‌تواند توسط دو اسکن پایگاه‌داده ساخته شوند. در اسکن اول مقدار TWU برای اقلام محاسبه می‌شود. اگر مقدار آن از آستانه سودمندی کمتر باشد دیگر در روند استخراج در نظر گرفته نمی‌شود. فرض کنیم $\succ$ ترتیبی برای آیتم‌ها باشد. معیاری که برای مرتب کردن آیتم‌ها استفاده می‌کنیم بر اساس معیار TWU آن‌هاست که کمتر از آستانه سودمندی نبودند و به صورت صعودی مرتب می‌شوند. حال بنا بر این توضیحات پایگاه داده بر اساس ترتیب جدید مرتب می‌شود. این مراحل مرتب‌سازی طی اسکن دوم پایگاه داده انجام می‌شود. برای پایگاه‌داده‌های بزرگ، به منظور کاهش بیشتر فضای جستجو و افزایش کارایی HUIM به حدود بالای دقیق‌تری نسبت به TWU نیاز است. محاسبه حدود بالایی جدید متکی بر سود باقی‌مانده است که به صورت زیر تعریف می‌شود. فرض کنیم $\succ$ ترتیبی برای آیتم‌ها باشد. معیاری که برای مرتب کردن آیتم‌ها استفاده می‌کنیم بر اساس معیار TWU آن‌هاست و به صورت صعودی مرتب می‌شوند. ترتیب آیتم‌های مرتبط با پایگاه داده‌ی جدول۱ به صورت $c \succ e \succ a \succ d \succ b \succ g \succ f$ خواهد بود.

لیست سودمندی مجموعه‌داده‌ی X یک سه‌تایی شامل (tid, iutil, rutil) خواهد بود که tid معرف شماره تراکنشی است که X در آن حضور دارد، iutil سود حاصل از مجموعه‌داده‌ی X موجود در

تراکنش مورد نظر است و rutil سود باقی‌مانده‌ی حاصل از آیتم‌هایی است که با توجه به ترتیب تعریف-شده بعد از X می‌آیند. تعریف سود باقی‌مانده در زیر آمده‌است.

تعریف۵: (سود باقی‌مانده برای یک مجموعه‌داده). فرض کنید X یک مجموعه‌داده در تراکنش T_c است. سود باقی‌مانده برای آن برابر با مجموع سود آیتم‌هایی است که بعد از تمام آیتم‌های موجود در X بر اساس ترتیب $\succ$ می‌آید.

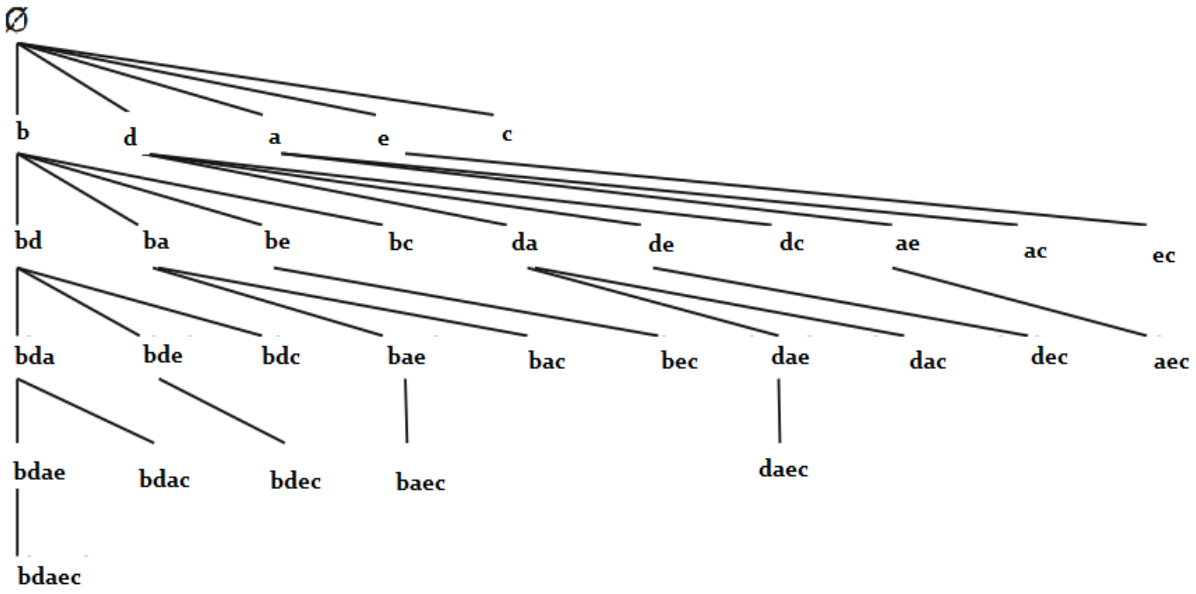
$$ru(X,T_c) = \sum_{i \in T_c \wedge i \succ x \forall x \in X} u(i,T_c) \qquad (۶-۲)$$

برای ساخت لیست سودمندی مجموعه‌داده‌های دوتایی از روی لیست سودمندی تک‌آیتم‌ها، مقدار tid لیست سودمندی دو آیتمی که قرار است با هم مجموعه‌داده‌ی دوتایی را بسازند، مقایسه می‌شوند. این مقایسه به دلیل مرتب بودن شماره تراکنش‌ها به صورت صعودی، به روش دو طرفه انجام می‌شود. سپس به ازای تراکنش‌های هم‌شماره مقدار iutil از جمع سود آن‌ها و مقدار rutil از لیست سودمندی آیتمی که اولویت بیشتری داشته حاصل می‌شود.

برای ساخت لیست سودمندی مجموعه‌داده‌هایی با طول بیش از دو، مانند روش قبلی ابتدا تراکنش-هایی که شماره یکسانی در هر دو لیست دارند مشخص می‌شوند. سپس برای محاسبه سود جدید، سود هر دو لیست پایه با هم جمع می‌شود و مقدار سود k-۲ آیتم قبلی از این عدد کم می‌شود. محاسبه سود باقی‌مانده نیز مانند قبل است.

روند اجرای الگوریتم‌های مبتنی بر سودمندی به این صورت است که برای جست‌وجوی مجموعه-داده‌های پرسود از درخت شمارشی و با استفاده از جست‌وجوی اول عمق عمل می‌کنند. اقلام موجود در درخت به ترتیب صعودی TWU مرتب می‌شوند. بدین ترتیب اگر جمع iutil و rutil یک مجموعه‌داده از مقدار آستانه سودمندی کمتر باشد آنگاه می‌توان خودش و زیردرخت‌اش را حذف نمود. اگر فرض کنیم مقدار آستانه سودمندی تعریف‌شده ۴۰ باشد آنگاه دو آیتم f و g حذف می‌شوند زیرا مقدار TWU

آن‌ها کمتر از minutil است. با این فرض در شکل۲-۱ درخت شمارشی مرتبط با پایگاه داده‌ی جدول (۲-۱) آمده است.



شکل۲-۱: درخت شمارشی برای نمایش فضای جست‌وجو

روند الگوریتم کاوش بدین صورت است که لیست سودمندی مجموعه‌داده‌ی مورد بررسی و تمام بسط‌های یک‌تایی آن به عنوان ورودی دریافت می‌شوند. اگر جمع مقدار iutil کمتر از آستانه سودمندی نبود آن را به عنوان مجموعه‌داده‌ی پرسود شناسایی می‌کند. در ادامه برای بسط بیشتر مجموعه‌داده شرط لازم این است که جمع iutil و rutil آن کمتر از آستانه سودمندی نباشد. اگر شرط برقرار بود در این صورت از بسط یک‌تایی، بسط‌های دوتایی ساخته می‌شوند و بررسی ادامه پیدا می‌کند تا تمام مجموعه‌داده‌های پرسود کشف شوند.

۲-۴-۲- روش‌های مبتنی بر پایگاه داده‌ی واکشی‌شده

دراین روش‌ها که مقالات [۵]، [۱۶] و [۱۷] از آن بهره برده‌اند، برای عمل جست‌وجو از روش جست‌وجوی عمق اول استفاده می‌شود. تفاوت این روش‌ها در این است که برای مثال در [۱۷] به حل بسطی از مسئله‌ی پایه که مربوط به یافتن kمجموعه‌داده‌ی پرسود است، پرداخته شده‌است، و در [۱۶]

سعی شده‌است تا با ارائه‌ی یک الگوریتم موازی روش ارائه‌شده در [۵] را بهبود بخشد. تعریف‌هایی که در این بخش به آن پرداخته می‌شود تعاریف کلی هستند که در همه‌ی کارهای مبتنی بر پایگاه داده‌ی واکشی‌شده یکسان است. بنابراین برای فهم بیشتر و یکپارچگی موضوع از تعاریف موجود در مقاله‌ی [۵] استفاده شده‌است. برای بسط هر مجموعه‌داده، با یک آیتم به صورت بازگشتی عمل می‌شود. برای این منظور مجموعه‌ی $E(X) = \{z \mid z \in I \wedge z \succ x, x \in X\}$ تعریف شده‌است.

تعریف۶: (مجموعه‌ی بسط برای یک مجموعه‌داده). تمام آیتم‌هایی که در بسط یک مجموعه‌داده مانند X می‌توانند مورد استفاده قرار گیرند در یک مجموعه $E(X) = \{z \mid z \in I \wedge z \succ x, x \in X\}$ تعریف می‌شود.

برای کاهش هزینه اسکن پایگاه داده، کاهش اندازه پایگاه داده بهتر است. برای این منظور هنگامی که یک مجموعه α در طول جست‌وجوی اولیه در نظر گرفته می‌شود، همه آیتم‌هایی که در مجموعه‌ی بسط α نیستند را می‌توان در هنگام اسکن پایگاه‌داده برای محاسبه سودمندی مجموعه‌داده‌ها در زیر درخت α نادیده گرفت.

تعریف۷: واکشی تراکنش T برای مجموعه‌داده‌ای مانند α به صورت $\alpha - T = \{i \mid i \in T \wedge i \in E(\alpha)\}$ تعریف می‌شود. حال واکشی پایگاه داده برای X به صورت $\alpha - D = \{\alpha - T \mid T \in D \wedge \alpha - T \neq \phi\}$ تعریف می‌شود.

این روش معمولا هزینه اسکن پایگاه داده را تا حد زیادی کاهش می‌دهد زیرا با بزرگ‌تر شدن تعداد آیتم‌هایی که مورد بررسی قرار می‌گیرند، تراکنش‌ها کوچک‌تر می‌شوند. اما روش پیاده‌سازی آن مهم است. یک روش ساده و ناکارآمد ایجاد کپی‌های فیزیکی از تراکنش‌ها برای هر واکشی است. برای این منظور ابتدا اقلام در هر تراکنش بر اساس ترتیبشان که قبلا ذکر شد، مرتب می‌شوند. سپس هر تراکنش واکشی‌شده با یک اشاره‌گر آفست بر روی پایگاه‌داده‌ی اصلی مورد اشاره قرار می‌گیرد. به عبارتی دیگر برای هر مجموعه‌داده به تعداد تراکنش‌هایی که در آن حضور دارد اشاره‌گر خواهیم داشت و به اقلامی

که در آن تراکنش اولویتشان بیشتر است اشاره می‌شود. در این صورت پیچیدگی محاسبه‌ی پایگاه‌داده‌ی واکشی‌شده $O(nw)$ خواهد بود که n نشان‌دهنده‌ی تعداد تراکنش‌ها و w نشان‌دهنده‌ی طول متوسط هر تراکنش است. با این حال، با بررسی مجموعه‌داده‌های بزرگ‌تر، اندازه پایگاه‌های داده پیش‌بینی‌شده کاهش می‌یابد.

روش دیگر برای کاهش هزینه اسکن پایگاه‌داده بر این اساس است که پایگاه‌های داده تراکنشی اغلب دارای تراکنش‌های یکسان هستند. این روش شامل شناسایی این تراکنش‌ها و سپس جایگزین کردن آن‌ها با تراکنش‌های تکی، و در عین حال جمع سود داخلی آن‌ها است. ادغام تراکنش‌های یکسان اندازه پایگاه‌داده را کاهش می‌دهد. اما این کاهش کوچک است اگر پایگاه‌داده شامل تعداد کمی تراکنش یکسان باشد. برای کاهش بیشتر پایگاه‌داده، تراکنش‌ها را در پایگاه‌های داده واکشی‌شده ادغام می‌کنیم. این امر به طور کلی به کاهش بسیار بالاتری دست می‌یابد زیرا تراکنش‌های پیش‌بینی‌شده کوچک‌تر از تراکنش‌های اصلی هستند، و بنابراین احتمال یکسان بودن آن‌ها بیشتر است. با این حال، مشکل اصلی پیاده‌سازی کارآمد آن است. روش ساده برای شناسایی تراکنش‌های یکسان، مقایسه تمام تراکنش‌ها با یکدیگر است. برای پیاده‌سازی کارآمد، این روش تراکنش‌ها را بر اساس یکسری تعاریف مرتب می‌کند. مثلا برای تراکنش‌هایی که یکسان هستند، آن تراکنشی از اولویت بیشتری برخوردار است که شماره بزرگ‌تری دارد؛ یا اگر اشتراک دو تراکنش، تراکنشی شود که تعداد اقلام آن کمتر است؛ تراکنش بزرگتر از اولویت بیشتری برخوردار است. به این ترتیب تراکنش‌ها در پایگاه داده تنها یک بار مرتب می‌شوند. در این حالت می‌شود که از زمان محاسبه‌ی این کار صرف نظر کرد. با استفاده از ویژگی فوق، تمام تراکنش‌های یکسان در یک پایگاه داده واکشی‌شده را می‌توان تنها با مقایسه هر تراکنش با تراکنش بعدی در پایگاه‌داده شناسایی کرد. هر مقایسه بین دو تراکنش می‌تواند در زمان خطی با استفاده از یک مقایسه دو طرفه انجام شود. بنابراین، هزینه کلی ادغام تمام تراکنش‌ها در یک پایگاه داده واکشی‌شده برابر است با $O(nw)$.

2-5- بسط مسئله‌ی استخراج مجموعه‌داده‌های پرسود

توضیحاتی که تاکنون داده‌شد مربوط به مسئله‌ی پایه‌ی HUIM بوده‌است که در آن مقدار سود اقلام تنها مثبت است و فرض می‌شود پایگاه داده حاوی آیتم‌هایی است که همیشه وجود دارند. اما در زندگی واقعی همیشه شرایط مسئله به این سادگی نیست. در ادامه به بسط‌های متفاوتی از مسئله‌ی HUIM خواهیم پرداخت.

۱-۵-۲- مسئله k مجموعه‌داده‌ی پرسود

تعیین مقدار آستانه سودمندی مناسب یک امر بسیار دقیق، حساس و زمان‌بر است به این معنی که اگر مقدار تعیین شده کوچک باشد تعداد بسیار زیادی مجموعه‌داده به عنوان اقلام پرسود استخراج می‌شوند. اگر این مقدار بزرگ باشد شاید هیچ مجموعه‌داده‌ی پرسودی استخراج نشود. بنابراین تعیین این مقدار توسط کاربر نیاز به تخصص لازم دارد. پس بر آن شدند که به جای تعیین مقدار آستانه سودمندی، کاربر تنها تعداد مجموعه‌داده‌های پرسودی که مد نظرش است را تعیین کند. برای حل این مسئله یک آستانه سودمندی داخلی تعریف می‌شود که آن را در ابتدا با صفر یا یک مقداردهی می‌کنند. سپس با استفاده از راهبردهای مناسب، این مقدار را افزایش می‌دهند تا k مجموعه‌داده‌ی مورد نظر استخراج شود. مشخص است که این مسئله از مسئله‌ی پایه پیچیده‌تر و زمان‌برتر است. مقالاتی که در آن‌ها به حل مسئله‌ی k مجموعه‌داده‌ی پرسود پرداخته شده‌است، [۱۵]، [۱۷] و [۱۸] هستند.

راهبردهای مختلفی برای افزایش مقدار آستانه سودمندی داخلی وجود دارد. در ادامه به برخی از این راهبردها اشاره خواهیم کرد. برای مثال از چهار تعریف زیر به عنوان راهبردی برای افزایش آستانه سودمندی در مقاله‌ی [۱۷] استفاده شده‌است.

تعریف۸: راهبرد[1] RIU مقدار سود واقعی تک‌آیتم‌ها را محاسبه می‌کند و مقدار آستانه سودمندی داخلی را از صفر یا یک به kامین عدد محاسبه شده که بیشترین سود را داراست، تغییر می‌دهد.

در بسیاری از راهبردهای معرفی‌شده از ساختار ماتریس مثلثی استفاده می‌شود. این ساختارها بسته به این‌که چه مقداری را در خود ذخیره می‌کنند، متفاوتند. برای مثال راهبردهای CUD[2] و COV[3] که در [۱۷] استفاده شده‌است، RIRU[4] و RIRU2[5] که در [۱۵] استفاده شده‌اند از ساختار ماتریس مثلثی استفاده می‌کنند.

تعریف۹: راهبرد CUD از یک ساختار ماتریس مثلثی به نام CUDM[6] استفاده می‌کند که در آن مقدار سود واقعی جفت‌آیتم‌هایی که از ترکیب هر آیتم با اقلام بعد از خودش به وجود می‌آید، ذخیره می‌شود. مقدار آستانه سودمندی داخلی به kامین عدد محاسبه‌شده که بیشترین مقدار را در این ماتریس داراست، تغییر می‌کند.

تعریف۱۰: با فرض دو آیتم x و y اگر $g(X) \subseteq g(Y)$ باشد، آنگاه y پوششی برای x خواهد بود.

تعریف۱۱: در راهبرد COV مقدار سود ترکیب هر آیتمی مانند i با اعضای مجموعه‌ی پوشش آن محاسبه می‌شود. مقدار آستانه سودمندی داخلی به kامین عدد محاسبه‌شده که بیشترین مقدار را داراست، تغییر می‌کند.

---

[1] Rael Itemset Utility

[2] Co-occurrence Utility Descending order

[3] Coverage

[4] Real 1-Itemset Relative Utility

[5] Real 2-Itemset Relative Utility

[6] Co-occurrence Utility Descending order Matrix

۲-۵-۲- مسئله مجموعه‌داده‌های پرسود موجود در قفسه

این مسئله از آن‌جایی مورد توجه قرار گرفت که در واقعیت اقلام همیشه موجود نیستند. برای مثال برخی اقلام تنها در تابستان هستند و یا برخی از اقلام تنها در زمان عید نوروز موجود هستند. درنتیجه پرداختن به این مسئله معیار عادلانه‌تری برای یافتن آیتم‌های پرسود است. برای حل این مسئله، دوره‌ی حضور هر آیتم در پایگاه داده، از قبل مشخص شده‌است. این دوره‌ها را با عدد مشخص می‌کنند. تمام تعاریفی که در مطالب قبلی برای محاسبه‌ی سود بیان شد، برای هر دوره زمانی مورد بررسی قرار می‌گیرد و مجموعه‌داده‌های سودمند بنا بر دوره‌ی زمانی که موجود هستند کشف می‌شوند. در [۱۵] به این مسئله نیز پرداخته شده‌است.

۳-۵-۲- مسئله مجموعه‌داده‌های پرسود با سود مثبت و منفی

در واقعیت ممکن است در پایگاه داده‌ی تراکنشی اقلام با سود منفی نیز وجود داشته باشند. این بدان معنی است که این اقلام زیر قیمت اصلی به فروش می‌رسند تا فروش اقلام دیگر بیشتر شود. دراین حالت مسئله‌ی HIUM باید با در نظر گرفتن هر دو نوع سود اجرا شود. در مقالات [۱۰]، [۱۱] و [۱۵] به بررسی این مسئله پرداخته شده‌است.

در ادامه روشی را معرفی می‌کنیم که هر سه مسئله‌ی بالا را مورد بررسی قرار داده‌است. الگوریتم KOSHU [۱۵] به حل مسئله‌ی استخراج مجموعه‌داده‌ی پرسود می‌پردازد در حالی که هر دو نوع سود مثبت و منفی اقلام را در نظر می‌گیرد، سود اقلام را با توجه به دوره زمانی که موجود هستند محاسبه می‌کند و برای این مسئله به جای آستانه سودمندی از تعداد K آیتم پرسود که توسط کاربر تعیین می‌شود بهره می‌برد. در ادامه به تعاریف مورد نیاز برای حل این مسئله می‌پردازیم.

❖ تعاریف مربوط به اقلام پرسود دوره‌ای

فرض کنید PE مجموعه‌ای است که شامل دوره‌هایی است که اقلام در آن‌ها به فروش می‌رسند و هر کدام با یک عدد نمایش‌داده‌می‌شوند. به هر تراکنش نیز یکی از اعداد مجموعه‌ی PE اختصاص داده می‌شود، $pt(T_d) \in PE$ ، که نشان می‌دهد هر تراکنش در چه دوره‌ی زمانی‌ای رخ داده‌است.

تعریف۱۲: دوره‌ی یک مجموعه‌داده مانند X، مجموعه‌ای از دوره‌هایی است که X در آن‌ها به فروش رسیده‌اند و به صورت $pi(X) = \{pt(T_d) \mid T_d \in D \wedge X \subseteq T_d\}$ تعریف می‌شود.

تعریف۱۳: سود یک مجموعه‌داده مانند X در یک دوره‌ی زمانی خاص مانند h از جمع سود آن مجموعه‌داده در تراکنش‌هایی به دست می‌آید که در دوره زمانی h رخ دادند.

$$u(X,h) = \sum_{T_d \in D \wedge h = pt(T_d)} u(X, T_d) \qquad (۷-۲)$$

سود یک مجموعه‌داده مانند X در پایگاه داده از رابطه (۴) به دست می‌آید.

$$u(X) = \sum_{h = pi(X)} u(X, h) \qquad (۸-۲)$$

تعریف۱۴: سود یک تراکنش مانند T_d در پایگاه داده از جمع سود آیتم‌های موجود در آن و به روش $TU(T_d) = \sum_{i \in T_d} u(i, T_d)$ به دست می‌آید.

سود کلی یک دوره‌ی زمانی برای مجموعه‌داده‌ی X از جمع سود تراکنش‌هایی به دست می‌آید که دوره‌ی زمانی آن‌ها برابر با دوره‌های زمانی است که X در آن‌ها حضور داشته‌است.

$$to(X) = \sum_{h \in pi(X) \wedge T_d \in D \wedge h = pt(T_d)} TU(T_d) \qquad (۹-۲)$$

تعریف۱۵: سود نسبی یک مجموعه‌داده مانند X برابر با $ru(X) = u(X) / |to(X)|$ است. سودمندی نسبی یک مجموعه‌داده مانند X نشان می‌دهد که چگونه سود و زیان تولید شده توسط X با سود و زیان کلی تولید شده در طول دوره‌های زمانی که X فروخته شد، مقایسه می‌شود.

مجموعه‌داده‌ی X یک HOU[1] است اگر منفعت نسبی آن (ru(X)) کم‌تر از حداقل آستانه سودمندی مشخص‌شده توسط کاربر نباشد.

همان‌طور که قبلا گفته‌شد معیار TWU معرفی‌شده برای پایگاه داده‌هایی که اقلام را تنها با سود مثبت در نظر می‌گیرند مناسب است. بنابراین برای پایگاه داده‌هایی با سود مثبت و منفی تعریف جدیدی ارائه شده‌است.

تعریف۱۶: سود تراکنشی وزنی برای مجموعه‌داده‌ای مانند X در دوره زمانی h از جمع سود تراکنش‌هایی بدست می‌آید که دوره زمانی آن‌ها h است و X در آن‌ها حضور دارد.

$$TWU(X,h) = \sum_{X \subseteq T_d \wedge T_d \in D \wedge h = pt(T_d)} TU(T_d) \qquad (۱۰-۲)$$

تعریف۱۷: سود یک دوره‌ی زمانی مانند h از جمع سود تراکنش‌هایی که دوره‌ی زمانی آن‌ها برابر با h است به دست می‌آید.

$$pto(h) = \sum_{T_d \in D \wedge h = pt(T_d)} TU(T_d) \qquad (۱۱-۲)$$

سود نسبی برای مجموعه‌داده‌ی X در دوره زمانی h برابر با $ru(X,h) = u(X) / |pto(h)|$ است.

ویژگی۲: با داشتن مجموعه‌داده X و دوره زمانی h همواره $TWU(X,h) \geq u(X,h)$ است.

ویژگی۳: اگر $X \subset Y$ آنگاه $TWU(X,h) \geq TWU(Y,h)$ خواهد بود.

---

[1] High On-Shelf Utility

ویژگی۴: همواره $TWU(X,h)/pto(h) \geq ru(X,h)$ است. یعنی عبارت سمت چپ نامساوی یک حد بالا برای سود نسبی مجموعه‌داده‌ی X در دوره زمانی h است.

ویژگی۵: با داشتن مجموعه‌داده‌ی X، اگر برای هیچ دوره‌ی زمانی عبارت $TWU(X,h)/pto(h) \geq \min util$ صادق نباشد آنگاه X یک HOU نخواهد بود. در غیر این صورت ممکن است HOU باشد یا نباشد.

- تعاریف مربوط به سود مثبت و منفی اقلام

تعریف۱۸: سود تراکنش بازتعریف‌شده (RTU[1]) باز تعریفی برای اقلام منفی است که از جمع سود آیتم‌هایی با سود مثبت به دست می‌آید.

$$RTU(T_d) = \sum_{x \in T_d \wedge p(x)>0} u(x, T_d) \qquad (۱۲-۲)$$

سود وزنی تراکنش بازتعریف‌شده (RTWU[2])، باز تعریفی از TWU است و به صورت زیر به دست می‌آید.

$$RTWU(X,h) = \sum_{T_d \in D \wedge X \subseteq T_d \wedge pt(T_d)=h} RTU(T_d) \qquad (۱۳-۲)$$

فرض کنید up(X) مجموعه‌ی آیتم‌هایی از X باشد که سود مثبت دارند و un(X) مجموعه‌ای از آیتم‌هایی با سود منفی باشند که در X وجود دارند.

ویژگی۶: با داشتن مجموعه‌داده‌ی X، خواهیم داشت که $u(X,h) \leq u(up(X),h)$.

[1] Redefined Transaction Utility

[2] Redefined Transaction Weighted Utility

ویژگی۷: فرض کنیم که X یک مجموعه‌داده و z یک آیتم با سود منفی باشد که عضو X نیست، در این صورت $u(up(X \cup \{z\}),h) \le u(up(X),h)$ خواهد بود.

ویژگی۸: فرض کنید Y بسطی منفی از X باشد، آنگاه $u(up(Y),h) \le u(up(X),h)$ خواهد بود.

ویژگی۹: اگر بر اساس ترتیب $\succ$ تنها اقلام منفی بتوانند با X ترکیب شوند و هیچ دوره‌ای وجود نداشته باشد که $u(up(X),h)/pto(h) \ge \min util$، آنگاه X و هر بسطی از آن HOU نخواهد بود.

❖ راهبردهای افزایش آستانه سودمندی

راهبردهای معرفی‌شده مربوط به الگوریتم KOSHU هستند که در [۱۸] آمده‌است.

۱- راهبرد افزایش آستانه سودمندی با استفاده از مقدار سودمندی نسبی تک‌آیتم‌ها

اولین روش، راهبرد افزایش آستانه سودمندی با استفاده از مقدار سودمندی نسبی تک‌آیتم‌ها (RIRU[1]) نام دارد. در این روش که بعد از اولین اسکن پایگاه داده اجرا می‌شود، مقدار سود نسبی تک‌آیتم‌ها حساب می‌شود. سپس بر اساس این راهبرد، مقدار آستانه سودمندی که برابر صفر بود به مقدار kامین سود نسبی محاسبه شده تغییر می‌کند.

۲- راهبرد افزایش آستانه سودمندی با استفاده از مقدار سودمندی نسبی آیتم‌های دوتایی

دومین روش، راهبرد افزایش آستانه سودمندی با استفاده از مقدار سودمندی نسبی تک‌آیتم‌ها (RIRU2[2]) نام دارد. در این روش که بعد از اسکن دوم انجام می‌شود، سود نسبی جفت آیتم‌ها بر اساس تعریف۵ حساب می‌شود. سپس مقدار آستانه سودمندی از مقداری که با روش RIRU تعیین شده‌بود به مقدار kامین سود نسبی بالای محاسبه‌شده در این مرحله افزایش می‌یابد.

---

[1] Real 1-Itemset Relative Utility

[2] Real 2-Itemset Relative Utility

۴-۵-۲- مسئله مجموعه‌داده‌های متناوب پرسود

در مسئله‌ی HUIM پایه مجموعه‌داده‌های پرسودی که استخراج می‌شوند به پایگاه داده‌ای تعلق دارند که برای مدت طولانی اطلاعات در آن جمع‌آوری شده‌است. برای مثال برای یک فروشگاه اینترنتی مثل دیجی‌کالا، جواهرات می‌توانند به عنوان خروجی مسئله‌ی HUIM پایه استخراج شوند. این درحالی است که خرید جواهرات به ندرت اتفاق می‌افتد. اقلام دیگری که به طور متناوب خریداری می‌شوند می‌توانند به عنوان اقلام پرسود در دوره تناوب خودشان محسوب شوند. برای مثال از بین اقلامی که هفتگی خریداری می‌شوند، لبنیات می‌تواند جز اقلام پرسود این دوره باشد. این مسئله می‌تواند رفتار مشتریان را بهتر تحلیل کند. برای حل این مسئله نیاز است علاوه‌بر مقدار آستانه سودمندی، مقدار کمینه و بیشینه‌ی دوره تناوب و همچنین مقدار کمینه و بیشینه‌ی میانگین تناوب توسط کاربر تعیین شود. با این شرایط برای هرس فضای جست‌وجو باید این مقادیر نیز مورد توجه قرار گیرند. مقالاتی که به این موضوع پرداخته‌اند، [۱۳] و [۱۹] هستند. در ادامه، تعاریف مربوط به این مسئله آمده‌است. برای یکپارچگی در تعریف مسئله تعاریف ارائه شده در این بخش مربوط به [۱۳] است.

❖ الگوهای متناوب تکراری

تعریف۱۹: (تناوب یک مجموعه‌داده). فرض کنید X یک مجموعه‌داده در پایگاه داده‌ای‌ست که شامل n تراکنش است. $g(X)=\{T_{g_1},T_{g_2},...,T_{g_k}\}$ مجموعه‌ی تراکنش‌هایی است که X در آن‌ها حضور دارد و رابطه‌ی $1\le g_1<g_2<...<g_k\le n$ برقرار است. مقدار تناوب برای دو تراکنش متوالی با شماره شناسایی x و y در رابطه با X به صورت $pe(T_x,T_y)=(y-x)$ به‌دست می‌آید که برابر با تعداد تراکنش‌هایی است که بین این دو تراکنش وجود دارد. تناوب یک مجموعه‌داده مانند X، لیستی از مقدار تناوب برای هر تراکنشی است که در مجموعه‌ی g(X) وجود دارند. و به‌صورت $ps(X)=\cup_{1\le z\le k+1}(g_z-g_{z-1})$ به‌دست می‌آید.

تعریف۲۰: (الگوهای متناوب تکراری). بیشینه‌ی تناوب یک مجموعه‌داده‌ای مانند X به صورت maxper(X)=max(ps(X)) تعریف می‌شود. یک مجموعه‌داده زمانی PFP[1] محسوب می‌شود که $|g(X)| \geq \min\sup$ و $\max per(X) < \max Per$ باشد. مقادیر minsup و maxPer توسط کاربر تعریف می‌شوند.

یکی از مهم‌ترین محدودیت‌ها در الگویتم‌های کاوش الگوهای تکراری متناوب این است که تنها به تعداد دفعات تکرار توجه می‌کنند و سود حاصل از آن‌ها (PFP) مهم نیست.

❖ الگوهای پرسود متناوب

در این بخش معیارهایی که برای پیدا کردن مجموعه‌داده‌های پرسود متناوب استفاده می‌شود، معرفی خواهدشد.

معیار بیشینه‌ی تناوب در بسیاری از الگوریتم‌های PFP استفاده شده‌است. با این معیار اگر مجموعه-داده‌ای، مجموعه تناوب‌اش شامل یک تناوب واحد باشد که از مقدار maxPer بیشتر باشد کل آن مجموعه‌داده نادیده گرفته‌می‌شود. برای رفع این مشکل از معیار میانگین تناوب استفاده می‌شود.

تعریف۲۱: (میانگین تناوب برای یک مجموعه‌داده). این معیار از رابطه (۱۰) به‌دست می‌آید.

$$avgper(X) = \sum_{g_i \in ps(X)} g_i / |ps(X)| \qquad (۱۴-۲)$$

ویژگی۱۰: (رابطه‌ی بین میانگین تناوب و مقدار پشتیبانی). فرض کنید X مجموعه‌داده‌ای در پایگاه داده‌ی D باشد. راه دیگری برای محاسبه میانگین تناوب به صورت (۱۱) است.

$$avgper(X) = |D| / (|g(X)| + 1) \qquad (۱۵-۲)$$

[1] Periodic Frequent Pattern

این روش به این دلیل مهم است که اندازه D تنها یک‌بار محاسبه می‌شود و میانگین تناوب هر مجموعه‌داده تنها با محاسبه |g(X)+1| به‌دست می‌آید. علاوه بر این، این قیاس مهم است چون نشان می‌دهد که یک رابطه بین پشتیبانی مورد استفاده در FIM و تناوب متوسط یک الگو وجود دارد. اگر چه میانگین تناوب مفید است، نباید به عنوان تنها معیار برای ارزیابی تناوب‌پذیری یک الگو استفاده شود زیرا در نظر نمی‌گیرد که آیا یک مجموعه دارای دوره‌های متفاوت با طول زیاد است یا خیر. بنابراین این مجموعه‌داده نباید یک مجموعه‌ی متناوب باشد. برای اجتناب از یافتن الگوهایی با دوره‌های متفاوت، راه‌حل ترکیب اندازه تناوب متوسط با دیگر معیارها را درپیش می‌گیرند. یکی از معیارهایی که استفاده می‌شود معیار کمینه‌ی تناوب است که به صورت minper(X)=min(ps(X)) تعریف می‌شود. این معیار برای جلوگیری از مجموعه‌داده‌هایی که تناوب کوتاهی دارند به‌کار گرفته‌می‌شود. اما تناوب‌هایی که برابر با ۰ و ۱ باشند مثل آن‌هایی که شامل تراکنش اول و آخر باشند را در نظر نمی‌گیرد. بنابراین راه حل این است که تناوب‌های اول و آخر از لیست تناوب هر مجموعه‌داده را درنظر نگیریم. اگر با این کار لیست تناوب خالی باشد آن را ∞ در نظر می‌گیریم. منطق استفاده از این مقیاس در ترکیب با تناوب متوسط این است که می‌تواند از کشف الگوهای دوره‌ای که برای دوره‌های طولانی رخ نمی‌دهند جلوگیری کند. دلیل استفاده از این سه معیار این است که از نظر محاسبه و مصرف حافظه کم‌هزینه هستند.

تعریف۲۲: (مجموعه‌داده‌های پرسود متناوب). با فرض این‌که مقادیر مثبت minutil، minAvg، maxAvg، minPer و maxPer را کاربر تعیین کند، مجموعه‌داده X پرسود است اگر $\min Avg \le avgper(X) \le \max Avg$ و $\min per(X) \ge \min Per$ و $\max per(X) \le \max Per$ و در نهایت $u(X) \ge \min util$ باشد.

با توجه به تعاریف ارائه شده در بالا می‌توان مجموعه‌داده‌های پرسود متناوب را استخراج نمود.

2-6- جمع‌بندی

در این فصل به شرح مفصل مسئله‌ی HUIM و بسط‌های آن پرداختیم و نحوه‌ی محاسبه‌ی سود و یافتن الگوی مناسب را بیان کردیم.

در نهایت لازم به ذکر است که برای مسئله‌ی HUIM پایه بسط‌های مختلف دیگری ارائه شده‌است که هر یک برای حل چالشی مناسب هستند. برای مثال مسئله‌ی HUIM را می‌توان با در نظر گرفتن داده‌های درجریان مورد بررسی قرار داد. مسئله‌ی مجموعه‌داده‌ی پرسود بسته مجموعه‌داده‌ای را استخراج می‌کند که پرسود است و از روی آن می‌توان مجموعه‌داده‌های پرسود دیگر را استخراج کرد. این روش باعث می‌شود تا زمان استخراج مجموعه‌داده‌های پرسود کمتر شود زیرا تعداد الگوهای بسته کمتر است و یا حتی می‌توان HUIM را با توجه به استراتژی‌های تخفیف مختلف حل کرد.

فصل سوم

نتیجه‌گیری و جمع‌بندی مطالب

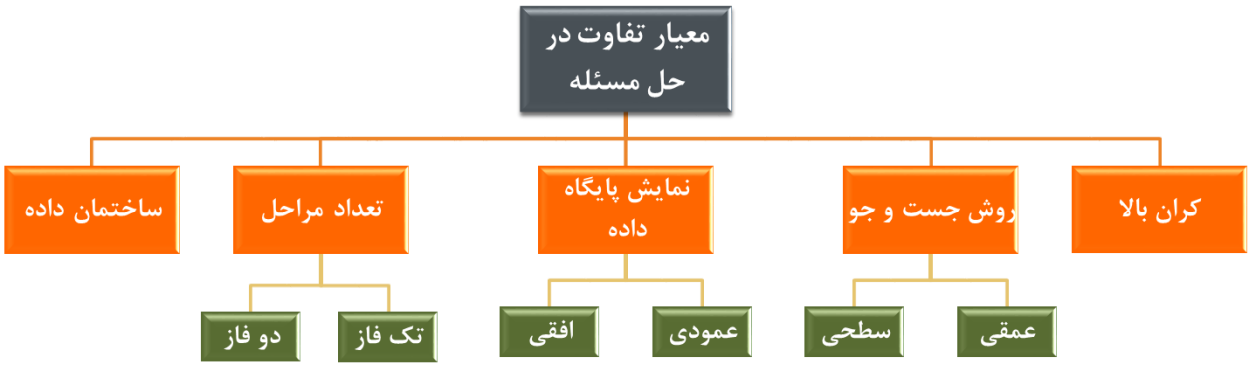
3-1- مقدمه

حل مسئله‌ی HUIM نسبت به مسئله‌ی FIM سخت‌تر و چالش‌برانگیزتر است. زیرا در مسئله‌ی FIM از ویژگی غیریکنواخت بودن پشتیبانی استفاده می‌کنند و فضای جست‌وجو را به‌طور موثر هرس می‌کنند. اما معیار سود در HUIM نه یکنواخت است و نه غیریکنواخت. به عبارتی سود مجموعه‌داده‌ای می‌تواند کمتر، مساوی و یا بیشتر از ابرمجموعه‌اش باشد در حالی که مقدار پشتیبانی یک مجموعه‌داده همواره از زیرمجموعه‌اش کمتر و یا با آن برابراست. همین امر سبب می‌شود تا برای هرس فضای جست‌وجو نیازمند به راهبردهای مختلفی باشیم تا کاهش به صورت موثری انجام شود. بنابراین پیچیدگی در حل این مسئله سبب می‌شود تا زمان اجرا و مصرف حافظه از مسئله‌ی FIM بیشتر باشد.

روش‌های ارائه‌شده برای انواع مختلف مسئله‌ی HUIM از راهبردهای مختلف هرس، ساختمان داده‌های متفاوت و روش‌های مختلفی برای جست‌وجوی فضای حالت استفاده می‌کنند که همه‌ی این‌ها سبب می‌شود تا میزان مصرف حافظه و زمان اجرای این روش‌ها با هم متفاوت باشد.

3-2- علت تفاوت روش‌ها

همان‌طور که قبلا اشاره شد روش‌ها ممکن است به دلیل نوع نمایش پایگاه داده، تعداد مراحل انجام کار، روش جست‌وجو و راهبردهای هرس مورد استفاده در زمان اجرا و میزان حافظه‌ی مصرفی با هم متفاوت باشند. دلایل تفاوت روش‌ها در شکل(۳-۱) نشان داده‌می‌شود.



شکل۳-۱: معیار تفاوت روش‌های حل

اگر از پایگاه داده‌ی تراکنشی و یا پایگاه داده‌ی واکشی‌شده استفاده کنیم، نمایش پایگاه داده به صورت افقی خواهد بود ولی اگر از لیست سودمندی استفاده کنیم از پایگاه داده‌ی عمودی استفاده کرده‌ایم. استفاده از روش جست‌وجوی عمقی نیز از روش جست‌وجوی سطحی بهتر است. زیرا تنها مجموعه‌داده‌ای را بسط می‌دهد که در پایگاه داده موجود است. بنابراین میزان حافظه کمتری استفاده می‌کند. اما در جست‌وجوی سطحی مجموعه‌داده‌ها را بدون در نظر گرفتن این‌که در پایگاه داده موجود است یا نه بسط می‌دهد. این امر سبب می‌شود تا فضای حالت بزرگتری داشته‌باشیم که عملا بسیاری از داده‌هایش مورد بررسی قرار نخواهند گرفت. در زیر الگوریتم‌های پایه‌ی مسئله‌ی HUIM مورد بررسی قرار گرفته‌اند.

جدول۳-۱: دسته‌بندی روش‌های پایه

الگوریتم	سال	روش جست‌وجو	تعداد مراحل	نمایش پایگاه داده	الگوریتم پایه
Tow phase[۴]	۲۰۰۵	سطحی	دو فاز	افقی	Apriori
PB [۲۰]	۲۰۱۴	سطحی	دو فاز	افقی	Apriori
IHUP [۲۱]	۲۰۰۹	عمق اول	دو فاز	افقی (درخت پیشوندی)	FPGrowth
UPGrowth [۲۲]	۲۰۱۳	عمق اول	دو فاز	افقی (درخت پیشوندی)	FPGrowth
HUPGrowth [۲۳]	۲۰۱۱	عمق اول	دو فاز	افقی (درخت پیشوندی)	FPGrowth
HUIMiner [۶]	۲۰۱۲	عمق اول	تک فاز	عمودی(لیست سودمندی)	Eclat
FHM [۲۴]	۲۰۱۴	عمق اول	تک فاز	عمودی(لیست سودمندی)	Eclat
HUIMiner* [۷]	۲۰۱۸	عمق اول	تک فاز	عمودی(لیست سودمندی*)	Eclat
ULBMiner [۸]	۲۰۱۷	عمق اول	تک فاز	عمودی(لیست سودمندی)	Eclat
EFIM [۵]	۲۰۱۵	عمق اول	تک فاز	افقی(پایگاه داده واکشی شده)	LCM

3-3- مقایسه روش‌های تک‌فاز و دوفاز

از آن‌جایی که روش‌های دوفاز از راه حل‌های اولیه به‌شمار می‌روند درنتیجه کاستی‌هایی نیز داشته‌اند. در زیر ویژگی‌های این دو روش را برخواهیم شمرد.

۱-۳-۳- ویژگی روش‌های دوفاز

- ❖ مزایا
  - سادگی در فهم و پیاده‌سازی
- ❖ معایب
  - مصرف زیاد حافظه به دلیل زیاد بودن تعداد کاندیدهای تولیدشده
  - بالا بودن تعداد اسکن پایگاه داده برای محاسبه سود هر مجموعه‌داده
  - استفاده از راهبردهای نه‌چندان دقیق برای هرس فضای‌جست‌وجو
  - بالا بودن زمان اجرا به دلیل اسکن‌های پی‌درپی پایگاه داده

۲-۳-۳- ویژگی روش‌های تک‌فاز

- ❖ مزایا
  - استفاده از ساختمان‌داده‌های جدید مانند لیست سودمندی
  - کاهش دفعات اسکن پایگاه داده به‌دلیل استفاده از لیست سودمندی و یا پایگاه داده‌ی واکشی‌شده
  - کاهش میزان مصرف حافظه نسبت به روش‌های دوفاز به دلیل عدم تولید مجموعه‌ی کاندید
  - کارآمدتر بودن از نظر زمان اجرا نسبت به روش‌های دوفاز به دلیل عدم اسکن پی-درپی پایگاه داده
  - کران‌های بالای دقیق‌تر
- ❖ معایب
  - بالا بودن هزینه ادغام در روش‌های مبتنی بر لیست سودمندی
  - پیچیدگی در پیاده‌سازی

3-4- جمع‌بندی

در این بخش قصد داریم تا راه‌کارهای مورد استفاده را براساس روش‌های مورد استفاده‌ی آن‌ها، دسته‌بندی نماییم. در جدول(۳-۲) به‌طور خلاصه راه‌کارهای ارائه‌شده به همراه نوع مسئله‌ی تحت پوشش و تحلیل مختصری از آن آورده شده‌است.

جدول۳-۲: خلاصه‌ای از راه‌کارهای ارائه‌شده

الگوریتم	سال انتشار	سود درنظر گرفته‌شده	نوع الگوریتم	ساختار داده‌ی مورد استفاده	راهبردهای هرس	تحلیل
HMiner_closed [۹]	۲۰۱۹	مثبت	تک فاز	لیست سودندی فشرده	LA-prune و C-prune	کاهش زمان و حافظه با استفاده از فشرده-سازی لیست و همچنین پرداختن به مسئله-ی مجموعه‌داده‌های پرسود بسته برای کاهش تعداد HUIها
MHAUIPNU [۱۰]	۲۰۱۹	مثبت و منفی	تک فاز	لیست سودمندی	Tubpn، بر اساس آیتم منفی و بر اساس بسط منفی	ارائه‌ی راهبردهای هرس دقیق برای کاهش فضای جست‌وجو
ULB-Miner [۸]	۲۰۱۷	مثبت	تک فاز	لیست سودمندی بافرشده	EUCP	بهبود مصرف حافظه و زمان با استفاده از لیست بافر شده و بهبود عملکرد الگوریتم-های قبلی با استفاده از این روش
ABSA [۱۱]	۲۰۱۹	مثبت و منفی	تک فاز	لیست سودمندی	ساختار EUCS	شاخه‌ای از متن کاوی که با استفاده از تحلیل نظر کاربران ویژگی‌های برتر یک محصول را استخراج می‌کند. هر ویژگی از قبل ارزش-گذاری می‌شود.
FUP-HAUIMD [۱۲]	۲۰۱۸	مثبت	تک فاز	لیست سودمندی AU-List	auub	محاسبه‌ی HAUI در پایگاه داده‌ی پویا با در نظر گرفتن حذف تراکنش‌ها. به دلیل پویا بودن نتوانست کران بالای دقیق‌تری برای کاهش فضا استفاده کند.
PHM [۱۳]	۲۰۱۶	مثبت	تک فاز	لیست سودمندی	maxPer، avgPer و EUCP	تحلیل رفتار مشتری بر اساس دوره‌های خرید او، بهره‌گیری از یک روش کارامد مانند FHM
PEFIM [۱۶]	۲۰۱۹	مثبت	تک فاز	آرایه سودمندی	Lc و su	واکشی پایگاه داده، ادغام تراکنش‌ها وکم کردن هزینه ادغام و حافظه در لیست سودمندی و موازی‌سازی برای کاهش زمان اجرا

MHUI [۱۴]	۲۰۱۷	مثبت	تک فاز	لیست سودمندی	TWU-M، U-M، EUCS-M و LA-M	استفاده از چند آستانه سودمندی برای کاوش مجموعه‌داده‌های سودمند
KOSHU [۱۵]	۲۰۱۶	مثبت و منفی	تک فاز	لیست سودمندی	PUP، EMPRP، CE2P	پرداختن به دو مسئله‌ی دوره‌ای بودن اقلام و k مجموعه‌ی برتر
TKEH [۱۷]	۲۰۱۸	مثبت	تک فاز	آرایه سودمندی و صف اولویت	Tm، sup	پرداختن به مسئله k مجموعه‌ی برتر و استفاده مفید از روش‌های قبلی و ارائه کران-های بالای دقیق
$SPHUI_{TP}$ [۱۹]	۲۰۱۷	مثبت	دو فاز	لیست	TWU و MP	پرداختن به مسئله اقلام دوره‌ی زمانی کوتاه اما پرهزینه در مصرف زمان و حافظه
EFIM [۵]	۲۰۱۶	مثبت	تک فاز	آرایه سودمندی	Lc و su	واکشی پایگاه داده، ادغام تراکنش‌ها وکم کردن هزینه ادغام و حافظه در لیست سودمندی
HUI-Miner [۶]	۲۰۱۲	مثبت	تک فاز	لیست سودمندی	Ru و TWU	بسیار سریع بودن در مقابل الگوریتم‌های دوفاز
HUI-Miner* [۷]	۲۰۱۹	مثبت	دو فاز	لیست سودمندی	Ru و TWU	کم کردن زمان مقایسه تراکنش‌های مشترک هنگام ادغام دو لیست و موثر برای داده‌های پراکنده
LUIM [۲۵]	۲۰۱۹	مثبت	دو فاز	آرایه	TWU و NUL	پرداختن به مسئله‌ی مجموعه‌داده‌های کم-سود- بالا بودن مصرف حافظه و زمان

5-3- نتیجه‌گیری

به‌طور کلی روش‌هایی که تک‌فاز هستند از نظر سرعت و حافظه بهتر از روش‌های دوفاز عمل می‌کنند. اما درهمین راستا نیز اگر روشی از لیست سودمندی استفاده کند هزینه ادغام لیست‌ها خودیک چالش است که هنوز جای کار دارد. برای بهبود این کاستی از پایگاه‌های داده‌ی واکشی‌شده، لیست‌های سودمندی بافرشده و یا لیست سودمندی فشرده‌شده استفاده شده‌است. برای کاهش فضای حالت نیز استفاده از راهبردهای دقیق‌تر پیشنهاد می‌شود که از محاسبه‌سود دقیق اقلام بهره‌مند شوند.

فهرست منابع و مراجع

[۱] تقوی، پروین؛ فاطمه، اکبرپور. ۱۳۹۵. طراحی و پیاده سازی سیستم تحلیل و بررسی میزان کتاب‌خوانی در ایران. پایان‌نامه کارشناسی، دانشگاه تربیت دبیر شهید رجایی.

[2] Fournier-Viger, Philippe, et al. "A survey of high utility itemset mining." *High-Utility Pattern Mining*. Springer, Cham, 2019. 1-45.

[۳] مدیر اموزش فرادرس. ۱۳۹۲. فیلم آموزشی جامع کاوش قواعد وابستگی. متلب سایت. https://matlabsite.com/933/mvrdm9206ij-association-rule-mining-in-data-mining.html (دسترسی در ۱۳۹۹/۳/۱۵)

[4] Liu, Ying, Wei-keng Liao, and Alok Choudhary. "A two-phase algorithm for fast discovery of high utility itemsets." Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, Berlin, Heidelberg, 2005.

[5] Zida, S., Fournier-Viger, P., Lin, J.C.-W., Wu, C.W., Tseng, V.S.: EFIM: a highly efficient algorithm for high-utility itemset mining. In: Proceedings of the 14th Mexican International
Conference Artificial Intelligence, pp. 530–546. Springer (2015)

[6] Liu, Mengchi, and Junfeng Qu. "Mining high utility itemsets without candidate generation." *Proceedings of the 21st ACM international conference on Information and knowledge management*. 2012.

[7] Qu, Jun-Feng, Mengchi Liu, and Philippe Fournier-Viger. "Efficient algorithms for high utility itemset mining without candidate generation." *High-Utility Pattern Mining*. Springer, Cham, 2019. 131-160.

[8] Duong, Quang-Huy, et al. "Efficient high utility itemset mining using buffered utility-lists." Applied Intelligence 48.7 (2018): 1859-1877.

[9] Nguyen, Loan TT, et al. "An efficient method for mining high utility closed itemsets." *Information Sciences* 495 (2019): 78-99.

[10] Yildirim, Irfan, and Mete Celik. "Mining High-Average Utility Itemsets with Positive and Negative External Utilities." ***New Generation Computing*** **(2019): 1-34.**

[11] Demir, Seyfullah, et al. "Extracting Potentially High Profit Product Feature Groups by Using High Utility Pattern Mining and Aspect Based Sentiment Analysis." ***High-Utility Pattern Mining*****. Springer, Cham, 2019. 233-260.**

[12] Lin, Jerry Chun-Wei, et al. "Maintenance algorithm for high average-utility itemsets with transaction deletion." ***Applied Intelligence*** **48.10 (2018): 3691-3706.**

[13] Fournier-Viger, Philippe, et al. "PHM: mining periodic high-utility itemsets." ***Industrial conference on data mining*****. Springer, Cham, 2016.**

[14] Krishnamoorthy, Srikumar. "Efficient mining of high utility itemsets with multiple minimum utility thresholds." ***Engineering Applications of Artificial Intelligence*** **69 (2018): 112-126.**

[15] Dam, Thu-Lan, et al. "An efficient algorithm for mining top-k on-shelf high utility itemsets." ***Knowledge and Information Systems*** **52.3 (2017): 621-655.**

[16] Nguyen, Trinh DD, Loan TT Nguyen, and Bay Vo. "A Parallel Algorithm for Mining High Utility Itemsets." ***International Conference on Information Systems Architecture and Technology*****. Springer, Cham, 2018.**

[17] Singh, Kuldeep, et al. "TKEH: an efficient algorithm for mining top-k high utility itemsets." ***Applied Intelligence*** **49.3 (2019): 1078-1097.**

[18] Krishnamoorthy, Srikumar. "Mining top-k high utility itemsets with effective threshold raising strategies." Expert Systems with Applications 117 (2019): 148-165.

[19] Lin, Jerry Chun-Wei, et al. "A two-phase approach to mine short-period high-utility itemsets in transactional databases." ***Advanced Engineering Informatics*** **33 (2017): 29-43.**

[20] Lan, Guo-Cheng, Tzung-Pei Hong, and Vincent S. Tseng. "An efficient projection-based indexing approach for mining high utility itemsets." Knowledge and information systems 38.1 (2014): 85-107.

[21] Ahmed, Chowdhury Farhan, et al. "Efficient tree structures for high utility pattern mining in incremental databases." ***IEEE Transactions on Knowledge and Data Engineering*** **21.12 (2009): 1708-1721.**

[22] Tseng, Vincent S., et al. "Efficient algorithms for mining high utility itemsets from transactional databases." ***IEEE transactions on knowledge and data engineering*** **25.8 (2012): 1772-1786.**

[23] Lin, Chun-Wei, Tzung-Pei Hong, and Wen-Hsiang Lu. "An effective tree structure for mining high utility itemsets." ***Expert Systems with Applications*** **38.6 (2011): 7419-7424.**

[24] Fournier-Viger, Philippe, et al. "FHM: Faster high-utility itemset mining using estimated utility co-occurrence pruning." ***International symposium on methodologies for intelligent systems*****. Springer, Cham, 2014.**

[25] Alhusaini, Naji, et al. "LUIM: New Low-Utility Itemset Mining Framework." IEEE Access 7 (2019): 100535-100551.

[26] Krishnamoorthy, Srikumar. "A comparative study of top-k high utility itemset mining methods." ***High-Utility Pattern Mining*****. Springer, Cham, 2019. 47-74.**

Abstract

With the explosive growth of data and information, the need for strategies and tools to convert data into useful knowledge is increased. In the most of the past literatures, although they find meaningful information, but they do not care about the utility of the items. So having information about the high utility itemsets can be useful to make right decisions. To satisfy this need, in this literature we investigate some solutions to mine the itemsets that have utilities more than a utility threshold.

Keywords: Data mining, high utility itemsets, frequent itemset, minimum utility

Shahid Rajaee Teacher Training University
Faculty of Computer Engineering
Department of Software

Title:
High Utility Itemset Mining

By:

Parvin Taghavi

Supervisor:

Dr. Negin Daneshpour

June 2020