وزارت علوم، تحقیقات و فناوری
موسسه آموزش عالی جهاد دانشگاهی
استان اصفهان

کنفرانس ملی فناوری های نوین در
مهندسی برق و کامپیوتر

۲۷ دی ۱۳۹۶

کنفرانس ملی فناوری های نوین در
مهندسی برق و کامپیوتر

مروری بر الگوریتم‌های متمرکز و توزیع‌شده در استخراج الگوهای پرتکرار

مهدی زمانی نوکابادی[1]، حمید رستگاری[2*]، مهدی شریفی[3]

1- **دانشکده مهندسی کامپیوتر، واحد نجف آباد، دانشگاه آزاد اسلامی، نجف آباد، ایران.**

Mehdi.zamani@sco.iaun.ac.ir

2- **دانشکده مهندسی کامپیوتر، واحد نجف آباد، دانشگاه آزاد اسلامی، نجف آباد، ایران.**

rastegari@iaun.ac.ir

3- **دانشکده مهندسی کامپیوتر، واحد نجف آباد، دانشگاه آزاد اسلامی، نجف آباد، ایران.**

m.sharifi@pco.iaun.ac.ir

چکیده

امروزه پایگاه داده‌های بزرگ، در اندازه‌های مختلف از ترابایت به پتابایت در حال ایجاد هستند. علاوه بر روند صعودی در اندازه‌ی پایگاه داده‌ها، استخراج اطلاعات دشوار می‌شود و منجر به مشکلاتی می‌شود. داده کاوی برای استخراج داده‌های مفید از مجموعه داده‌های بزرگ و استخراج الگوهای معنی دار از آن‌ها استفاده می‌شود. بسیاری از سازمان‌ها و همچنین بازار، اکنون تکنیک‌های داده کاوی را یک رویکرد مفید برای پیش برد کارهایشان می‌دانند. استخراج الگوهای پرتکرار به یک تکنیک مهم داده کاوی و یک تمرکز برای تحقیق تبدیل شده است. الگوهای پرتکرار الگوهایی هستند که اغلب در یک مجموعه داده ظاهر می‌شوند. برای بهبود عملکرد الگوریتم‌های استخراج الگوهای پرتکرار، روش‌های مختلفی پیشنهاد شده است. بعضی ازین روش‌ها متمرکز و بعضی دیگر توزیع‌شده اند در این مقاله، ما یک مرور کلی بر روی چند الگوریتم متمرکز وتوزیع‌شده موجود ارائه می‌دهیم.

کلیدواژه‌ها: قوانین انجمنی، الگوهای پرتکرار، الگوریتم‌های متمرکز، الگوریتم‌های توزیع‌شده، داده کاوی

1- مقدمه

در سال‌های اخیر، تحول گسترده‌ای در فن‌آوری و علم رخ داده که منجر به رشد ناگهانی اندازه‌ی داده‌ها شده است. چنین اطلاعات عظیمی نه تنها از نظر مقیاس بزرگ هستند، بلکه دارای طبیعت بدون ساختار یا نیمه ساختار یافته می‌باشند. این داده‌ها، به نام "مه داده "شناخته می‌شوند. برای توصیف مه داده، از حجم، سرعت و تنوع استفاده می‌شود، که به اختصار 3V نامیده می‌شود [1]. واژه‌ی حجم اندازه‌ی مجموعه داده است، سرعت نشان دهنده‌ی سرعت تولید داده‌ها و تنوع طیف وسیعی از انواع داده‌ها و منابع را توصیف می‌کند. کشف اطلاعات ناشناخته و مفید از داده‌های خام، داده کاوی [2] تعریف می‌شود. استخراج الگوهای پرتکرار، کشف قوانین انجمنی، طبقه‌بندی و خوشه بندی، تکنیک‌های داده کاوی هستند که به طور گسترده‌ای مورد استفاده قرار می‌گیرند. یکی از تکنیک‌های داده کاوی استخراج الگوهای پرتکرار می‌باشد. چالش‌های بسیاری برای استخراج الگوهای پرتکرار از نظر نیاز به منابع سیستم، سرعت الگوریتم، زمان اجرا و غیره وجود دارند. در این مقاله ما چند نمونه از تکنیک‌ها و الگوریتم‌های کاربردی متمرکز و توزیع‌شده را از نظر توانایی در مورد بهبود این چالش‌ها مرور می‌کنیم.

2- الگوریتم‌های اجرا شده‌ی متمرکز

این الگوریتم‌ها بر روی یک سیستم اجرا شده اند و هدف آن‌ها بهبود برخی از مشکلات مربوط به اجرای این دسته ازالگوریتم‌ها از قبیل نیاز فراوان به منابع سیستم، افزایش فضای جستجو، زمان استخراج، هزینه اسکن پایگاه داده می‌باشد.

استخراج الگوهای پرتکرار اولین قدم برای استخراج قوانین انجمنی است. با این حال، تعداد الگوهای پرتکرار به دست آمده معمولا بسیار بزرگ است و اکثر آن‌ها می‌توانند نا کارامد و زائد باشند. چندین روش برای مدلهای استخراج الگوهای حداکثری (MFP) پیشنهاد شده است اما با توجه به پیچیدگی‌های زیاد، این روش‌ها نیاز فراوانی به منابع سیستم دارند. بنابراین، الگوریتم موثری بنامINLA-MFP [3] برای استخراج MFP[1] پیشنهاد شد که از ساختار N-list و تکنیک هرس استفاده می‌کند. تکنیک هرس، بر اساس ساختار N-list، فضای جستجو را کاهش می‌دهد. سپس این الگوریتم از نظر عملکرد و کارایی با 2الگوریتم

[1] maximal frequent patterns

بنام‌های dGenMax[2] وTDM-MFI مقایسه شد که نتایج تجربی نشان داد که INLA-MFP در بیشتر موارد سریعتر اجرا می‌شود و در مقایسه با dGenMax و TDM-MFI از حافظه کمتری استفاده می‌کند.

+PrePost [4]، یک الگوریتم با کارایی بالا برای مجموعه آیتم‌های پرتکرار است که از N-list برای نشان دادن مجموعه آیتم‌ها استفاده می‌کند و به طور مستقیم با استفاده از یک درخت جستجوی set-enumeration، مجموعه‌های پرتکرار را کشف می‌کند. این الگوریتم یک استراتژی هرس کردن کارآمد به نام همسان سازی فرزندان-والدین[3] را به کار می‌برد که به شدت باعث کاهش فضای جستجو می‌شود. +PrePost با سه الگوریتم پیشرفته‌ی PrePost، FIN و* FP-growth مقایسه شد و نتایج نشان داد که +PrePost همیشه سریع‌ترین الگوریتم است. همچنین +PrePost عملکرد خوبی را از لحاظ مصرف حافظه نشان می‌دهد، زیرا تنها به میزان کمی بیشتر از حافظه نسبت به * FP-growth و حافظه کمتر از PrePost و FIN استفاده می‌کند.

الگوریتمiNTK [5] یک نسخه بهبود یافته از الگوریتم NTK است. الگوریتمNTK برای استخراج top-rank-k الگوی پر تکرار بر اساس ایده PPC-tree[4] پیشنهاد شده است. NTK به دلیل ارائه الگوهای خود بر اساس ساختار لیست گره[5] کارآمد است. الگوریتمiNTK از یک ساختار N-list برای نمایش الگوها و برای سرعت بخشیدن به فرآیند استخراج top-rank-k الگوهای پرتکرار از مفهوم رده بندی کردن استفاده می‌کند. این روش از نظر زمان استخراج و استفاده از منابع سیستم با الگوریتم NTK مقایسه شده است و نتایج نشان می‌دهد که iNTK کارآمدتر و سریع‌تر از NTK است. آن‌ها همچنین یک روش مبتنی بر CLA[6] به نام CLA-Mining [6] برای استخراج مجموعه آیتم‌های پرتکرار ارائه کرده اند. آن‌ها تکنیکی را ایجاد کردند که بتواند پایگاه داده‌های تراکنشی را آنلاین پردازش و الگوهای پرتکرار را پیدا کنند. روش پیشنهادی آن‌ها با روش Apriori، FP-Growth و BitTable مقایسه شده است و در نهایت نتیجه‌گیری شد که استخراج مجموعه آیتم‌های پرتکرار با این تکنیک، در زمان کمتری انجام می‌شود.

به طور کلی، اکثر مطالعات بر روی توسعه الگوریتم‌های کارآمد برای استخراج الگوهای پرتکرار در پایگاه داده‌های تراکنشی تمرکز می‌کنند. این رویکردها از یک محدودیت مهم رنج می‌برند، آن‌ها باید از یک

[2] GenMax algorithm that uses diffset strategy
[3] Children–Parent
[4] the Pre-order and Post-order Code tree
[5] Node-list
[6] Cellular Learning Automata

آستانه حداقل پشتیبانی به عنوان معیاری برای تعیین مجموعه الگوهای پرتکرار استفاده کنند. استفاده از یک مقدار آستانه پشتیبانی برای تشخیص فراوانی رخداد همه آیتم‌ها در یک پایگاه داده کافی نیست، زیرا هر آیتم متفاوت است و نباید آن‌ها را با یکدیگر مقایسه کرد. برای حل این مشکل الگوریتم FP-ME [7] معرفی شد، که به جای یک آستانه حداقل پشتیبانی یکنواخت برای همه موارد، هر آیتم دارای معیار آستانه منحصر به فرد پشتیبانی خود باشد. این قابلیت کاربرد استخراج الگوهای پرتکرار را در شرایط واقعی افزایش می‌دهد، و به کاربر اجازه می‌دهد چندین حداقل پشتیبانی را مشخص کند و ماهیت و فراوانی موارد مختلف را منعکس کند. علاوه بر این، الگوریتم بهبود یافته با اتخاذ مفهوم DiffSet برای سرعت بخشیدن به روند استخراج، باعث کاهش هزینه اسکن پایگاه داده و بهبود استفاده از فضای جستجو می‌شود. الگوریتم پیشنهادی با الگوریتم CFPgrowth++ مقایسه شد که نتایج نشان داد الگوریتم FP-ME از نظر زمان اجرا، استفاده از حافظه و مقیاس پذیری، از الگوریتم پیشرفته CFPgrowth++ به طور قابل ملاحظه‌ای عملکرد بهتری دارد.

یکی از بزرگترین مشکلات در استخراج مجموعه آیتم‌ها پرتکرار، انفجار تعداد نتایج است که به طور مستقیم بر زمان اجرای الگوریتم‌های استخراج مجموعه آیتم‌های پرتکرار تاثیر می‌گذارد. محققان برای حل این مشکل، استخراج مجموعه آیتم‌های بزرگ را به کمک الگوریتم SABMA[7] [8]در مجموعه‌های بسیار بزرگ پیشنهاد می‌دهند. در این الگوریتم، از یک ماتریس بیتی برای فشرده سازی مجموعه داده‌ها و نگاشت الگوریتم استخراج بر روی معماری آرایه‌های systolic استفاده می‌شود. نتایج، عملکرد بهتر این الگوریتم را نسبت به PP-tree[8] و LFP-tree[9] و DFP[10] نشان می‌دهد.

PIETM[11] [9] الگوریتمی است که به فرآیند استخراج درApriori شبیه است با این تفاوت که اساس کار این الگوریتم با دو بار اسکن است و از اصل Inclusion–Exclusion برای محاسبه پشتیبانی مجموعه آیتم‌های نامزد استفاده می‌کند و از فواصل تراکنش‌ها برای نشان دادن و ذخیره شناسه‌های تراکنش هر آیتم استفاده می‌کند که روند پردازش شمارش آیتم‌ها را آسان می‌کند. این الگوریتم زمان اجرای پایین تری نسبت به Apriori و FP-growth دارد زمانی که مجموعه ی داده‌ها دارای آیتم‌های زیاد است.

[7] systolic array based parallel algorithm
[8] Parallel Pattern tree
[9] Load balancing FP-tree
[10] Distributed frequent pattern
[11] Principle of Inclusion–Exclusion and Transaction Mapping

3- الگوریتم‌های توزیع‌شده

مدیریت داده‌های بزرگ و پردازش آن‌ها با منابع بسیار محدود چالش برانگیزاست. به عنوان مثال، حجم زیادی از داده‌ها به سرعت در حال تولید اند و در مکان‌های مختلف ذخیره می شوند که متمرکز کردن آن‌ها در یک مکان واحد به طور فزاینده‌ای پر هزینه است. علاوه بر این، الگوریتم‌های داده کاوی سنتی به طور کلی برخی از مشکلات و چالش‌ها مانند محدودیت حافظه، توانایی پردازش پایین وهارد دیسک ناکافی دارند. برای حل این مشکلات، الگوریتم‌های توزیع‌شده به عنوان یک جایگزین با ارزش در بسیاری از برنامه‌های کاربردی به نظر می رسد. چالش‌هایی از قبیل اتلاف انرژی محاسباتی و پهنای باند شبکه، هزینه بالا ورودی خروجی، تولید تعداد زیاد مجموعه آیتم‌های نامزد برای این الگوریتم‌ها نیز وجود دارند. در این بخش بعضی از الگوریتم‌های توزیع‌شده معرفی می‌شوند که به بهبود چالش‌های الگوریتم‌های توزیع‌شده پرداخته‌اند.

بینش بر این است که سرعت اجرا متناسب با تعداد گره‌های محاسباتی است، یعنی تعداد گره‌های محاسباتی بیشتر، باعث سرعت اجرای سریعتر می‌شود. با این وجود این موضوع برای الگوریتم‌های توزیع‌شده که الگوهای پرتکرار را استخراج می‌کنند اشتباه است زیرا الگوریتم‌ها نمی‌توانند استخراج را تا زمانی که داده‌های مورد نیاز را دریافت نکرده اند شروع کنند و باید اطلاعات را در میان گره‌های محاسباتی در شبکه فیزیکی مبادله کنند. تخصیص بیش از حد گره‌های محاسباتی می‌تواند به زمان اجرای طولانی منجر شود. علاوه بر ناکارآمدی اجرا، تخصیص منابع نا مناسب، منجر به اتلاف انرژی محاسباتی و پهنای باند شبکه می‌شود. در عین حال، اگر تعداد گره‌های اختصاص داده شده بیش از حد کم باشد، بارکاری نمی‌تواند به طور موثر توزیع شود. الگوریتم FLR-Mining [10] براساس تعیین تعداد مناسب گره‌های محاسباتی، با در نظر گرفتن کارامدی استفاده از منابع و توازن بار برای کشف الگوهای پرتکرار در سیستم‌های محاسباتی توزیع‌شده عمل می‌کند. FLR-Mining می‌تواند به طور خودکار تعداد گره‌های محاسباتی را تعیین و همچنین در مقایسه با PSWS و Dan-Mining می‌تواند به توازن بار بهتری دست پیدا کند.

مشکل اسکن تمام مجموعه داده‌ها در هر تکرار که منجر به هزینه بالا ورودی خروجی واستفاده‌ی زیاد از منابع می‌شود یک مسئله‌ی مهم است که برای برخی از الگوریتم‌ها مثل Apriori اتفاق می‌افتد. علاوه بر

این، در این الگوریتم هر گره در زمان فرایند استخراج، به کل مجموعه داده‌ها دسترسی دارد، که برای ذخیره سازی نیاز به مقدار زیادی حافظه دارد.برای حل این مشکل الگوریتم HFIM [11] ارائه شد که فضای جستجو را با تولید مجموعه آیتم‌های نامزد از تراکنش‌های جداگانه کاهش می‌دهد. علاوه بر این، پس از تکرار اول، همه آیتم‌های غیر تکراری از مجموعه داده‌های اصلی حذف می‌شوند. بنابراین مجموعه داده‌های تجدید نظرشده شامل تعداد کمتر از آیتم‌هاست. این الگوریتم با YAFIN که الگوریتمی مبتنی بر Apriori است مقایسه شد و ازمایشات صورت گرفته نشان داد که الگوریتم HFIM زمان اجرای بهتر و سرعت بیشتری را ارائه می‌دهد.

[12]DFIMA یک الگوریتم Apriori-like توزیع‌شده است که به طور قابل توجهی عملکرد را برای استخراج الگوهای پرتکرار بهبود می‌بخشد. از آنجا که الگوریتم Apriori یک فرآیند تکراری است. DFIMA با استفاده از یک روش هرس مبتنی بر ماتریکس، تعداد مجموعه آیتم‌های نامزد را کاهش می‌دهد.

[13]FDMCN یک الگوریتم سریع و توزیع‌شده مبتنی بر CARM برای استخراج الگوهای پرتکرار در شبکه‌های شلوغ می باشد.این الگوریتم از فشرده سازی خاصی از fp-tree استفاده می کند،که میزان انتقال داده‌ها را به صورت موثر کاهش می دهد و هدف اصلی این است که انتقال fp-tree را کاهش دهد تا فقط بخشی از اطلاعات برای استخراج با استفاده از گره‌های محاسباتی مورد نیاز باشد.این الگوریتم با الگوریتم PSWS مقایسه شد و نتایج نشان داد که FDMCN عملکرد بسیار خوبی را در زمینه کارایی و مقیاس پذیری ارائه می دهد.

4- نتیجه گیری

رشد روز افزون داده‌ها و استفاده از این داده‌ها یک مسئله‌ی مورد توجه در سال‌های اخیر می‌باشد. بسیاری از سازمان‌ها و شرکت‌های تجاری و همچنین بازار از تکنیک‌های داده کاوی که روشی کاربردی برای استخراج داده‌های مفید است و یک رویکرد موثر برای پیش برد کارها وافزایش سود دهی می‌باشد استفاده می‌کنند. در این مقاله ما یک مرور کلی بر تعدادی از الگوریتم‌های داده کاوی توزیع‌شده و متمرکز پیشرفته انجام داده ایم. اکثر الگوریتم‌های متمرکز سعی در کاهش فضای جستجو و استفاده از منابع کمتر دارند با این حال، با توجه به رشد بسیار زیاد داده‌ها، و هزینه جمع‌آوری آن‌ها استفاده از الگوریتم‌های توزیع‌شده نسبت به متمرکز کاربردی تر است.

5- مراجع

1. Sagiroglu, S. and D. Sinanc. *Big data: A review*. in *2013 International Conference on Collaboration Technologies and Systems (CTS)*. 2013.
2. Han J, K.M., Pei J, *Data mining: concepts and techniques.* Elsevier. 2011, New York.
3. Vo, B., et al., *A novel approach for mining maximal frequent patterns.* Expert Systems with Applications, 2017. 73(Supplement C): p. 178-186.
4. Deng, Z.-H. and S.-L. Lv, *PrePost+: An efficient N-lists-based algorithm for mining frequent itemsets via Children–Parent Equivalence pruning.* Expert Systems with Applications, 2015. 42(13): p. 5424-5432.
5. Huynh-Thi-Le, Q., et al., *An efficient and effective algorithm for mining top-rank-k frequent patterns.* Expert Systems with Applications, 2015. 42(1): p. 156-164.
6. Sohrabi, M.K. and R. Roshani, *Frequent itemset mining using cellular learning automata.* Computers in Human Behavior, 2017. 68(Supplement C): p. 244-253.
7. Gan, W., et al., *Mining of frequent patterns with multiple minimum supports.* Engineering Applications of Artificial Intelligence, 2017. 60(Supplement C): p. 83-96.
8. Sohrabi, M.K. and A.A. Barforoush, *Parallel frequent itemset mining using systolic arrays.* Knowledge-Based Systems, 2013. 37(Supplement C): p. 462-471.
9. Lin, K.-C., et al., *A frequent itemset mining algorithm based on the Principle of Inclusion–Exclusion and transaction mapping.* Information Sciences, 2014. 276(Supplement C): p. 278-289.
10. Lin, K.W. and S.-H. Chung, *A fast and resource efficient mining algorithm for discovering frequent patterns in distributed computing environments.* Future Generation Computer Systems, 2015. 52(Supplement C): p. 49-58.
11. Sethi, K.K. and D. Ramesh, *HFIM: a Spark-based hybrid frequent itemset mining algorithm for big data processing.* The Journal of Supercomputing, 2017. 73(8): p. 3652-3668.
12. Zhang, F., et al., *A distributed frequent itemset mining algorithm using Spark for Big Data analytics.* Cluster Computing, 2015. 18(4): p. 1493-1501.
13. Lin, K.W., S.-H. Chung, and C.-C. Lin, *A fast and distributed algorithm for mining frequent patterns in congested networks.* Computing, 2016. 98(3): p. 235-256.