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

بررسی الگوریتم‌های موازی و توزیع‌شده برای استخراج مجموعه آیتم پرتکرار

سیاوش فخری[1]، حمید رستگاری[2*]، مهدی شریفی[3]

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

Siavash.Fakhri@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

در جهان امروز، حجم زیادی از داده‌ها به طور مداوم توسط بسیاری از برنامه های کاربردی علمی مانند بیوانفورماتیک یا شبکه ها ایجاد می‌شوند. برای استخراج ارزش از این مجموعه داده‌های پیچیده، الگوریتم های استخراج داده‌ی مختلفی برای کشف ارتباطات پنهان و غیر بدیهی بین داده‌ها استفاده شده است. استخراج مجموعه آیتم‌های پرتکرار یکی از مهم‌ترین تکنیک‌های استخراج دانش از داده‌ها در بسیاری از برنامه‌های دنیای واقعی است. الگوریتم‌های مختلفی به طور گسترده برای استخراج مجموعه آیتم‌های پرتکرار از مجموعه‌های داده استفاده شده است. با این حال، فرآیند استخراج مجموعه آیتم‌های پرتکرار با چالش‌های فضای حافظه محدود و قدرت محاسبات روبه‌رو است. اعمال الگوریتم‌های استخراج مجموعه آیتم‌های پرتکرار سنتی برای کار بر روی داده‌های بزرگ، بسیار وقت‌گیر است. محاسبات موازی و توزیع‌شده استراتژی موثر و اغلب استفاده شده برای سرعت بخشیدن به الگوریتم‌ها برای مجموعه داده‌های بزرگ است. در این مقاله، به مرور کارهای انجام شده در این زمینه پرداخته شده است.

کلیدواژه‌ها: الگوریتم‌های توزیع شده، الگوریتم‌های موازی، الگوهای پرتکرار، قوانین انجمنی، اسپارک، هدوپ

1- مقدمه

کشف قوانین انجمنی، یک روش بسیار محبوب و موثر برای استخراج اطلاعات معنی‌دار از مجموعه داده‌های بزرگ می‌باشد. این کار تلاش می‌کند که ارتباطات احتمالی میان آیتم‌های موجود در مجموعه داده‌های بزرگ مبتنی بر تراکنش را پیدا کند. برای ایجاد این انجمن‌ها، مجموعه آیتم‌های پرتکرار باید تولید شوند.

بسیاری از روش‌ها برای استخراج مجموعه آیتم‌های پرتکرار در پایگاه داده‌ها بر روی یک ماشین ارائه شده‌اند. این روش‌ها در مجموعه داده‌های کوچک به خوبی کار می‌کنند. با این حال، مجموعه داده‌ها بزرگ‌تر و بزرگ‌تر شده‌اند و به دلیل ظرفیت محدود حافظه و توانایی محاسبه ماشین‌های تکی، این روش‌ها برای استخراج مجموعه آیتم‌های پرتکرار در مجموعه داده‌های عظیم ناکارآمد هستند. ظرفیت‌های حافظه، برای مدیریت کردن مجموعه‌ای کامل از آیتم‌های کاندید مورد نظر، به سرعت منفجر می‌شوند و هزینه‌های محاسباتی می‌تواند در یک ماشین، گران‌قیمت باشد.

به طور کلی، استفاده از تکنیک‌های داده کاوی برای مجموعه داده‌های عظیم نیاز به مقدار زیادی از منابع دارد. به همین دلیل ما شاهد استفاده‌ی فراوان از روش‌های موازی و توزیع‌شده هستیم که معمولا بر اساس چارچوب‌های توزیع‌شده مانند هدوپ[1] و اسپارک[2] هستیم؛ این دو بر پایه‌ی چارچوب برنامه‌نویسی نگاشت-کاهش[3]، که یک مدل برنامه‌نویسی ساده برای حل مسائل محاسباتی در مقیاس وسیع است می‌باشند. در واقع چارچوب برنامه‌نویسی نگاشت-کاهش مجموعه‌ای از توابع کتاب‌خانه را در دل خود دارد که جزئیات و پیچیدگی را از دید برنامه‌نویس پنهان می‌کند.

2- چالش‌ها

تکنیک‌های داده کاوی و الگوریتم‌های برای پردازش مقدار زیادی از اطلاعات به منظور استخراج اطلاعات مفید و جالب در بسیاری از زمینه‌های مختلف محبوب شده‌اند. الگوریتم‌هایی لازم است که داده‌ها را به طور خودکار و در روش‌های کارآمد مورد استفاده قرار دهند. با این وجود، حتی اگر عملکرد سیستم‌های کامپیوتری پی‌درپی بهبود یابد، آن‌ها مناسب نیستند تا با افزایش تقاضا برای برنامه‌های داده کاوی واندازه

[1] Hadoop
[2] Spark
[3] MapReduce

داده‌ها هماهنگ شوند. علاوه بر این، حافظه اصلی سیستم‌ها ممکن است به اندازه کافی برای نگهداری تمام اطلاعات مربوط به برنامه‌های فعلی کافی نباشد.

2-1- چالش‌های الگوریتم‌های موازی

الگوریتم‌های موازی با بهره‌گیری از حافظه اصلی و قدرت پردازش پردازنده و شتاب دهنده‌های موجود در رایانه‌های موازی، می‌توانند به راحتی به مسائل مربوط به زمان اجرا و حافظه مورد نیاز، رسیدگی کنند. با این حال، موازی سازی الگوریتم‌های موجود برای به دست آوردن عملکرد خوب و مقیاس پذیری با توجه به مجموعه داده‌های عظیم، بی‌اهمیت است. در حقیقت، سازماندهی داده‌ها و استراتژی تجزیه و تحلیل به منظور متعادل سازی حجم بار کار برای به حداقل رساندن وابستگی به داده‌ها اهمیت اساسی دارد. نگرانی دیگر مربوط به به حداقل رساندن هماهنگ سازی و سربار ارتباطات است. هزینه‌های I/O نیز باید به حداقل برسد. و در نهایت، و مصرف انرژی نیز باید بهینه باشد. برای ایجاد الگوریتم‌های موازی پیشرفته برای برنامه‌های کاربردی داده کاوی با کارایی بالا، نیاز به رسیدگی به چندین مشکل محاسباتی کلیدی است که می‌تواند به راه حل‌های جدید و بینش‌های جدید در برنامه‌های بین رشته‌ای منجر شود.

2-2- چالش‌های الگوریتم‌های توزیع‌شده

به طور فزاینده داده‌ها در مکان‌های توزیع‌شده جغرافیایی مختلف پخش می‌شوند. پردازش متمرکز این داده‌ها بسیار ناکارآمد و گران است. در برخی موارد ممکن است حتی غیر عملی باشد و به خطرات امنیتی منجر شود. بنابراین، در پردازش داده‌ها، به حداقل رساندن میزان داده‌هایی که رد و بدل می‌شوند و در عین حال تضمین صحت و کارایی یک چالش بسیار مهم است. استخراج داده‌های توزیع‌شده، تجزیه و تحلیل داده‌ها و استخراج را باید به شیوه‌ای گسترده انجام دهد و به محدودیت منابع، بخصوص محدودیت‌های پهنای باند، نگرانی‌های حفظ حریم خصوصی و قدرت محاسبات توجه کند.

3- بررسی الگوریتم های موازی

یکی از این الگوریتم‌ها، الگوریتم PHIKS[4] [1] مبتنی بر هدوپ است که مجموعه‌ای از بهینه سازی‌های قابل توجه را برای محاسبه‌ی انتروپی مشترک از Miki[5] با اندازه‌های مختلف ارائه می‌دهد. استخراج کردن Miki یک مشکل کلیدی در تجزیه و تحلیل داده ها با تاثیر بالقوه زیاد بر روی انواع مختلف وظایف، مانند یادگیری تحت نظارت، یادگیری بدون نظارت و یا بازیابی اطلاعات است. به کمک این الگوریتم به طور چشمگیری زمان اجرا، هزینه ارتباطات و مصرف انرژی را در یک پلتفرم محاسباتی کاهش می‌دهد. الگوریتم PaMPa-HD[6] [2]، یک الگوریتم استخراج مجموعه آیتم بسته‌ی پرتکرار موازی مبتنی بر هدوپ برای مجموعه داده‌هایی با ابعاد بزرگ است. استخراج مجموعه آیتم بسته‌ی پرتکرار یکی از پیچیده‌ترین روش های کاوش در داده کاوی است. اغلب برای کشف آیتم‌هایی که با هم اتفاق می‌افتند با توجه به یک آستانه فراوانی ارائه شده توسط کاربر، به نام حداقل پشتیبانی مورد استفاده قرار می‌گیرد. کارایی این روش در زمان اجرا، متعادل سازی بار و استحکام حافظه است. FiDoop-DP[7] [3] با استفاده از مدل برنامه‌نویسی هدوپ یک رویکرد پارتیشن‌بندی برای حل مشکل ارتباطات بالا و سربار استخراج ناشی از تراکنش‌های افزونه ارسال شده در میان گره‌های محاسباتی ارائه می‌دهد. این الگوریتم، از همبستگی بین تراکنش‌ها استفاده می‌کند و با تطبیق معیار شباهت و تکنیک Hash کردن حساس به مکان، مکان‌های بسیار مشابه را به یک پارتیشن داده برای بهبود محل، بدون ایجاد تعداد بیش از حد تراکنش های افزونه می‌دهد.

R-Apriori[8] [4] یک الگوریتم Apriori موازی بر اساس چارچوب اسپارک است که تولید مجموعه کاندید را برای رسیدن به سرعت بالا با حذف کلی تولید کاندید سرعت می‌بخشد. در این الگوریتم عملکرد تکرار با حذف مرحله تولید مجموعه کاندید و با استفاده از یک فیلتر bloom به جای یک درخت Hash بهبود داده می شود. آیتم های پرتکرار یکتا در فیلتر bloom ذخیره می‌شوند چرا که فیلترهای bloom سریع‌تر از درخت‌های Hash هستند و می‌توانند به راحتی آیتم‌های با طول یک را ذخیره کنند.

[4] Parallel Highly Informative K-ItemSet
[5] maximally informative k-itemset
[6] Parallel MapReduce-Based Frequent Pattern Miner for High-Dimensional Data
[7] Data partitioning in frequent itemset mining on HaDoop clusters
[8] Redu*ced-Apriori

MR-Apriori[9] [5] برای مشکل کمبود قدرت محاسبات مبتنی بر هدوپ و بر اساس الگوریتم Apriori است. این الگوریتم کارایی خود را در مجموعه داده‌های عظیم نشان می‌دهد.

YAFIM[10] [6]، یک الگوریتم دیگر Apriori موازی بر اساس چارچوب اسپارک است که در مقایسه با الگوریتم Apriori که با هدوپ پیاده سازی شده است، 18 برابر سریع‌تر است. S-FPG[11] [7] یک الگوریتم FP-Growth مبتنی بر اسپارک است که می‌تواند به خوبی و به طور قابل توجهی داده‌های بزرگ را پردازش کند. کارایی و مقیاس پذیری از ویژگی‌های این الگوریتم است.

4- بررسی الگوریتم های توزیع شده

DFIMA[12] [8] یک الگوریتم استخراج مجموعه آیتم پرتکرار توزیع‌شده مبتنی بر اسپارک است که با استفاده از رویکرد هرس مبتنی بر ماتریس، تعداد مجموعه آیتم‌های کاندید را کاهش می‌دهد. این الگوریتم سرعت و مقیاس پذیری بالایی را ارائه می‌دهد. FDMCN[13] [9] یک الگوریتم سریع و توزیع‌شده مبتنی بر CARM برای استخراج الگوهای پرتکرار در شبکه‌های شلوغ می‌باشد. بنابراین انتقال داده‌های رد و بدل شده را کاهش می‌دهد و به سرعت اجرا و مقیاس پذیری در نظر می‌گیرد. DSCAN-MAX[14] [10] برای استخراج توزیع‌شده مجموعه آیتم‌های پرتکرار حداکثری است. این الگوریتم از ویژگی‌های SCan-Treeاستفاده کامل می‌کند که این کار به طور قابل توجهی ارتباطات مورد نیاز بین گره‌ها را کاهش می‌دهد، و بدین گونه بهره‌وری الگوریتم را افزایش می‌دهد.

HFIM[15] [11] الگوریتمی مبتنی بر اسپارک است که از ترتیب عمودی مجموعه داده‌ها برای حل مشکل اسکن‌ها در هر تکرار استفاده و فضای جستجو را با تولید مجموعه آیتم‌های کاندید از تراکنش‌های جداگانه کاهش می‌دهد. نتایج تجربی نشان می‌دهد که HFIM از لحاظ زمان اجرا و مصرف حافظه، عملکرد قابل قبولی دارد. FAMR[16] [12] یک الگوریتم Apriori استخراج الگوهای پرتکرار مبتنی بر هدوپ است که از

[9] MapReduce Apirori
[10] Yet Another Frequent Itemset Mining
[11] FP-Growth Algorithm under Apache Spark
[12] Distributed Frequent Itemset Mining Algorithm
[13] fast and distributed algorithm for mining frequent patterns in congested networks
[14] Distributed Global Maximal Frequent Itemsets based on Sorted SCan-Tree
[15] Hybrid Frequent Item Mining
[16] Frequent Patterns Mining Algorithm Based on MapReduce

تولید کاندیدهای بی‌شمار پرهیز می‌کند و باعث استفاده‌ی مناسب از حافظه و ذخیره‌ی مقدار زیادی زمان می‌شود.

Dist-Eclat و BigFIM دو الگوریتمی هستند که در [13] معرفی شدند. روش Dist-Eclat، Eclat محض است که فضای جستجو را به صورت یکنواخت در بین ماشین‌ها توزیع می‌کند. این الگوریتم قادر به استخراج داده‌های بزرگ است، اما در هنگام برخورد با مقادیر عظیم داده‌ها به مشکل می‌خورد. الگوریتم BigFIM ابتدا از Apirori برای استخراج مجموعه‌های پرتکرار به طول k استفاده می کند و بعد از Eclat استفاده می‌کند.

5- نتیجه گیری

در این مقاله، ما یک مرور کلی بر الگوریتم‌های موازی و توزیع‌شده و چالش‌های موجود در این الگوریتم‌ها انجام دادیم. با توجه به رشد عظیم داده‌های خام و بزرگ شدن پایگاه داده‌ها، تغییر رویکردها از الگوریتم‌های سنتی و متمرکز به الگوریتم‌های موازی و توزیع شده، بیشتر و بیشتر احساس می‌شود. استفاده از این رویکردها، علاوه بر برطرف کردن بسیاری از مشکلات و چالش‌های سیستم‌های متمرکز، خود نیز با چالش‌هایی روبه‌رو هستند که باید به آن‌ها توجه ویژه‌ای شود. در صورت بی‌توجهی مشکلات چندین برابر می‌شود و این رویکردها کارایی خود را از دست می‌دهند.

6- مراجع

[1] S. Saber, A. Reza, and M. Florent, "A highly scalable parallel algorithm for maximally informative k-itemset mining," *Knowledge and Information Systems,* journal article vol. 50, no. 1, pp. 1-26, January 01 2017.

[2] A. Daniele, B. Elena, C. Tania, G. Paolo, P. Fabio, and M. Pietro, "A Parallel MapReduce Algorithm to Efficiently Support Itemset Mining on High Dimensional Data," *Big Data Research,* vol. 10, no. Supplement C, pp. 53-69, 2017/12/01/ 2017.

[3] Y. Xun, J. Zhang, X. Qin, and X. Zhao, "FiDoop-DP: Data Partitioning in Frequent Itemset Mining on Hadoop Clusters," *IEEE Transactions on Parallel and Distributed Systems,* vol. 28, no. 1, pp. 101-114, 2017.

[4] S. Rathee, M. Kaul, and A. Kashyap, "R-Apriori: An Efficient Apriori based Algorithm on Spark," presented at the Proceedings of the 8th Workshop on Ph.D. Workshop in Information and Knowledge Management, Melbourne, Australia, 2015.

[5] X. Lin, "MR-Apriori: Association Rules algorithm based on MapReduce," in *2014 IEEE 5th International Conference on Software Engineering and Service Science*, 2014, pp. 141-144.

[6] H. Qiu, R. Gu, C. Yuan, and Y. Huang, "YAFIM: A Parallel Frequent Itemset Mining Algorithm with Spark," in *2014 IEEE International Parallel & Distributed Processing Symposium Workshops*, 2014, pp. 1664-1671.

[7] A. D. d. Gassama, F. Camara, and S. Ndiaye, "S-FPG: A parallel version of FP-Growth algorithm under Apache Spark," in *2017 IEEE 2nd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, 2017, pp. 98-101.

[8] Z. Feng, L. Min, G. Feng, S. Weiming, S. Abdallah, and M. Yunlong, "A distributed frequent itemset mining algorithm using Spark for Big Data analytics," *Cluster Computing,* journal article vol. 18, no. 4, pp. 1493-1501, December 01 2015.

[9] L. K. W., C. Sheng-Hao, and L. Chun-Cheng, "A fast and distributed algorithm for mining frequent patterns in congested networks," *Computing,* journal article vol. 98, no. 3, pp. 235-256, March 01 2016.

[10] Y. Huang, J. Wang, Y. Li, and Q. Lin, "A mining algorithm for distributed global maximal frequent itemsets based on Sorted SCan-Tree," in *2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, 2016, pp. 1818-1824.

[11] S. K. Kumar and R. Dharavath, "HFIM: a Spark-based hybrid frequent itemset mining algorithm for big data processing," *The Journal of Supercomputing,* journal article vol. 73, no. 8, pp. 3652-3668, August 01 2017.

[12] R. Yu, M. Lee, Y. Huang, and S. Chen, "An efficient frequent patterns mining algorithm based on MaPreduce framework," *IET Conference Proceedings,* pp. 1-5

[13] S. Moens, E. Aksehirli, and B. Goethals, "Frequent Itemset Mining for Big Data," in *2013 IEEE International Conference on Big Data*, 2013, pp. 111-118.