الجمهورية الجزائرية الديمقراطية الشعبية
People's Democratic Republic of Algeria
وزارة التعليم العالي والبحث العلمي
Ministry of Higher Education and Scientific Research

جامعة غرداية
كلية العلوم والتكنولوجيا
قسم الرياضيات والإعلام الآلي

مذكرة تخرج لنيل شهادة الماستر
شعبة: الرياضيات والإعلام الآلي
تخصص: الأنظمة الذكية لإستخراج المعارف (SIEC)

الموضوع

مجموعات العناصر المتكررة القصـوى
Maximal Frequent Itemsets

من إعداد الطالبتين
إيمان رزاق
مرية رزاق

تمت مناقشتها وإجازتها علنا يوم: 2017/06/11، أمام اللجنة المكونة من:

الأستاذ. سليمان بلعور	أستاذ مساعد أ	جامعة غرداية	– رئيسا
الأستاذ. سليمان أولاد النوي	أستاذ مساعد أ	جامعة غرداية	– مشرفا
الأستاذ. عبد القادر بوهاني	أستاذ مساعد أ	جامعة غرداية	– مناقشا
الأستاذ. خالد كشيدة	أستاذ مساعد ب	جامعة غرداية	– مناقشا
البروفيسور. جلول زيادي	أستاذ محاضر	جامعة روان– فرنسا	– ضيفا شرفيا

السنة الجامعية
2017 – 2016

شكر وعرفان

الحمد لله رب العالمين حمدا يليق بجلال وجهه وعظيم سلطانه
والصلاة والسلام على قدوة المربين نبينا محمد
وعلى آله وصحبه أجمعين.

وعملا بقوله صلى الله عليه وسلم:
<< مَنْ لَّمْ يَشْكُرِ القَلِيلْ لَّمْ يَشْكُرِ الكَثِيرْ وَمَنْ لَّمْ يَشْكُرِ النَّاسْ لَّمْ يَشْكُرِ الله >>
رواه أحمد والترمذي.

نتقدم بالشكر الجزيل والعرفان الجميل
إلى الأستاذ المشرف أولاد النوي سليمان لما قدمه لنا
من جهد ونصح ومعرفة طيلة إنجاز هذه المذكرة،
لك منا أسمى معاني التقدير والعرفان،
أدام الله عليك الصحّة والعافية.

كما نتقدم أيضا بالشكر الجزيل إلى
الأخ الكريم أقموم أسامة والأخ العزيز رزاق عبد الجليل
على مجهوداتهما القيمة.

كما لا ننسى أن نتقدم بأرقى وأثمن عبارات الشكر و العرفان
إلى أستاذتنا الكرام الذين أشرفوا وساهمو في تكويننا
طوال مشوارنا الجامعي ونخص بالذكر
الأستاذ بلعور سليمان والبروفيسور زيادي جلول.

إلى كل من ساهم في إنجاز هذا العمل المتواضع
من قريب أو من بعيد.

إلى أعضاء لجنة المناقشة لقبولهم
مناقشة وإثراء هذه المذكرة.

إهداء

إلهي لا يطيب الليل إلا بشكرك و لا يطيب النهار إلا بطاعتك
ولا تطيب اللحظات إلا بذكرك ولا تطيب الجنة إلا برؤيتك تباركت ربنا وتعاليت.

إلى من بلغ الرسالة وأدى الأمانة ونصح الأمة
إلى نبي الرحمة سيدنا محمد صلى الله عليه وسلم.

إلى من أحمل اسمه بكل إفتخار إلى من كلله الله بالهيبة والوقار
أرجو من الله أن يمد في عمرك لترى ثمارا قد حان قطافها بعد طول إنتظار والدي العزيز.

إلى من بها أكبر وعليها أعتمد إلى معنى الحب والحنان والتفاني
إلى من كان دعاؤها سر نجاحي و بها عرفت معنى الحياة أمي الغالية أطال الله عمرك في طاعته.

إلى من آثروني على أنفسهم وعلموني علم الحياة إخوتي وأخواتي.

إلى البراعم : عمر، عثمان، صهيب، تسنيم، إيناس...

إلى كل من عرفتهم وأخص بالذكر صديقاتي...

إلى كل من كنت يوما تلميذة أو طالبة عنده...

إلى جميع أقسام كلية العلوم والتكنولوجيا
وأخص بالذكر طلبة ثانية ماستر إعلام آلي 2016 . 2017...

إلى كل من نساه قلمي ولم ينساه قلبي...

مارية.

إهداء

أهدي بكل فرح ومحبة وسرور هذا العمل
إلى والديّ العزيزين عرفانًا لكل ما بذلوه من جهد وتضحيات
ودعم بمحبة خالصة في سبيل أن أصل إلى ما أنا عليه اليوم
كل كلمات الشكر تقف عاجزة أن توفيكما حقكما
أو أن تعبر عن مدى إمتناني لكما.

إلى جدتي الغالية حفظها الله ورعاها تقديرا لمحبتها العميقة
واللامحدودة، أبقاك الله نورا لنا دائما.

إلى أخواتي العزيزات اللاتي لا يحلو العيش إلا بهنّ
أحبكم جدا جدا جدا Moon Zahra .

إلى من ليسوا في البيت ولكنهم في البال دائما.

إلى العائلة الكبيرة كلها، خاصة جدتي وخالتي العزيزتين.

إلى من رافقنني على درب المحبة والأخوة
صديقتيّ الحبيبتين.

إلى من لم تمنعنا اللغة من أن نكون أعز الأصحاب.

إلى كل أصدقائي وزملائي في الدراسة وفي العمل
أشكركم، وأعتز بمعرفتكم جميعا.

إلى كل الأهل والأحباب، وإلى كل من يعرفني من قريب أو من بعيد.

إلى جميع طلبة ماستر إعلام آلي دفعة 2016 و2017.

إلى كل من نسيه القلم ولم ينساه القلب.
إيمـان.
28 مـاي 2017
02 رمضان 1438

ملخص

التنقيب عن قواعد الإرتباط واحدة من أشهر عمليات التنقيب في البيانات، والتي تستخدم في البحث عن العلاقات غير الظاهرة أو المخفية بين العناصر داخل قواعد المعاملات. ونتيجة لذلك فإن تطبيقاتها اليوم تشمل العديد من المجالات المختلفة، أبرزها المجال التجاري حيث تتواجد قواعد الإرتباط في كلَ من عمليات تحليل سلة التسوق، عمليات تجميع المنتجات وترتيبها في المراكز والمحلات التجارية إضافة إلى عمليات تصميم دليل السلع والمنتجات... .

لكن المشكلة الرئيسية في التنقيب عن قواعد الإرتباط تكمُن في البحث عن مجموعات العناصر المتكررة التي تُساهم في إنشاء هذه القواعد. لذلك أقتُرِحت خلال العقدين الماضيين العديدُ من الخوارزميات لحلَ هذه المشكلة، أبرزها خوارزمية Apriori التي تعتمد المقاربةَ بالمستويات، خوارزمية Eclat والتي تستخدِم المُقاربة العمودية وأيضا خوارزمية المقاربة بالإسقاط FPGrowth.

حقَقت مُختلف هذه الخوارزميات نجاحًا باهرا على مدى سنوات، لكنها بقيت تُعاني من كونها "مكلفة للغاية " من حيث الزمن المُستغرق والمساحة التخزينية للذاكرة عند التنفيذ.

في هذا العمل ستتِم دراسة التمثيلات المتراصَة لمجموعات العناصر المتكررةِ المُقترحة في بداية الألفية الجديدة، وبشكل خاص تمثيلات مجموعات العناصر المتكررة القصوى، من حيث مفاهيمها الأساسية وإستعراض لبعض الخوارزميات الموجودة حاليا للبحث عن مجموعات العناصر المتكررة القصوى في قواعد المعاملات، ليتم بعدها القيام بمقارنة لإثنتين من هذه الخوارزميات هما FPMax و Charm-MFI والموجودتين في منصة SPMF مفتوحة المصدر بإستعمال قواعد معاملات مختلفة الأنواع والأحجام.

حيث أظهرت النتائج التجريبية تفوق خوارزمية FPMax على خوارزمية Charm-MFI من حيث الإستهلاك الأقل للمساحة التخزينية للذاكرة، أمّا بالنسبة للزمن المستغرق للتنفيذ فإنّ خوارزمية Charm-MFI تفوقت في قواعد المعاملات المتناثرة بالمقابل رجحت كفة خوارزمية FPMax في قواعد المعاملات المكثفة.

كلمات مفتاحية: التنقيب في البيانات، قواعد الإرتباط، مجموعة العناصر المتكررة، مجموعة العناصر المتكررة القصوى، خوارزميات، دراسة تجريبية، جافا.

ABSTRACT

Association Rules Mining is one of the most famous Data Mining tasks, it is used to find the hidden relationships in a given transactional databases. Therefore, it makes part nowadays inside a variety of applications such as Market Basket Analysis, products clustering, catalog design, store layout and many more.

The major problem in mining Association Rules is enumerating the whole frequent itemsets that help in generating such rules, many algorithms were proposed to solve this problem; among them was the Apriori Algorithm that uses the Level-wise approach, Eclat Algorithm which uses the Vertical approach and the FPGrowth Algorithm which uses the Projection approach, these algorithms made a huge success during the last two decades, but in a computational point of view, all of them suffer from being "expensive", whether in time consumption or in memory requirements during the execution.

In this work, we study some compact representations of the frequent itemsets especially the Maximal Frequent Itemsets and its basic concepts. In addition, we also study some of the current algorithms used to find the Maximal Frequent Itemsets in a given transactional base, and finally testing two of them that are already implemented in SPMF the java open source platform which are FPMax and Charm-MFI, using transactional datasets with different types and sizes.

Experimental results showed that FPMax algorithm has exceeded Charm-MFI algorithm in terms of lower requirements of the memory, whereas in the running-time point of view, results showed that Charm-MFI algorithm has outperformed FPMax algorithm in sparse Datasets, while in a condensed Datasets FPMax algorithm has shown a good performance.

Key words: Data mining, Association rules, Frequent itemsets, Maximal Frequent itemsets, Algorithms, Experimental Analysis, Java.

الفهـــــرس

قائمة الأشكال

قائمة الجداول

قائمة الخوارزميات

إختصارات

الإختصار	مدلوله
CFI	Closed Frequent Itemset مجموعة العناصر المتكررة المغلقة
CPB	Conditional Pattern Base قاعدة الأنماط الشرطية
$ECLAT$	Equivalence CLAss Transformation تحويلات فئة التكافؤ
FI	Frequent Itemset مجموعة العناصر المتكررة
$FPGrowth$	Frequent Pattern Growth نمو النمط المتكرر
$FP-tree$	Frequent Pattern-tree شجرة النمط المتكرر
I	Itemset مجموعة العناصر
KDD	Knowledge Discovry from Databases إستخراج المعارف من قواعد البيانات
LCM	Linear time Closed itemset Miner مجموعة العناصر المغلقة ذات الزمن الخطي
$MAFIA$	MAximal Frequent Itemset Algorithm خوارزمية مجموعة العناصر المتكررة القصوى
MFI	Maximal Frequent Itemset مجموعة العناصر المتكررة القصوى
$MFI-Tree$	Maximal Frequent Itemset-Tree شجرة العناصر المتكررة القصوى
$SPMF$	Sequential Pattern Mining Framework منصة التنقيب عن الأنماط المتسلسلة

مقدمة

لا يخفى على أحد اليوم أن البيانات غمرت مختلف نواحي الحياة منذ دخول الحاسبات وتطبيقات قواعد البيانات سوق العمل في الإدارات والشركات زمنًا مضى، وصولا إلى القفزة التكنولوجية التي إجتاحت العالم في الآونة الأخيرة أين أصبحت البيانات لا تُنشأ من حواسيب المؤسسات والشركات فقط، بل أضحى الشخص العادي وبمجرد إستخدامه لمختلف وسائل التكنولوجيا مصدرا للبيانات هو الآخر. لينتقل الأمر إلى الآلات والأجهزة الذكية المرتبطة ببعضها عبر "إنترنت الأشياء" (Internet of Things) وتصبح هي الأخرى مصدرا غنيا للبيانات المختلفة.

عملية تخزين هذه الكميات الهائلة من البيانات ليست حقا بالمشكلة الكبيرة، فلقد إزدادت سعات التخزين بشكل ملحوظ لتثبت قدرتها على مواكبة الأحجام والكميات الضخمة للبيانات المتزايدة يوما بعد يوم، ليبرز تحدٍ آخر إلى الساحة نتيجة لهذا التراكم ألا وهو محاولة إكتشاف وإستخلاص معارف من داخل تلك البيانات من أجل فهم أعمق لها، بغرض توفير خدمات حسب الطلب، ومن أجل القدرة على التنبؤ بأحداث مُستقبلية بناءً على معطيات سابقة، لكن الوسائل والأساليب الإحصائية العادية لم تتمكن من معالجة بيانات بهذه الضخامة، فكان لابد من البحث عن طرق بديلة تضمن الوصول إلى هذه المعرفة.

من هنا أستُحدث مجال التنقيب في البيانات (Data Mining) أو ما يعرف بشكل أدق بإستخلاص المعارف من قواعد البيانات (Knowledge Discovery from Databases (KDD)) الذي يُعتبر نتاج تكامل عدة فروع معرفية من بينها: أنظمة قواعد البيانات (Database Systems)، الإحصاء (Statistics) بالإضافة إلى مجالات التعلّم الآلي والذكاء الإصطناعي (Machine Learning and Artificial Intelligence) إلى غير ذلك من التخصصات، مما جعله أكثر كفاءة وفعالية في تحليل ومعالجة البيانات مقارنةً بالوسائل التقليدية، ولعل هذا ما رشَحه لأن يكون واحدا من أهم التقنيات الحديثة التي من شأنها أن تغيَر العالم حسب تقرير لمعهد MIT للتكنولوجيا [13].

يعرَف إستخلاص المعارف من قواعد البيانات على أنه "إستخراج معارف جديدة، مفهومة وواضحة، ويحتمل أن تكون مفيدة إنطلاقا من حقائق مخفية داخل كميات كبيرة من البيانات"، هذه المعارف تُساعد على إتخاذ قرارات سليمة ومدروسة بعناية [5].

تمرُ عملية إستخلاص المعارف من قواعد البيانات قبل ذلك بعدة مراحل تبدأ من مرحلة إختيار وإعداد البيانات (Selecting and Preprocessing Phase)، لتنتهي عند مرحلة تفسير النتائج المتحصل عليها، مرورا بمرحلة تُعد قلب عملية إستخلاص المعارف وهي مرحلة "التنقيب في البيانات" والتي يتم على مستواها تشكيل النماذج وإختبارِها لتحديد مدى قابليتها للتعميم.

يضم التنقيب في البيانات أيضا عدة مهام (Tasks) كالوصف (Description)، التقدير (Estimation)، التنبؤ (Prediction) والتصنيف (Classification)، إضافة إلى إستخراج قواعد الإرتباط (Association Rules) والذي يُعتبر من المهام القاعدية للتنقيب في البيانات التي جذبت إهتمام العديد من الباحثين في هذا المجال.

يُمكن تقسيم عملية إستخراجِ قواعد الإرتباط إلى مرحلتين هما [13]:

- أولا: إيجاد مجموعات العناصر المتكررة (Frequent Itemsets) في قاعدة البيانات.
- ثانيا: إنشاء قواعد الإرتباط إنطلاقا من تلك المجموعات المتكررة.

يُعتبر الجزء الخاص بعملية إنشاء قواعد الإرتباط "بسيطا" بالمقارنة مع عملية إيجاد مجموعات العناصر المتكررة داخل قواعد البيانات، فهذه الأخيرة كانت في مرحلة أولى تتم عبر التعداد الشامل لجميع مجموعات العناصر المتكررة (All Frequent Itemsets) مما يجعلها عملية "مكلَفة" عند الأخذ بعين الإعتبار الزمن المُستغرق وسعة المساحة التخزينية للذاكرة وقت التنفيذ، لتظهر بعد ذلك من أجل تفادي هذه المشكلة تمثيلات متراصة (Compressed Representations) لمجموعات العناصر المتكررة كتمثيلات مجموعات العناصر المتكررة المغلقة (Closed Frequent Itemsets (CFI)) وتمثيلات مجموعات العناصر المتكررة القصوى (Maximal Frequent Itemsets (MFI))، حيث ستتركز الدراسة على عملية إيجاد مجموعات العناصر المتكررة القصوى (MFI) من حيث المفاهيم الأساسية والخوارزميات المتبعة حاليًا.

سيتم بالتالي تقسيم مذكرة البحث إلى ثلاثة فصول:

يهتم الفصل الأول بالتنقيب في البيانات ومفهومه بصفة عامة، ثم المراحل المختلفة لعملية إستخلاص المعارف من البيانات إضافة لعرض مهام وتقنيات التنقيب في البيانات، ليُختتم بأمثلة عن إستخدامات التنقيب في البيانات على أرض الواقع.

أما الفصل الثاني فيشرح مفهوم قواعد الإرتباط ومجموعات العناصر المتكررة ويُلقي الضوء على المقاربات وأبرز الخوارزميات المنتهجة في عملية الإستخراج الكلي لمجموعات العناصر المتكررة: كالمقاربة البديهية وخوارزمية Brute Force، فالمقاربة بالمستويات التي تُعتبر

خوارزمية Apriori الخوارزمية المرجِع فيها، ثم المقاربة العمودية والخوارزمية الأشهر خوارزمية Eclat ليكتمل الفصل مع خوارزمية المقاربة بالإسقاط FPGrowth، مع توضيح لعملية إنشاء قواعد الإرتباط وشروط إختيار قاعدة الإرتباط القوية وذات الموثوقية.

ويختص الفصل الثالث بالتمثيلات المتراصة لمجموعات العناصر المتكررة وبشكل خاص مجموعات العناصر المتكررة القصوى (MFI) وكيفية إنتقاء هذه الأخيرة من مجموعات العناصر، تتم بعدها دراسة لبعض خوارزميات إيجاد مجموعات العناصر المتكررة القصوى المُستعملة حاليا.

لتُجرى في الفصل الرّابع مقارنة بين خوارزميتي FPMax و Charm-MFI من ناحية الزمن المستغرق ومساحة الذاكرة عند التنفيذ بإستعمال منصة SPMF مفتوحة المصدر.

ثم تُختتم المذكرة بخاتمة عامة وآفاق وتطلعات بحثية.

الفصل الأول
التنقيب في البيانات

1.1 مقدمة

أصبحت البيانات في عصرنا الحالي تتزايد بشكل لا يمكن تصوره، فنرى أن تراكم البيانات قد شمل مختلف الميادين (الاقتصادية، العلمية، الصناعية...إلخ) بل ويمكن أن تتولد البيانات حتى من طرف الأشخاص العاديين، فكل شخص يترك "أثرا رقميا" عند القيام بأعماله اليومية المعتادة كالتسوق عبر الأنترنت، أو تصفح الجرائد، قراءة الكتب، الإستماع إلى الموسيقى أو مشاهدة الفيديوهات عبر التطبيقات المختلفة.

مثل هذه البيانات يمكنها أن توفر مصدرا غنيا يساعد على إتخاذ قرارات دقيقة وسليمة وفهم أوضح لسلوك الأفراد بفضل تكنولوجيا التنقيب في البيانات.

في هذا الفصل سيتم التطرق إلى مفهوم التنقيب في البيانات، أسباب ظهوره، المراحل المختلفة لعملية التنقيب في البيانات، المهام والتقنيات إضافة إلى تطبيقاته في أرض الواقع.

2.1 مفهوم التنقيب في البيانات

التنقيب في البيانات هو جوهر عملية إستخلاص المعارف من البيانات يتم من خلاله البحث بطريقة آلية وذكية عن الأنماط المفيدة وإستخراجها في شكل نماذج توضيحية من أجل إستخلاص معلومات يمكن الإستفادة منها مستقبلا [10]. بمعنى آخر: هو إنشاء علاقات وإستنباط أنماط من خلال أحجام هائلة من البيانات الخام [15]. توجد مسميات أخرى لمصطلح التنقيب في البيانات من بينها:

- تحليل البيانات / تحليل الأنماط.
- ذكاء إدارة الأعمال (Business Intelligence).
- إستخراج المعارف... .

وقبل التعمق أكثر في التنقيب في البيانات ينبغي التفريق بين البيانات، المعلومات والمعارف [3]:

- المعطيات في صورتها الخام دون أي فحص أو تحليل تُسمى بيانات (Data).
- البيانات التي تم تجميعها ووضعها في سياق معين تُسمى معلومات (Informations).
- المعلومات التي تم تحليلها واستخلاصها بطرق أكثر تعقيداً، والتي خضعت للتفسيرات من أجل استنباط العلاقات المختلفة بين الظواهر للمساعدة على اتخاذ القرارات تسمى بالمعارف (Knowledge).

أصبح للتنقيب في البيانات اليوم أهمية كبيرة جدا في مختلف المجالات خاصة في المجال الإقتصادي حيث أن التنقيب في البيانات يسمح بتحسين تسيير الموارد سواء كانت مادية أو بشرية، أمثلة على ذلك [15]:

- البنوك: قرار الموافقة على منح القروض من عدمها يتم إعتمادا على تحليل البيانات الشخصية لصاحب الطلب، أيضا يعتمد ذلك على تحليل معلومات القرض وكذا التحليل الزمني لطلبات قروضه السابقة.
- المراكز التجارية: من أجل تسهيل عملية اقتناء المنتجات من طرف الزبائن تتم عملية إعادة ترتيب رفوف المحلات من خلال تجميع السلع التي غالبا ما تشترى مع بعضها في أماكن متقاربة داخل هذه المحلات.

3.1 التنقيب في البيانات، لماذا؟

" نحن غارقون في البيانات لكننا متعطشون للمعرفة " هذا ما قاله الكاتب John Naisbitt في كتابه Megatrends الذي نشر سنة 1984 [13].

مشكلة اليوم ليست في قلة تدفق البيانات والمعلومات، ففي الواقع نحن مغمورون بها في مختلف الميادين لكن المشكلة الأبرز تكمن في كيفية إستخراج المعارف من هذه البيانات الضخمة، فهي معقدة بما يكفي لأن تعجز التقنيات التقليدية عن التعامل معها، لذا فإننا نحتاج إلى تقنيات تساعد في تحسين عملية البحث والإستنباط وهذا ما يعرف بتقنيات الذكاء الصناعي والتي من بينها "التنقيب في البيانات".

التنقيب في البيانات وعملية إكتشاف المعارف والأنماط جاء نتيجة لإلتقاء عدة عوامل منها [13]:

- النمو الهائل في تجميع البيانات (كمثال على ذلك عملية تخزين المعلومات عن المشتريات في المراكز التجارية المختلفة).

- تخزين هذه البيانات داخل مستودعات البيانات (Data Warehouses) مما يسمح للمؤسسة بالوصول إلى قاعدة بيانات معتمدة للتحليل.
- تطور برمجيات للتنقيب في البيانات، جاهزة للاستغلال.
- وفرة البيانات وسهولة الوصول إليها من خلال الانترنت أو الشبكات الداخلية (Intranets).
- ضغط المنافسة من أجل تعزيز حصة السوق للشركات في السوق العالمية.
- تطور عتاد الإعلام الآلي وسعة التخزين.

4.1 التنقيب في البيانات، أي نوع من البيانات؟

عملية التنقيب في البيانات تُطبَّق على جميع البيانات أيّا كان نوعها، سواءً كانت قواعد بيانات علائقية أو قواعد معاملات، قواعد بيانات كائنية التوجه، مستودعات البيانات، قواعد البيانات غير المتجانسة، بالإضافة الى قواعد البيانات المتقدمة والتي من بينها قواعد البيانات الزمنية وقواعد البيانات الحيزية، قواعد البيانات النصية، قواعد البيانات متعددة الوسائط [10].

- **قواعد البيانات العلائقية** Relational Databases

يعتبر هذا النوع من قواعد البيانات من أكثر الأنواع إستعمالا حيث أنه يعتمد على تنظيم البيانات في جداول تتميز بوجود علاقة فيما بينها مما يمكّن المستخدِم من الوصول إلى البيانات من مختلف أجزاء قاعدة البيانات.

- **قواعد المعاملات** Transactional Databases

هي عبارة عن قاعدة سجلات بحيث يعبر كل سجل عن معاملة ما، يكون لكل معاملة معرّف ومجموعة عناصر تمثلها (مثال: قائمة مشتريات الزبائن في المراكز التجارية).

- **مستودعات البيانات** Data Warehouses

هي قاعدة بيانات ضخمة يتم إنشاؤها بواسطة دمج بيانات مختلفة ومتباينة، على مستواها يتم تخزين البيانات الحالية والبيانات السابقة للشركات، من أجل إستخدامها لتقديم تقارير للجهات المعنية حول توجهات نشاطاتها (مثال على ذلك المقارنات السنوية والفصلية).

– **قواعد البيانات كائنية التوجه Object-Oriented Databases**

هي قواعد بيانات تكون فيها المعلومات بشكل كائنات (objects)، تستخدم هذه القواعد في البرمجة كائنية التوجه وتظهر إستعمالاتها في المجالات التطبيقية مثل الهندسة، والمجالات العلمية مثل فيزياء الجسيمات وعلم الأحياء الجزيئي.

– **قواعد البيانات الزمنية Temporal Databases**

دراسة قواعد البيانات الزمنية وتحليلها يكون من أجل التعرف على التغيرات التي تطرأ على الظاهرة الممثلة بهذه البيانات، تحليل أسبابها ونتائجها وتحديد إتجاهها يجعل من الممكن إستخدامها للتقدير والتنبؤ.

– **قواعد البيانات الحيزية Spatial Databases**

تحتوي هذه القواعد على معلومات جغرافية مثل الخرائط، المواقع العالمية والإقليمية وصور من الأقمار الصناعية.

– **قواعد البيانات غير المتجانسة Heterogeneous Databases**

والتي تكون فيها قواعد البيانات ونظام إداراتها مختلفة عن بعضها البعض حيث يتركز الإختلاف في طرق معالجة وتخزين البيانات، مثل الشبكة العنكبوتية العالمية (WWW) والتي تعتبر أكبر مستودع للبيانات غير المتجانسة.

– **قواعد البيانات النصية Textual Databases**

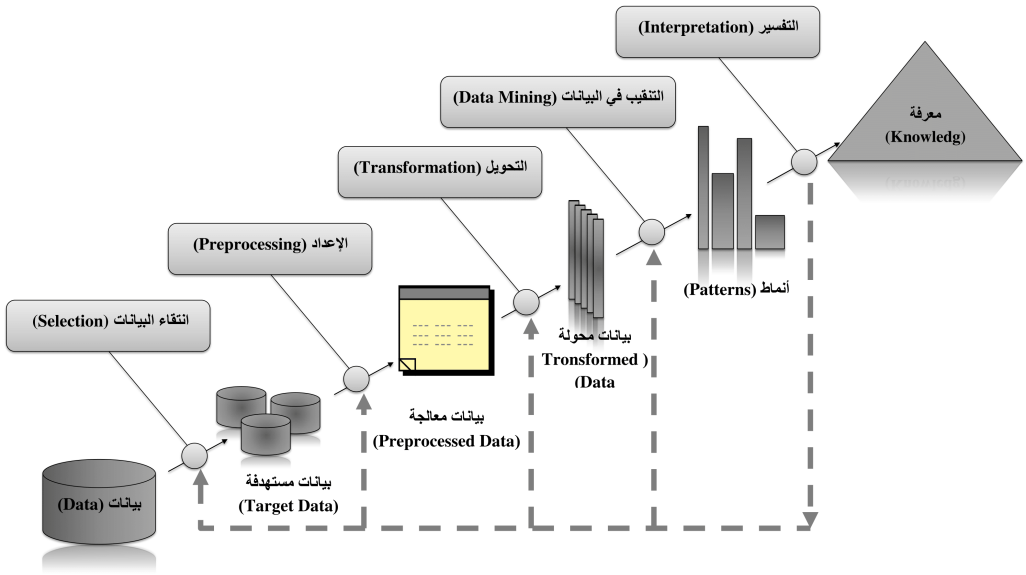
عبارة عن وصف البيانات عن طريق الكلمات والنصوص مثل البريد الإلكتروني وصفحات الويب.

– **قواعد البيانات متعددة الوسائط Multimedia Databases**

تشمل هذه القواعدُ البياناتِ متعددة الوسائط مثل: الفيديو والصوت والصورة، يمكن تخزينها على قواعد البيانات كائنية التوجه أو قواعد البيانات العلائقية، من خصائصها الأبعاد العالية مما يجعل إستخرج البيانات منها صعبا بعض الشيء.

5.1 مراحل إستخلاص المعارف من البيانات

عملية التنقيب في البيانات ليست فقط مشكلة إستخراج النماذج في مجموعة من البيانات، فهذه الأخيرة تُعد خطوة واحدة في العملية التي يتبعها أي شخص يسعى لإستخراج المعرفة من البيانات.
فمن البيانات الخام إلى المعارف المستخرجة، تمر عملية التنقيب في البيانات أو بشكل أدق عملية إستخلاص المعارف من قواعد البيانات عبر عدة مراحل كما يظهره الشكل 1.1 [5].



شكل 1.1: مراحل عملية إستخلاص المعارف من البيانات [5].

1.5.1 مرحلة إنتقاء البيانات Selection

الهدف منها هو تنظيف البيانات الخام (إزالة البيانات غير المفيدة والبيانات غير الجاهزة للإستعمال) من أجل الحصول على بيانات مستهدفة والتي تُجرى عليها عملية إكتشاف المعرفة.

2.5.1 مرحلة الإعداد Preprocessing

تهدف هذه العملية إلى ضمان موثوقية البيانات المتحصّل عليها في المرحلة السابقة، حيث تقوم على عمليتين أساسيتين هما تحديد البيانات الشاذة (outliers detection) وإكتشاف البيانات المفقودة (detect missing data values)، وهما من أكثر العمليات المعروفة في هذه المرحلة.

3.5.1 مرحلة التحويل Transformation

تقوم هذه المرحلة بإعداد البيانات للمرحلة الموالية من خلال تحويل بيانات المرحلة السابقة. يعتمد هذا التحويل على جانبين أساسين هما تسوية البيانات (Data Normalization) وذلك عبر عملية تقليص الأبعاد الخاصة بها (dimension reduction) وتفكيك إرتباط البيانات (Dissociation of Correlated Data)، كما تشمل كلَّ عمليّةٍ من شأنها ملائمة البيانات مع المرحلة الموالية والخوارزمية المنتقاة.

4.5.1 مرحلة التنقيب في البيانات Data Mining

هي جوهر عملية إستخلاص المعارف من البيانات، يتم على ضوئها البحث عن الأنماط المفيدة (useful patterns) وإستخراجها في شكل نماذج توضيحية، تنطوي هذه المرحلة على العديد من الأساليب (methods) المندرجة ضمن تقنيات مختلفة.

5.5.1 مرحلة التفسير Interpretation

هي معاينة الأنماط المستخرجة وتقديم تفسيرات لها قصد الحصول على المعرفة التي يقدمها كل نمط من أجل تحديد الخطوات المتبعة على ضوء هذه المعارف المستخلصة.

من المهم معرفة أن عملية إستخلاص المعارف من البيانات هي عملية مترابطة، بحيث يمكن العودة إلى أي مرحلة من المراحل من أجل القيام بتحسينات (improvements) على عمليات إكتشاف الأنماط وإستخلاص المعارف.

6.1 مهام التنقيب في البيانات

مهام التنقيب في البيانات متنوعة ومستقلة نظرا لوجود العديد من الأنماط في قواعد البيانات الكبيرة. توجد عدة أساليب وتقنيات لإيجاد الأنماط المختلفة، يمكن تلخيص المهام الأكثر إستعمالا في عملية التنقيب في البيانات كما يلي:

1.6.1 التصنيف Classification

نموذج التصنيف يبنى عن طريق تحليل العلاقة بين خصائص الكائنات (attributes) وفئاتها (classes) في مجموعة التدريب (training set)، هذا النموذج يستعمل لتصنيف الكائنات مستقبلا كما يحسن من فهم فئات الكائنات في قواعد البيانات بحيث تكون بيانات الفئة متماثلة فيما بينها ومختلفة عن البيانات في فئات أخرى، كمثال على ذلك مجموعة المرضى الذين تم تشخيص حالتهم الصحية لمرض ما يمثلون مجموعة التدريب، نموذج التصنيف المبني يستنتج الحالة الصحية للمريض إنطلاقا من بيانات التشخيص، هذا النموذج يمكن إستخدامه

لتشخيص الحالة الصحية لمريض جديد بناءً على بيانات التشخيص الخاصة به (العمر، الوزن، الجنس، درجة الحرارة وضغط الدم...) [19].

2.6.1 التقدير Estimation

هي عملية مشابهة للتصنيف، غير أن المتغير المتحصل عليه يكون رقميا (numerical) بدل أن يكون على أصناف أو فئات (classes).
تقوم عملية التقدير بإستكمال القيم المفقودة في نطاق معين للسجل إعتمادا على النطاقات الاخرى التابعة له، فعلى سبيل المثال تقدير ضغط الدم الإنقباضي للمريض يرتكز على العمر، الجنس، مؤشر الوزن ونسبة الصوديوم في دم المريض. فبالتالي تمنح العلاقة بين ضغط الدم الانقباضي والمعطيات في مجموعة التدريب نموذجا للتقدير يمكن تطبيقه على الحالات الجديدة [13].

3.6.1 قواعد الإرتباط Association Rules

قواعد الإرتباط هي التي تسمح بتحديد أي المتغيرات تكون جنبا إلى جنب، وهي من بين مهام التنقيب في البيانات واسعة الإنتشار في عالم الأعمال أين تعرف أكثر بإسم تحليل الائتلاف (affinity analysis) أو تحليل سلة التسوق (Market Basket Analysis) حيث تقوم بالبحث عن نظم (Systems) من أجل حساب العلاقات بين متغيرين أو أكثر. عموما قواعد الإرتباط هي من الشكل: "إذا <سوابق> إذن <نتائج>" [13].

4.6.1 التجميع العنقودي Clustering

هي عملية تشكيل تجمعات من الكائنات التي تكون فئاتها غير معروفة مسبقاً، بحيث تكون التشابهات أكبر ما يمكن داخل التجمع (intra cluster) وأقل ما يمكن ما بين التجمعات (inter cluster) وفقا لمعايير محددة في خصائص هذه الكائنات.
بمجرد تحديد التجمعات تعرّف الكائنات وفقا للتجمع الذي تنتمي إليه، كما تصف الخصائص المشتركة للكائنات في التجمع الواحد ذلك التجمع.
على سبيل المثال يمكن للبنوك أن تقوم بعملية فرز زبائنها إلى عدة تجمعات إعتمادا على التشابهات في خصائص معينة كالعمر، الدّخل ومكان السكن... ، حيث تصف الخصائص المشتركة للزبائن في التجمع الواحد ذلك التجمع من الزبائن.
تساعد هذه التجمعات البنوك في فهم أكبر لزبائنها، بالتالي تقديم عروض ملائمة وخدمات حسب الطلب [19].

7.1 تقنيات التنقيب في البيانات

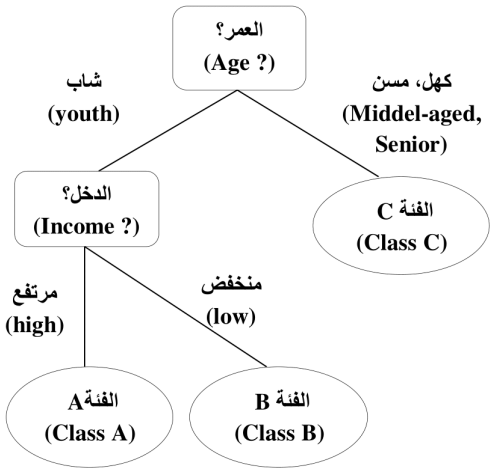
كون التنقيب في البيانات مجالا متعدد التخصصات فهو يتبنى تقنياته من عدة ميادين بحثية (الإحصاء، التعلّم الآلي، أنظمة قواعد البيانات) من أجل إظهار الترابط الكامن داخل البيانات، فإختيار التقنية المناسبة يعتمد على طبيعة البيانات تحت الدراسة وعلى حجمها أيضا [19]. أمثلة عن بعض تقنيات التنقيب في البيانات [10]:

- **شجرة القرارات Decision Tree**

هي تقنية تقوم بالتصنيف من خلال مخطط يعتمد هيكلة شجرية، وذلك عبر تحديد فئة الكائن بإتباع المسار من الجذر وصولا إلى إحدى الأوراق (Leaf Node) بإختيار المسارات وفقا لخصائص قيم الكائن.

كمثال توضيحي يظهر الشكل 2.1 نموذجا لشجرة القرارات، بحيث تقوم هذه الأخيرة بتصنيف الأشخاص في 3 فئات مختلفة: الفئة A، الفئة B أو الفئة C حسب عمر الشخص ثم الدّخل الخاص به، فإذا كان الشخص كهلا أو مسنّا يتم وضعه مباشرة في الفئة C. أمّا إذا كان الشخص شابّا فعندئذ يأتي دور الدّخل في تحديد الفئات، فإذا كان الشخص ذو دخل مرتفع فإنه يصنّف من الفئة A، أمّا إذا كان دخله منخفضا فيتم تصنيفه من الفئة B.

تمتاز شجرة القرارات بسهولة نمذجتها للمعطيات المراد تصنيفها بالإضافة إلى القراءة والتفسير المباشرين للنتائج المتوصل إليها.



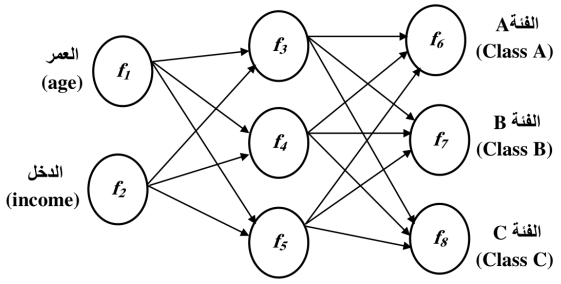
شكل 2.1: نموذج شجرة القرارت Decision Tree [10].

- **الشبكة العصبونية Neural Network**

هي مجموعة من العقد المرتبطة فيما بينها تسمى "عصبونات" وظيفتها مستوحاة من طريقة عمل الشبكة العصبونية البيولوجية، حيث يمكن أن تكون مدخلاتها عبارة عن مخرجات عصبونات أخرى، أو يمكن لها أن تكون خصائص قيم للكائن.

تعتبر الشبكات العصبونية من التقنيات المشهورة في التنقيب في البيانات، حيث تستعمل في التصنيف، التجميع، التنبؤ والتقدير.

يظهر الشكل 3.1 مثالا عن الشبكة العصبونية.



شكل 3.1: مثال عن الشبكة العصبونية Neural Network [10].

- **أمثلة عن تقنيات أخرى**

توجد العديد من الطرق لإنشاء نماذج للتصنيف مثل: نموذج Naive Bayes الذي يفترِض أن الحوادث تكون منفصلة عن بعضها (Independent Events) ويستعمل نظرية Bayes في الإحتمالات.

كما يوجد نموذج Support Vector Machines والذي يقوم بفصل المعطيات بناءً على الخصائص المشتركة لها، وأيضا نموذج K-Nearest-Neighbours للتصنيف.

هذه النماذج تعتمد على معطيات متقطعة (discrete)، ومن أجل المعطيات المتصلة والرقمية (numeric) توجد عمليات أخرى مثل الإنحدار (regression)... .

8.1 إستخدامات التنقيب في البيانات

حققت تطبيقات التنقيب في البيانات نجاحات باهرة في العديد من المجالات كالإقتصاد والعلوم وإدارة الأعمال لتجتاح مجالات جديدة مثل الرياضة بأنواعها إضافة إلى المجالات الطبية المختلفة، ومن الأمثلة العملية لإستخدامات التنقيب في البيانات نجد [19]:

- **التسويق:** يعتبر واحدا من أكثر تطبيقات التنقيب في البيانات نجاحا، فمن خلال التنقيب في قواعد المعاملات الخاصة بالمشتريات السابقة للزبائن يمكن إستخلاص الأنماط والعادات الشرائية من أجل بناء ملفات للزبائن تهدف إلى التسويق الفعال إضافة إلى توقع نسبة تجاوب المستهلكين مع الحملات التسويقية.
- **شركات البطاقات الإئتمانية:** من خلال مستودعات البيانات الكبيرة الخاصة بها، أصبح بإمكان شركات الإئتمان تحديد الزبائن الأكثر ترجيحا للإستفادة من مزايا إئتمانية جديدة.
- **التسوق الإلكتروني:** عند القيام بعملية شراء منتجات من موقع Amazon.com مثلا، فإنّ الموقع يعرِض للمستهلك المنتجات المشابهة لذلك النمط من العادات الشرائية.
- **الطب:** بفضل التنقيب في البيانات إرتفعت على سبيل المثال لا الحصر العلامات الدالة على وجود الخلايا السرطانية في الأنسجة إلى 12 علامة دالّة بعد أن كانت 09 علاماتٍ فقط، وهذا من خلال تحليل ومعالجة صور لعيّنات من الجزء المصاب.
- **مكافحة الجريمة:** تحديد عمليات الشراء المشبوهة (غير الإعتيادية) كجزء من عمليات كشف الإحتيال وكشف سرقة البطاقات الإئتمانية.

9.1 خاتمة

في هذا الفصل تم التطرق إلى المفاهيم الأساسية للتنقيب في البيانات كما تم التوصل إلى أنه يمكن أن يساهم التنقيب في البيانات في حلَّ مشاكل تحليل البيانات التي تواجهها العديد من المؤسسات والشركات، وهذا ما يجعل له أهمية كبرى في الحاضر والمستقبل.

أمَّا في الفصل الموالي، فستتمحور الدراسة حول مفهوم تحليل التّرابط وقواعده وخاصة مجموعات العناصر المتكررة (Frequent Itemsets) والتعداد الكلي لها عبر دراسة خوارزميات لمختلف المقاربات الموجودة، إضافة إلى كيفية إنشاء قواعد الإرتباط (Association Rules) من خلال هذه المجموعات.

الفصل الثاني

قواعد الإرتباط والتعداد الكلي لمجموعات العناصر المتكررة

1.2 مقدمة

ينصب إهتمام المتخصصين في تكنولوجيا المعلومات حاليا حول مجال التنقيب في البيانات بمختلف ميادينه، خاصة حول عملية إستخلاص قواعد الإرتباط
(Association Rules Mining) والتي تُستخدم اليوم في العديد من مجالات الحياة.

وبالعودة إلى الوراء قليلا، فإن مفهوم قواعد الإرتباط ظهر لأول مرة من خلال عملية تحليل سلة التسوق (Market Basket Analysis) بهدف فهمٍ أفضل للعادات الشرائية للزبائن، وبالتالي تصنيفٍ أمثلٍ للمنتجات في المراكز التجارية، وإطلاق عروض ترويجية وأسعار تشجيعية مدروسة.

تنقسم عملية البحث في قواعد المعاملات (Transactional Databases) عن قواعد الإرتباط إلى قسمين هما:

- إيجاد جميع مجموعات العناصر المتكررة: أي البحث عن جميع المجموعات التي يكون تكرارها مساويا أو أكبر من عتبة محددة مَا.
- إنشاء قواعد الإرتباط: يتم إنشاء قواعد الإرتباط من خلال المجموعات المتكررة المتحصل عليها بحيث يتوجب أن تستوفي بعض الشروط المحددة مُسبقا.

عملية إيجاد مجموعات العناصر المتكررة الكلية هي العملية الأكثر تكلفةً من ناحية الزمن المستغرق والمساحة التخزينية للذاكرة مقارنة بعملية إنشاء قواعد الإرتباط، ومن أجل هذا قام الباحثون على مدى عِقدين من الزمن وإلى الآن بإقتراح مقاربات وخوارزميات تعمل على تحسين

تكلفة عملية إيجاد مجموعات العناصر المتكررة الكلية في مجموعة بيانات كبيرة جدا، وهذا ما ستتركز عليه الدراسة فيما يأتي.

بالتالي سيتمحور هذا الفصل حول المقاربات والخوارزميات المقترحة والمستعملة حاليا في إيجاد مجموعات العناصر المتكررة الكلية، كالمقاربة البديهية (Naïve Approach)، والمقاربة بالمستويات (Level-wise Approach)، المقاربة العمودية (Vertical Approach) إضافة إلى المقاربة بالإسقاط (Projection Approach) وكل مايتعلق بها من مصطلحات ومفاهيم أولية. ومن ثمَ تسليط الضوء على عملية إنشاء قواعد الإرتباط إنطلاقا من مجموعات العناصر المتكررة المتحصل عليها.

2.2 مجموعات العناصر المتكررة وقواعد الإرتباط
Frequent Itemsets and Association Rules

تعتبر قواعد الإرتباط أحد أهم الأدوات في عملية إستخراج المعارف من البيانات كونها تُعنى بتصفح كميات هائلة من البيانات الموجودة داخل قواعد بيانات ضخمة من أجل إكتشاف علاقات أو إرتباطات فيما بينها إعتمادا على وجود خصائص وصفات أخرى.

قواعد بيانات المراكز التجارية هي إحدى قواعد البيانات الضخمة حيث أنها تجمع كمّا معتبرا من البيانات حول مشتريات الزبائن المتنوعة من خلال تسجيل "إيصال الدفع" (receipt) الخاص بكل زبون [13].

الجدول 1.2 يبين جزءً من عمليات الشراء في أحد محلات المراكز التجارية، حيث تمثل الأسطر عمليات الشراء والتي تتضمن رقم عملية الشراء، وقائمة بالأصناف التي تم شراؤها. (لأهداف الدراسة، لن يكون الإهتمام حول الكمية المشتراة لكل صنف بل سينصب حول تواجد الصنف في القائمة أم لا) [13].

رقم عملية الشراء	قائمة الأصناف المشتراة
1	{ بروكلي، فلفل، ذرة }
2	{ هليون، كوسا، ذرة }
3	{ ذرة، طماطم، فاصوليا خضراء، كوسا }
4	{ فلفل، ذرة، طماطم، فاصوليا خضراء }
5	{ فاصوليا خضراء، هيليون، بروكلي }
6	{ كوسا، هيليون، فاصوليا خضراء، طماطم }
7	{ طماطم، ذرة }

جدول 1.2: عمليات الشراء في أحد المحلات التجارية.

نظريا، يُعبر عن الأصناف المشتراة بـالعناصر(Items)، بينما تأخذ عمليات الشراء إسم المعاملات (Transactions)، وجميعها تكون محتواة داخل قاعدة معاملات (Transactional Database) [13].

3.2 مجموعة العناصر ومجموعة معرِّفات المعامَلات Itemsets and Tidsets

لتكن المجموعة $I = \{x_1, x_2, x_3, x_4, ..., x_m\}$ والتي تحتوي على m عنصرا، تمثل جميع العناصر (Items) في قاعدة المعاملات [20]:

- مجموعة العناصر (Itemset): هي مجموعة جزئية X محتواة أو تساوي المجموعة الكلية للعناصر I، يعبر عن ذلك بالعلاقة $X \subseteq I$.
  مجموعة العناصر ذات K عنصرا هي المجموعة التي تحتوي على k عنصرا وتسمى (k-itemset)، مثال على ذلك: المجموعة { بروكلي، فلفل، ذرة } هي مجموعة ذات 3 عناصر (3-itemset).

ولتكن المجموعة $T = \{t_1, t_2, t_3, t_4, ..., t_n\}$ الممثلة لمعرِّفات المعامَلات $(Tids)$ والتي تحتوي على n معرفا [20]:

- مجموعة معرِّفات المعامَلات (Tidsets): هي المجموعة الجزئية **T** المحتواة داخل مجموعة معرِّفات المعامَلات T بحيث ($\mathbf{T} \subseteq T$)
- المعاملة (Transaction) هي عبارة عن ثنائية من الشكل $< t, X >$، بحيث [20]:
  - $t \in T$ والذي هو بمثابة معرف وحيد للمعاملة في قاعدة المعاملات.
  - X يمثل مجموعة من العناصر.
  - يشار الى المعاملة $< t, X >$ بمعرفها t.

4.2 أنواع تمثيل قاعدة المعاملات

تتشكل قاعدة المعاملات من مجموعة متسلسلة من المعاملات التي يرمز لها بالثنائية $< t, X >$، في المثال أدناه الموضّح من خلال الجدول 2.2 تكون مجموعة العناصر I ومجموعة معرِّفات المعامَلات T هما على الترتيب: $I = \{A, B, C, D, E\}$ و $T = \{1, 2, 3, 4, 5, 6\}$، فيظهر الجدول 2.2.أ مثالا على قاعدة المعاملات للمجموعتين I و T [20].

يمكن القيام بتحويلات على هذه القاعدة لتصبح قاعدة معاملات ثنائية الشكل (Binary) كما يظهره الجدول 2.2.ب بحيث تظهر المعاملات مؤشرة بالرمزين "0" و "1"، فظهور الرمز "1" يمثل وجود العنصر أما غياب العنصر فيعبر عنه بظهور الرمز "0".

كما يمكن أيضا تحويل قاعدة المعاملات لتصبح قاعدة معاملات عمودية (Vertical) مثل ما يظهره الجدول 2.2.ج.

t	$i(t)$
1	$ABDE$
2	BCE
3	$ABDE$
4	$ABCE$
5	$ABCDE$
6	BCD

أ: قاعدة المعاملات.

D	A	B	C	D	E
1	1	1	0	1	1
2	0	1	1	0	1
3	1	1	0	1	1
4	1	1	1	0	1
5	1	1	1	1	1
6	0	1	1	1	0

ب: قاعدة المعاملات الثنائية.

X	A	B	C	D	E
$t(X)$	1	1	2	1	1
	3	2	4	3	2
	4	3	5	5	3
	5	4	6	6	4
		5			5
		6			

ج: قاعدة المعاملات العمودية.

جدول 2.2: أنواع تمثيل قاعدة المعاملات.

حيث تمثّل:

$i(t)$: مجموعة العناصر المحتواة في معاملة t $(t \in T)$

$t(X)$: مجموعة المعاملات التي تحتوي العناصر الموجودة في مجموعة العناصر X

إنطلاقا من الجدول 2.2.ج فإن مجموعة معرِّفات المعامَلات المقابلة للعنصر A يمكن كتابتها على الشكل $< A, \{1, 3, 4, 5\} >$، كما يمكن كتابتها أيضا على الشكل التالي $< A, 1345 >$. وفيما يأتي، ستتم كتابة الترميز الدال على مجموعة معرِّفات المعامَلات بالشكل $< A, 1345 >$، كما ستكتب من أجل مجموعة العناصر على الشكل ABC بدلا من $\{A, B, C\}$.

5.2 الداعم ومجموعات العناصر المتكررة Support and Frequent Itemsets

من أجل فهم عملية تطبيق قواعد الارتباط في قواعد المعاملات، يتوجب الإلمام ببعض المفاهيم من بينها [20]:

- الداعم المطلق (Support) لمجموعة العناصر X الموجودة في قاعدة المعاملات D والذي يرمز إليه بـ $sup(X, D)$ هو عبارة عن عدد المعاملات التي تحتوي مجموعة العناصر X في تلك القاعدة، ويعطى بالعلاقة التالية:

$$sup(X, D) = |\{t \backslash < t, i(t) > \in D \quad and \quad X \subseteq i(t)\}| = |t(X)|$$

حيث يمثل الرمز | |: عدد عناصر المجموعة (cardinality of the set).
مثلا: في الجدول 1.2 الداعم لمجموعة العناصر { طماطم، ذرة } يساوي 3.

- الداعم النسبي (Relative Support) لمجموعة العناصر X هو نسبة حاصل قسمة الداعم المطلق لمجموعة العناصر X على العدد الإجمالي للمعاملات في القاعدة D ويعطى بالعلاقة:

$$rsup = \frac{sup(X, D)}{|D|} \qquad , \qquad |D| = |T|$$

بالإسقاط على نفس المثال في الجدول 1.2 الداعم النسبي لمجموعة العناصر { طماطم، ذرة } يساوي $\frac{3}{7}$ أي 0.42.

- مجموعات العناصر المتكررة (Frequent Itemsets): يقال عن مجموعة العناصر X أنها **مجموعة عناصر متكررة** في قاعدة المعاملات D إذا كان الداعم لهذه المجموعة أكبر من أو يساوي قيمة عددية معرفة من قبل المستخدم وتمثل العتبة الدنيا للداعم (minimum support threshold)، يرمز لها بـ μ:

$$\text{X is a Frequent Itemset} \equiv \text{sup(X, D)} \geq \mu$$

بالعودة إلى الجدول 2.2.أ وبفرض أن العتبة الدنيا للداعم تساوي 3 أي $\mu = 3$
(في الداعم النسبي: قيمة العتبة الدنيا لهذا الداعم تساوي إلى 0.5).
مجموعة العناصر BCE من خلال الجدول محتواة داخل معرفات المعاملات 2، 4 و 5 على التوالي، هذا يعني أن $< BCE, 245 >$ ويحسب الدّاعم الخاص بهذه المجموعة كالتالي: $sup(BCE) = |t(BCE)| = 3$ أي أن $sup(BCE) \geq \mu$، بالتالي تكون المجموعة BCE مجموعة عناصر متكررة.

6.2 قواعد الإرتباط Association Rules

قاعدة الإرتباط هي العلاقة من الشكل $X \to Y$، أين تكون X و Y مجموعتين جزئيتين من المجموعة الكلية للعناصر I تقاطعهما يعطي مجموعة خالية وكلاهما لا يمثل مجموعة خالية. بكتابة رياضية يعبر عن ذلك بـ: $X, Y \subseteq I, X \cap Y = \varnothing, X, Y \neq \varnothing$ [20].
يسمى الطرف الأيسر من القاعدة بـ المقدمة (Premise) بينما يسمى الطرف الأيمن منها بـ النتيجة (Consequence) [20].

- داعم قاعدة الإرتباط $sup(X \to Y)$ هو مؤشر يدل على موثوقية قاعدة الإرتباط. يحسب بعدد المعاملات التي تظهر فيها كل من المجموعتين X و Y بجميع عناصرهما في قاعدة المعاملات D كالتالي [20]:

$$s = sup(X \to Y) = |t(XY)| = sup(XY)$$

أين يدل الرمز XY على إتحاد المجموعتين الجزيئتين X و Y أي $X \cup Y$.

- الداعم النسبي لقاعدة الإرتباط يعرف على أنه نسبة حاصل قسمة المعاملات التي تحتوي على مجموعتي العناصر X و Y معا على العدد الكلي للمعاملات في قاعدة المعاملات D، يقدم الداعم النسبي تقديرا للإحتمالات المشتركة لمجموعتي العناصر X و Y، يعبر عنه كالآتي: [20]

$$rsup(X \to Y) = \frac{sup(XY)}{|D|} = p(X \wedge Y)$$

- معامل الثقة (confidence): هو مؤشر يدل على مدى قوة ودقة قاعدة الإرتباط، بلغة الإحتمالات يعرف معامل الثقة على أنه الإحتمال الشرطي لأن تكون المعاملة تحتوي على Y بالنظر إلى كونها تحتوي أيضا على X.
بمعنى آخر يعرف معامل الثقة على أنه نسبة حاصل قسمة طرفي قاعدة الإرتباط على الطرف الأيسر منها (أي المقدمة)، يعطى معامل الثقة بالعلاقة التالية [20]:

$$c = conf(X \to Y) = p(Y|X) = \frac{p(X \wedge Y)}{p(X)} = \frac{sup(XY)}{sup(X)}$$

- العتبة الدنيا لمعامل الثقة (minimum confidence threshold) هو مقدار يأخذ قيمة الحد الأدنى لمعامل ثقة قاعدة الإرتباط، يكون محددا من قبل المستخدم ويرمز له بـ γ.
تكون قاعدة الإرتباط $X \to Y$ متكررة إذا كانت عناصر المجموعتين XY متكررة، بمعنى آخر داعم هذه القاعدة يكون أكبر من العتبة الدنيا المحددة للداعم $sup(XY) \geq \mu$.

كما تكون قاعدة الإرتباط قوية ومهمة إذا كان معامل ثقتها أكبر من العتبة الدنيا المحددة لمعامل الثقة $conf(X \rightarrow Y) \geq \gamma$.
بعبارة أخرى: قاعدة إرتباط جيدة تعني أن الداعم ومعامل الثقة مرتفعان [20].

تجدر الإشارة إلى أن العتبة الدنيا للداعم μ وكذا موثوقية القاعدة γ تحدد حسب ميدان التطبيق، فمحللو البيانات لقواعد الإرتباط الخاصة بالمشتريات في المراكز التجارية يحددون قيما مرتفعة لـ μ و γ نظرا لوجود عدد كبير من المعاملات في القاعدة، بالمقابل فإن محللي البيانات الخاصة بكشف الإحتيال أو العمليات الإرهابية الوشيكة يستعملون قيما منخفضة لـ μ و γ بسبب حساسية ميدان التطبيق (قلة عدد المعاملات المرتبطة بهذا المجال) [13].

مثال: في الجدول 2.2.أ، و بإعتبار $BC \rightarrow E$ قاعدة إرتباط، فإن حساب الداعم ومعامل الثقة لهذه القاعدة يكونان كالآتي:

$$s = sup(BC \rightarrow E) = sup(BCE) = 3$$
$$c = conf(BC \rightarrow E) = \frac{sup(BCE)}{sup(BC)} = \frac{3}{4} = 0.75$$

بفرض أن العتبة الدنيا للداعم تساوي 3 بمعنى $\mu = 3$، والعتبة الدنيا لمعامل الثقة تساوي 0.7 أي $\gamma = 0.7$ وإنطلاقا من العمليات الحسابية أعلاه، بما أن $s \geq \mu$ و $c \geq \gamma$ فإن قاعدة الإرتباط $BC \rightarrow E$ متكررة، دقيقة وذات موثوقية .

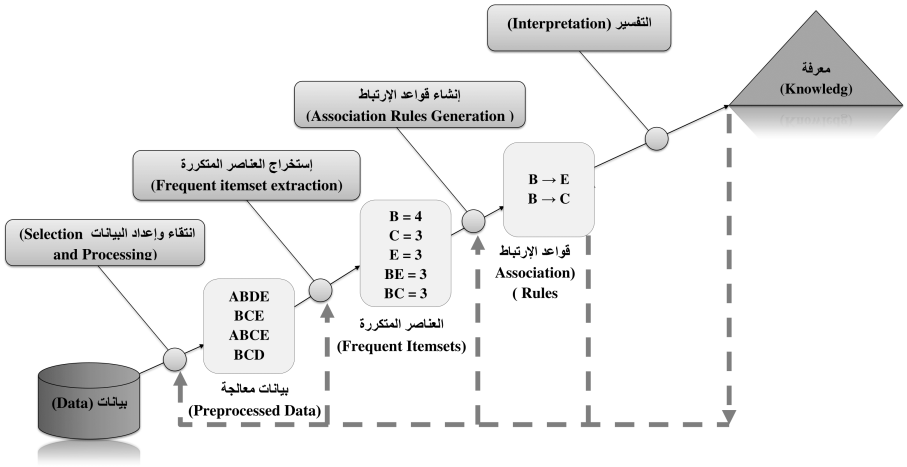
7.2 مجموعة العناصر والتنقيب عن قواعد الإرتباط Itemset and Rule Mining

يحتاج إنشاء قواعد إرتباط متكررة وذات موثوقية عالية إلى تعداد كلي لمجموعة العناصر مع حساب قيمة الداعم الخاص بكل مجموعة [20].

يتم التنقيب عن مجموعة العناصر المتكررة في قاعدة المعاملات D، بتعداد كل مجموعات العناصر X التي يكون الداعم الخاص بها أكبر من أو يساوي العتبة الدنيا للداعم بحيث: $sup(X, D) \geq \mu$ [20].

بعد نهاية العملية تتكون مجموعة تحوي جميع العناصر المتكررة يرمز لها بـ F، وإنطلاقا من هذه المجموعة يتم التنقيب عن جميع قواعد الإرتباط المتكررة والقوية التي يكون معامل الثقة الخاص بها مساويا أو أكبر من العتبة الدنيا لمعامل الثقة المحددة من قبل المستخدم: $conf(X \rightarrow Y) \geq \gamma$ [20].

يظهر الشكل 1.2 المراحل المختلفة لعملية التنقيب عن قواعد الإرتباط.



شكل 1.2: مراحل عملية التنقيب عن قواعد الإرتباط (Association Rules Mining) [14].

8.2 مقاربات عملية إستخراج مجموعات العناصر المتكررة

من أجل إستخراج مجموعات العناصر المتكررة في قواعد المعاملات، توجد أنواع من المقاربات هي [20]:

1.8.2 المقاربة البديهية Naïve Approach or Brute-force Approach

يتم على مستواها تعداد جميع العناصر الممكنة والتي تسمى بالعناصر المرشحة، ومن ثم حساب قيمة الداعم الخاص بكل عنصر والإبقاء فقط على العناصر المتكررة.

ا) **الخوارزمية البديهية Brut-force Algorithm**

الخوارزمية 1 أدناه توضح من خلال سلسلة الأوامر البرمجية آلية عمل الخوارزمية البديهية (Brute-force Algorithm).

خوارزمية 1 سلسلة الأوامر البرمجية للخوارزمية البديهية (Brute-Force Algorithm).

خوارزمية	Brute-force (D, I, μ)
المدخلات	العتبة الدنيا للداعم μ، المجموعة الكلية للعناصر I، قاعدة المعاملات D
المخرجات	مجموعة العناصر المتكررة F
المبدأ	مجموعة العناصر المتكررة تبدأ بالمجموعة الخالية $F \longleftarrow \emptyset$
1	**foreach** X $\subseteq I$
2	sup(X) $\leftarrow$ ComputeSupport(X,D)
3	**if** sup(X) $\geq \mu$ **then**
4	$F \leftarrow F \cup \{(X, Sup(X))\}$
5	return F
	ComputeSupport(X,D)
6	sup(X) $\leftarrow 0$
7	**foreach** $\{t, i(t)\} \in D$ //t مجموعة العناصر المرتبطة بـ المعاملة $i(t)$
8	**if** X $\subseteq i(t)$ **then**
9	sup(X) $\leftarrow$ sup(X)+1
10	return sup(X)

ب) **التعقيد الحسابي للخوارزمية algorithm complexity**

تعداد العناصر المرشحة: من أجل مجموعة I يوجد $2^{|I|}$ مرشحا محتملا، هذا يعني أن التعقيد الحسابي لعملية التعداد الكلي للعناصر هو $O(2^{|I|})$.

حساب الداعم: تتطلب عملية حساب الداعم لعنصر مَا وتحديد ما إذا كان هذا العنصر متكررا أم لا مسحا كليا لقاعدة المعاملات، وهذا يحتاج إلى $O(|I||D|)$ في أسوأ الحالات.

إذن التعقيد الحسابي للخوارزمية البديهية (Brute-force Algorithm) هو:
$O(|I||D|2^{|I|})$.

بالتالي فإن هذه المقاربة من الناحية الحسابية غير قابلة للتطبيق حتى مع فضاء صغير لمجموعة العناصر، بينما في الواقع يمكن للمجموعة الكلية للعناصر I أن تكون كبيرة جدا (أوضح مثال على ذلك هو المراكز التجارية التي تحوي آلاف المنتجات).

2.8.2 المقاربة بالمستويات Level-wise Approach

توجد العديد من الخوارزميات التي تندرج تحت نطاق المقاربة بالمستويات (لا يتسع المجال هنا لذكرها جميعا)، لكن خوارزمية Apriori المقترحة من قبل Agrawal سنة 1994 [2] تعد

من أولى الخوارزميات لهذه المقاربة أو بالإمكان إعتبارها الخوارزمية المرجع.

تقوم خوارزمية Apriori على فكرة إستخلاص العناصر المتكررة من قواعد المعاملات بهدف إشتقاق قواعد إرتباط من هذه العناصر، بإعتماد خاصيتين أساسيتين هما:

الخاصية 1: كل المجموعات الجزئية (Subsets) لمجموعة عناصر متكررة تعتبر مجموعات متكررة، فمثلا إذا كانت X مجموعة عناصر متكررة فإن أي مجموعة جزئية Y محتواة في X $(Y \subset X)$ تعتبر مجموعة عناصر متكررة أيضا.

الخاصية 2: كل مجموعة عليا (Superset) لمجموعة عناصر غير متكررة ليست مجموعة عناصر متكررة، فإذا كانت X مجموعة عناصر غير متكررة فإن إتحاد أي مجموعة عليا Y مع المجموعة X لا تجعل منها مجموعة عناصر متكررة.

هاتان الخاصيتان مفيدتان في تقليص فضاء الدراسة بشكل كبير لخوارزمية Apriori.

ا) آلية عمل خوارزمية Apriori

تعمل خوارزمية Apriori بطريقة تكرارية، فبداية يتم تعداد جميع عناصر المجموعة الكلية في قاعدة المعاملات بشكل فردي (singelton) ثم يتم حساب الداعم الخاص بكل عنصر عن طريق مسح شامل لقاعدة المعاملات D، وهذا ما يسمى بمرحلة إنشاء المجموعة المرشحة (candidate set) ذات عنصر واحد (1-itemset)، تسمى هذه المجموعة بـ (C_1).

بعد ذلك يتم حذف كل عنصر مرشح من المجموعة (C_1) يكون داعمه أقل من العتبة الدنيا المحددة للداعم μ لأنه لن يكون عنصرا متكررا، ويتم الإبقاء فقط على العناصر المرشحة التي تحمل داعما أكبر من أو يساوي العتبة الدنيا المحددة للداعم، لتشكل مجموعة العناصر المتكررة (L_1) ذات عنصر واحد (1-itemset).

ثم يتم إنشاء المجموعة المرشحة (C_2) ذات عنصرين (2-itemset) من خلال ربط العناصر المتكررة ذات عنصر واحد الموجودة في المجموعة (L_1) مع نفسها، ثم بعد التحقق من داعم كل عنصر مرشح بعد المسح الكلي لقاعدة المعاملات D، يتم حذف العناصر المرشحة غير المتكررة، والإبقاء على العناصر المرشحة المتكررة لإنشاء مجموعة العناصر المتكررة (L_2) ذات عنصرين (2-itemset).

وهكذا تتواصل نفس آلية العمل مادامت مجموعة العناصر المتكررة (L_i) تحمل أكثر من مجموعة واحدة.

الملاحظ هنا أن عملية إنشاء مجموعة العناصر المرشحة تعتمد أساسا على خطوتين هما:

الإقتران (Join): هو ربط مجموعة متكونة من $k-1$ عنصرا متكررا (k-1 frequent itemset) مع نفسها، يؤدي هذا الى إنشاء مجموعة مكونة من k عنصرا مرشحا (k-candidates).

التقليم (Pruning): تتم إزالة المجموعات المرشحة التي يكون لديها على الأقل مجموعات جزئية تعتبر عناصر غير متكررة في المجموعة ذات $k-1$ عنصرا متكررا (k-1 frequent itemset).

ب) مثال يوضح آلية عمل خوارزمية Apriori

الشكل 2.2 أدناه يصف مثالا عمليا لطريقة عمل خوارزمية Apriori، إستنادا إلى المعاملات الموجودة في قاعدة المعاملات الموضحة من خلال الجدول 2.2.أ.
في هذا المثال تحدد قيمة العتبة الدنيا للداعم بـ 3 ($\mu = 3$)، هذا يعني أن مجموعة العناصر تكون متكررة إذا ظهرت ثلاث مرات أو أكثر في قاعدة المعاملات:

التكرار الأول (First Iteration)
يتم تعداد العناصر الموجودة في قاعدة المعاملات بشكل فردي (singelton)، وحساب الداعم الخاص بكل عنصر، هذا ما ينتج عنه مجموعة العناصر المرشحة ذات عنصر واحد C_1 والتي هي كالتالي $C_1 = \{A, B, C, D, E\}$، ومن ثم الإبقاء على العناصر التي يكون الداعم الخاص بها مساويا أو أكبر من العتبة الدنيا المحددة μ وإقصاء جميع العناصر ذات الداعم الأقل من قيمة العتبة الدنيا من أجل تشكيل مجموعة العناصر المتكررة ذات عنصر واحد L_1، في المثال هنا جميع العناصر الفردية هي عناصر متكررة، لذا فإن المجموعة L_1 تكون كالتالي $L_1 = \{A, B, C, D, E\}$

التكرار الثاني (Second Iteration)
يتم إنشاء المجموعة المرشحة ذات عنصرين C_2 إنطلاقا من إقتران العناصر المتكررة ذات عنصر واحد الموجودة في المجموعة L_1 مع نفسها، لتكون C_2 كالتالي
$$C_2 = \{AB, AC, AD, AE, BC, BD, BE, CD, CE, DE\}$$
ثم بنفس العملية المتبعة سابقا، بعد مسح قاعدة المعاملات وعلى أساس قيمة الداعم الخاص بكل مجموعة عناصر مرشحة من C_2 يتم حذف مجموعة العناصر ذات الداعم الأقل من العتبة الدنيا μ والإبقاء على مجموعة العناصر ذات الداعم الأكبر من μ لإنشاء مجموعة العناصر المتكررة L_2 ذات عنصرين.
يمكن ملاحظة أن الخانات المظللة باللون الرمادي في جداول الشكل 2.2 تمثل مجموعة العناصر المحذوفة من مجموعة العناصر المرشحة.
تكون المجموعة L_2 إذن كما يلي $L_2 = \{AB, AD, AE, BC, BD, BE, CE, DE\}$

التكرار الثالث (Third Iteration)

تتشكل مجموعة العناصر المرشحة ذات 3 عناصر C_3 إنطلاقا من إقتران عناصر المجموعة L_2 مع نفسها لتصبح كالتالي

$$C_3 = \{ABD, ABE, ADE, BCE, BDE\}$$

وبعد المسح الموالي لقاعدة المعاملات من أجل حساب الداعم الخاص بكل مجموعة من العناصر المرشحة وحذف العناصر ذات الداعم الأقل من العتبة الدنيا المحددة والإبقاء فقط على العناصر المتكررة، تكون مجموعة العناصر المتكررة ذات 3 عناصر كالتالي

$$L_3 = \{ABD, ABE, ADE, BCE, BDE\}$$

التكرار الرابع (Fourth Iteration)

تتشكل مجموعة العناصر المرشحة ذات 4 عناصر C_4 إنطلاقا من إقتران عناصر المجموعة L_3 مع نفسها لتصبح كالتالي $C_4 = \{ABDE, BCDE\}$

وبعد إعادة مسح قاعدة المعاملات D من أجل حساب الداعم الخاص بكل مجموعة من العناصر المرشحة حيث يتم تصفية العناصر ذات الداعم الأقل من العتبة الدنيا المحددة، والإبقاء فقط على العناصر المتكررة لتكون مجموعة العناصر المتكررة ذات 4 عناصر كالتالي $L_4 = \{ABDE\}$.

التكرار الخامس (Fifth Iteration)

بما أن مجموعة العناصر المتكررة ذات 4 عناصر L_4 تحتوي على مجموعة واحدة فقط، فإنه لا يمكن إنطلاقا منها إنشاء مجموعة مرشحة C_5 ذات 5 عناصر (5-itemset). ومن ثم فإن عملية التنقيب عن مجموعة العناصر المتكررة تنتهي عند مجموعة العناصر المتكررة L_4.

شكل 2.2: آلية عمل خوارزمية Apriori.

ج) خوارزمية Apriori

الخوارزمية 2 أدناه توضح آلية عمل الخوارزمية Apriori من خلال سلسلة الأوامر البرمجية [2].

خوارزمية 2 سلسلة الأوامر البرمجية لخوارزمية Apriori.

خوارزمية	Apriori (D, μ)
المدخلات	العتبة الدنيا للداعم μ، قاعدة المعاملات D
المخرجات	مجموعة العناصر المتكررة F
المبدأ	مجموعة العناصر المتكررة تبدأ بالمجموعة الخالية $F \longleftarrow \emptyset$
1	$F_1 = \{1 - frquent \quad itemsets\}$
2	$L_1 \longleftarrow F_1$
3	**for** $(k = 2;\ L_{k-1} \neq \emptyset; k++)$**do begin**
4	$C_k =$ Apriori-Gen(L_{k-1})
5	**for all** transactions $t \in D$ **do begin**
6	$C_t =$ subset(C_k,t)// $C_k = \{c \in C_k, c \subseteq t\}$
7	**foreach** candidate $c \in C_t$ **do**
8	sup(c)++
9	$L_k = \{c \in C_k \vert sup(c) \geqslant \mu\}$
10	return $F = \cup_k\ L_k$
	function Apriori-Gen(L_{k-1})
11	**insert into** C_k
12	**select** p [1], p [2],..., p $[k-1]$, q $[k-1]$
13	**from** $L_{k-1}\ p, L_{k-1}\ q$
14	**where** p [1] = q [1] ... p $[k-2]$ = q $[k-2]$ p $[k-1]$ < q $[k-1]$
	// تقليم كل العناصر الذين يكون لديهم على الأقل مجموعات جزئية تعتبر عناصر غير متكررة في المجموعة ذات $k-1$ عنصرا متكررا.
15	**forall** itemsets $c \in C_k$ **do**
16	**forall** $(k-1)$ subsets s of c **do**
17	**if** $s \notin L_{k-1}$ **then** delete c from C_k
18	return C_k
19	end

د) التعقيد الحسابي للخوارزمية algorithm complexity

يبقى التعقيد الحسابي لخوارزمية Apriori في أسوأ الحالات $O(|I||D|2^{|I|})$ في حالة ما إذا كانت جميع عناصر المجموعة متكررة، لكن عمليا تكون تكلفة الخوارزمية أقل بكثير وهذا بفضل عملية التقليم (Pruning) لفضاء الدراسة.

ويستلزم خوارزمية Apriori من أجل تعداد العناصر المرشحة في أسوأ الحالات $O(|I|)$ مسحا لقاعدة المعاملات، على عكس الخوارزمية البديهية التي تتطلب $O(2^{|I|})$ مسحا من أجل ذلك. ولكن في المجال التطبيقي يتطلب الأمر في المجمل تكلفة تقدر بطول أطول مجموعة عناصر متكررة، حيث تأخذ العملية l مسحا لقاعدة المعاملات أين يمثل l الطول الخاص بأطول مجموعة عناصر متكررة. **(يمثل l أيضا آخر مستوى Level وصلت إليه الخوارزمية).**

على الرغم من كون خوارزمية Apriori من أشهر الخوارزميات المعروفة في مجال التنقيب في البيانات بفضل بساطتها وبالتالي سهولة تنفيذها، وأيضا بفضل خصائصها التي تؤدي إلى تقليص فضاء الدراسة من أجل إيجاد مجموعات العناصر المتكررة بشكل كبير مقارنة بالخوارزمية البديهية، إلا أنها تشكو من بعض العيوب التي تحد من إستعمالاتها كالحاجة إلى المسح الكلي لقاعدة المعاملات في كل تكرار من أجل حساب قيم الداعم وهذا ما يجعلها أقل فاعلية مع البيانات ذات الأبعاد الكبيرة.

3.8.2 المقاربة العمودية Vertical Approach

خوارزمية ECLAT والتي تعني (Equivalence CLAss Transformation) هي من أولى الخوارزميات التي إنتهجت المقاربة العمودية أو المقاربة نحو العمق، قُدّمت أول مرة من قبل الباحثين Mohammed J.Zaki, Parthasarathy, Li and Ogihara من خلال مجموعة من الأبحاث تم عرضها سنة 1997 [22]، وتعتبر Eclat أول خوارزمية تعتمد الشكل العمودي لقاعدة المعاملات (Vertical Transactional Base)، الأمر الذي أدى بالإضافة إلى إستخدام إستراتيجية البحث نحو العمق (Depth-First search) وإستخدام تقاطع مجموعات معرِّفات المعامَلات (Tidsets Intersection) إلى تحسين عملية حساب الداعم بشكل كبير.

تتم عملية حساب الداعم (support) لمجموعة العناصر المرشحة عن طريق تعداد أفراد المجموعة الناتجة عن تقاطع قوائم معرِّفات المعامَلات (Lists of Tidsets) للمجموعات الجزئية الموافقة لهذه المجموعة من العناصر المرشحة.

بشكل عام وبفرض أن X و Y يعبران عن مجموعتي عناصر متكررة، معرِّفات معامَلاتها على الترتيب $t(X)$ و $t(Y)$ فإن مجموعة معرِّفات المعامَلات الخاصة بالمجموعة العليا التي تضمهما يعبر عنها كالتالي: $t(XY) = t(X) \cap t(Y)$، ويحسب الداعم الخاص بها عن طريق إيجاد عدد عناصر مجموعة معرفاتها **(أي معرفات المجموعة العليا)** (the cardinality of its Tidset)، ويكتب على الشكل: $sup(XY) = |t(XY)|$.

مثـال: إذا كانت مجموعة معرِّفات المعامَلات الخاصة بالعنصرين A و B هما $< t(A), 1345 >$ و $< t(B), 2456 >$ على الترتيب، فبالإمكان تحديد مجموعة معرِّفات المعامَلات الخاصة

بالمجموعة العليا لهما عن طريق تقاطع مجموعتي معرِّفات المعامَلات $t(A)$ و $t(B)$ بحيث تكون:

$$t(AB) = t(A) \cap t(B) = 1345 \cap 2456 = 45$$

وتحسب قيمة الداعم الخاص بها كما يلي: $sup(AB) = |45| = 2$

خوارزمية Eclat تقوم بعملية تقاطع معرِّفات المعامَلات (Tidsets intersection) فقط إذا كانت مجموعة العناصر المتكررة تحمل نفس البادئة (common prefix)، وتجوب الخوارزمية محتفظة بالبادئة قاعدة المعاملات بطريقة البحث نحو العمق (Depth-First Search) لمعالجة مجموعة العناصر التي تحمل نفس البادئة، والتي يطلق عليها إسم "فئة تكافؤ البادئة" (prefix equivalence class)، كمثال توضيحي على ذلك: فئة تكافؤ البادئة A هي المجموعة P_A والتي تضم كل العناصرالتي تشترك في A كعنصر بادئ لها، يرمز لهذه الفئة بـ:

$$P_A = \{AB, AC, AD, AE\}$$

ا) آلية عمل خوارزمية Eclat

تستخدم خوارزمية Eclat قاعدة المعاملات العمودية (Vertical Transactional Base)، بالتالي فإن مدخلاتها تكون عبارة عن مجموعة الثنائيات $< i, t(i) >$ التي تضم العنصر من المجموعة الكلية للعناصر إلى جوار مجموعة معرِّفات المعامَلات التي تحتوي ذلك العنصر في القاعدة.

خوارزمية Eclat هي خوارزمية تراجعية (Recursive Algorithm)، تقوم بعملية البحث نحو العمق بداية مع مجموعة العناصر المتكررة ذات العنصر الواحد (1-itemset) والتي تمثل فئة تكافؤ البادئة الخالية $\emptyset$، ثم تقوم بالإحتفاظ بنتائج تقاطع معرِّفات معامَلات مجموعات العناصر من نفس فئة التكافؤ، وعند كل إستدعاء تراجعي (recursive call) تقوم الخوارزمية بالتحقق من كل ثنائية (مجموعة العناصر – مجموعة معرِّفات المعامَلات) (itemset-tidset) فإذا كانت متكررة، تقوم الخوارزمية بعملية تقاطع مجموعة معرِّفات المعامَلات مع باقي العناصر من نفس فئة التكافؤ من أجل إنشاء مجموعات جديدة من العناصر المتكررة المرشحة، وتستمر العملية إلى أن تنتهي إمتدادات مجموعات العناصر في كل فئات التكافؤ، ويتم الحصول على مجموعات العناصر المتكررة النهائية.

ب) مثال يوضح آلية عمل خوارزمية Eclat

من أجل إستخدام خوارزمية Eclat تم تحويل قاعدة المعاملات في الشكل 2.2.أ إلى قاعدة المعاملات العمودية كما هو يظهره الشكل (2.2.ج)، وبفرض العتبة الدنيا للداعم

تساوي 3 ($\mu = 3$) فإن جميع العناصر ذات العنصر الواحد الخاصة بفئة تكافؤ البادئة الخالية P_ϕ هي عناصر متكررة ، فتكون P_ϕ على النحو التالي:

$$P_\phi = \{< A, 1345 >, < B, 123456 >, < C, 2456 >, \\ < D, 1356 >, < E, 12345 >\}$$

تقوم خوارزمية Eclat بتنفيذ عملية تقاطع مجموعة معرِّفات المعامَلات $t(A)$ الموافقة للعنصر A مع باقي معرِّفات المعامَلات $t(B)$، $t(C)$، $t(D)$ و$t(E)$ بهدف الحصول على معرِّفات المعامَلات الخاصة بمجموعة العناصر AB، AC، AD وAE على الترتيب، ومن ثم حساب الداعم الخاص بكل مجموعة لإقصاء مجموعات العناصر غير المتكررة والإبقاء فقط على مجموعات العناصر المتكررة التي تشترك في البادئة A في فئة التكافؤ P_A، كما يأتي:

- إيجاد معرِّفات المعامَلات مجموعات العناصر:

$$t(AB) = t(A) \cap t(B) = 1345 \cap 123456 = 1345$$
$$t(AC) = t(A) \cap t(C) = 1345 \cap 2456 = 45$$
$$t(AD) = t(A) \cap t(D) = 1345 \cap 1356 = 134$$
$$t(AE) = t(A) \cap t(E) = 1345 \cap 12345 = 1345$$

- حساب الداعم الخاص بها:

$$sup(AB) = |1345| = 4$$
$$sup(AC) = |45| = 2$$
$$sup(AD) = |134| = 3$$
$$sup(AE) = |1345| = 4$$

- مقارنة الداعم الخاص بكل مجموعة عناصر مع μ العتبة الدنيا للداعم:

$sup(AB) = 4 > \mu \longleftarrow$ هذا يعني أن AB **عنصر متكرر** (يتم الإحتفاظ به في فئة التكافؤ P_A)

$sup(AC) = 2 < \mu \longleftarrow$ هذا يعني أن AC **ليس عنصرا متكررا** (يتم إقصاؤه من فئة التكافؤ P_A)

$sup(AD) = 3 \geq \mu \longleftarrow$ هذا يعني أن AD **عنصر متكرر** (يتم الإحتفاظ به في فئة التكافؤ P_A)

$sup(AE) = 4 > \mu \longleftarrow$ هذا يعني أن AE **عنصر متكرر** (يتم الإحتفاظ به في فئة التكافؤ P_A)

لتصبح بذلك مجموعة العناصر المكونة لفئة تكافؤ البادئة A من الشكل:

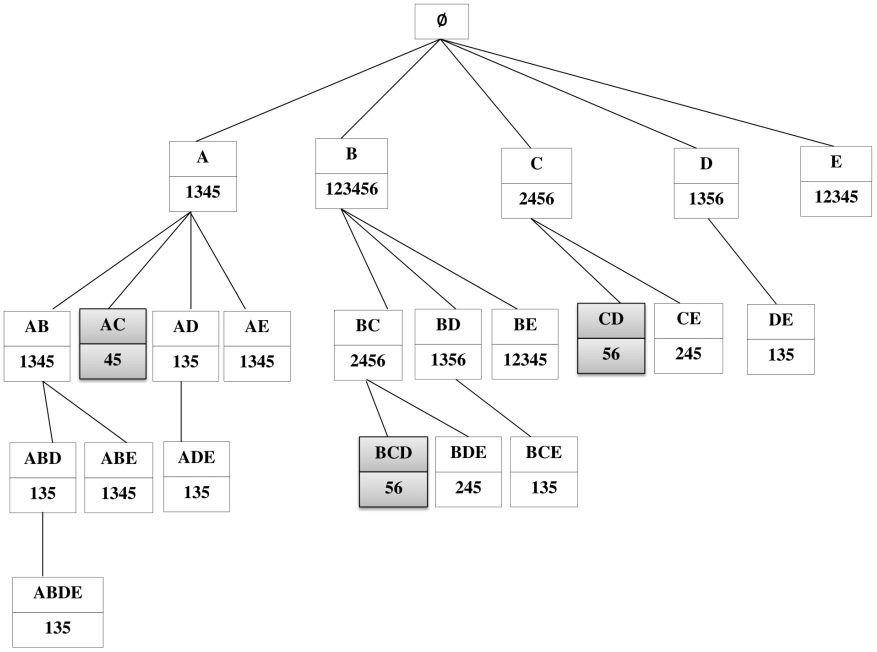
$$P_A = \{< AB, 1345 >, < AD, 135 >, < AE, 1345 >\}$$

يتم إنشاء P_B فئة تكافؤ البادئة B المكونة من مجموعة العناصر المتكررة ذات البادئة B والتي هي نتاج عملية تقاطع $t(B)$ معرِّفات معامَلات العنصر B مع باقي معرِّفات معامَلات $t(C)$، $t(D)$ و $t(E)$ كما يلي:

$$P_B = \{< BC, 2456 >, < BD, 1356 >, < BE, 12345 >\}$$

وبطريقة تراجعية تتم معالجة بقية الفروع وتنفيذ عمليات تقاطع معرِّفات المعامَلات بين العناصر المرشحة إلى غاية إستخراج مجموعة العناصر المتكررة النهائية.

الشكل 3.2 يوضح العملية الكاملة لخوارزمية Eclat على مجموعة عناصر قاعدة المعاملات العمودية المبينة سابقا من خلال الجدول 2.2.ج، حيث تمثل الخانات الرمادية مجموعات العناصر غير المتكررة.



شكل 3.2: خوارزمية Eclat (تقاطع مجموعة معرِّفات المعامَلات) [20].

ج) خوارزمية Eclat

الخوارزمية 3 أدناه توضح آلية عمل خوارزمية Eclat من خلال سلسلة الأوامر البرمجية [20].

خوارزمية 3 سلسلة الأوامر البرمجية لخوارزمية Eclat.

خوارزمية	Eclat(P, μ, F)
المدخلات	فئات التكافؤ P، العتبة الدنيا للداعم μ، مجموعة العناصر المتكررة F
المخرجات	مجموعة العناصر المتكررة F
المبدأ	$\mathbb{F} \longleftarrow \emptyset$, $P \leftarrow \{\langle \mathrm{i}, t(i)\rangle \mid \mathrm{i} \in I, \mid t(i) \mid \geq \mu\}$
Eclat(P, μ, F)	
foreach $\langle X_a, t(X_a)\rangle \in P$ **do**	1
$F \leftarrow F \cup (X_a, sup(X_a))$	2
$P_a \leftarrow \emptyset$	3
foreach $\langle X_b, t(X_b)\rangle \in P$ with $X_b > X_a$ **do**	4
$X_{ab} = X_a \cup X_b$	5
$t(X_{ab}) = t(X_a) \cap t(X_b)$	6
if $sup(X_{ab} \geq \mu)$ **then**	7
$P_a \leftarrow P_a\{\langle X_{ab}, t(X_{ab})\rangle\}$	8
if $P_a \neq \emptyset$ **then** Eclat(P_a, μ, F)	9

د) التعقيد الحسابي للخوارزمية algorithm complexity

في أسوء الحالات التعقيد الحسابي لخوارزمية Eclat هو $O(|D|2^{|I|})$ لإحتمالية أن يكون هناك $2^{|I|}$ عنصرا متكررا، إضافة الى أن تقاطع مجموعتي معرِّفات معامَلات يأخذ من الوقت $O(|D|)$ على الأكثر.

قامت خوارزمية Eclat بتسريع عملية حساب الداعم بفضل إعتمادها على مبدأ تقاطع مجموعات معرِّفات المعامَلات من خلال قاعدة المعاملات العمودية والتي تقوم بعملية المسح خلالها مرة واحدة فقط، لكن الإحتفاظ بالنتائج الوسطية لتقاطع مجموعات معرِّفات المعامَلات (Intermediate tidsets) قد يصبح أكبر من حجم الذاكرة في حالة قواعد المعاملات كثيرة العمليات.

ظهر تحسين لهذه لخوارزمية بظهور خوارزمية D-Eclat (Diffsets Eclat) [21] والتي ساهمت في تقليص حجم مجموعة المعرفات الوسطية من خلال الإحتفاظ بالفروقات بين مجموعات معرِّفات المعامَلات (Difference of tidsets) بدل الإحتفاظ بها كلية.

4.8.2 المقاربة بالإسقاط Projection Approach

طريقة المقاربة بالإسقاط تم تقديمها سنة 2000 من قبل Han, Pei, Yin من خلال خوارزمية FPGrowth [9] والتي تقوم على مبدأ نمو النمط المتكرر (Frequent Pattern Growth) بهدف تجنب العمليات المكلفة لإنشاء مجموعة العناصر المرشحة و تفادي المسح المتكرر لقاعدة المعاملات من أجل حساب الداعم الخاص بكل مجموعة عناصر. يتم ذلك بفضل الهيكلة الجديدة التي تعطيها الخوارزمية لقاعدة المعاملات والتي تسمى بـ: شجرة النمط المتكرر (Frequent Pattern tree (FP-tree)) بحيث يتم إستخراج مجموعة العناصر المتكررة منها بشكل مباشر.

بإعتبار شجرة النمط المتكرر (FP-tree) بنية متراصة، فإنها تتكون من:

- الشجرة (Tree): حيث تعرف الشجرة بالجذر الخاص بها (root) والذي يمثل العنصر الخالي $\emptyset$، إضافة إلى العقد (nodes) التي تمثل جميع العناصر المتواجدة في قاعدة المعاملات، تضم كل عقدة عنصرا مختلفا من قاعدة المعاملات كما تقوم كل عقدة أيضا بتخزين المعلومات حول الداعم لمجموعة العناصر التي تطوق المسار من الجذر وصولا إلى تلك العقدة.
- لائحة العناصر المتكررة (Frequent Items Header Table): هو جدول يضم قائمة بالعناصر المتكررة في قاعدة المعاملات، حيث يرتبط كل عنصر فيه بمؤشر (pointer) يدل على أول عقدة في الشجرة تضم ذلك العنصر.

ا) عملية إنشاء شجرة النمط المتكرر FP-tree

يتم إنشاء شجرة النمط المتكرر (FP-tree) على النحو الآتي:
في البداية تتكون الشجرة من العقدة التي تحتوي العنصر الخالي $\emptyset$ فقط والذي يمثل الجذر(root)، ولكون شجرة النمط المتكرر بنية مدمجة لقاعدة المعاملات D وبغرض ضغطها قدر ما يمكن، تكون العناصر الأكثر تكرار في أعلى الشجرة أقرب إلى الجذر، ثم يتم ترتيب باقي العناصر بشكل تنازلي حسب قيمة الداعم الخاص بكل عنصر في قاعدة المعاملات الأولية مع إهمال العناصر غير المتكررة.

تحتفظ كل عقدة (Node) من الشجرة بالعنصر الذي تمثله إلى جانب عداد (counter) لحساب عدد مرات المرور بتلك العقدة.
يتم إدراج كل معاملة t من قاعدة المعاملات D تحتوي مجموعة العناصر X حيث $< t, X > \in D$ ضمن هيكلة شجرة النمط المتكرر (FP-Tree) بعد القيام بعملية ترتيب تنازلي (Descendant order) للعناصر المكونة للمجموعة X، بحيث تزيد قيمة العداد الخاص بكل عقدة من الشجرة تحمل نفس العنصر والمعاملة X في المسار المتبع، أما

إذا وجدت عناصر جديدة فإنه يتم إنشاء عقد جديدة لها مع بدء العداد فيها بالقيمة 1.

يكتمل بناء شجرة النمط المتكرر عند إدراج جميع المعاملات من القاعدة D.

- **مثال 1 إنشاء شجرة النمط المتكرر FP-tree**

بفرض العتبة الدنيا للداعم مساوية للقيمة 3 أي أن $(\mu = 3)$، الجدول 3.2 يمثل قاعدة المعاملات الأولية D.

Tid	$itemset$
1	$ABDE$
2	BCE
3	$ABDE$
4	$ABCE$
5	$ABCDE$
6	BCD

جدول 3.2: قاعدة المعاملات الأولية D.

يتم حساب الداعم لكل عنصر من العناصر الموجودة في قاعدة المعاملات D، حيث يتم الإحتفاظ بالعناصر التي تكون قيمة داعمها أكبر من العتبة الدنيا للداعم μ بإعتبارها عناصر متكررة، وإهمال العناصر التي يكون الداعم الخاص بها أقل من العتبة الدنيا للداعم.

يمثل الجدول 4.2 مجموعة العناصر المتكررة المستخرجة من قاعدة المعاملات 3.2 أعلاه، أما الجدول 5.2 فيمثل مجموعة العناصر المتكررة بعد الترتيب التنازلي لقيم الداعم الخاص بكل عنصر

$item$	$support$
A	4
B	6
C	4
D	4
E	5

جدول 4.2: مجموعة العناصر المتكررة.

$\longleftarrow$

$item$	$support$
B	6
E	5
A	4
C	4
D	4

جدول 5.2: مجموعة العناصر المتكررة بعد الترتيب التنازلي.

بعد ذلك يتم إعادة ترتيب مجموعات العناصر المتكررة فقط حسب الترتيب التنازلي لقيمة الداعم الخاص بكل عنصر متكرر، وبالإسقاط على مجموعة العناصر في قاعدة المعاملات فإن ترتيبها يكون كالتالي (من اليسار الى اليمين)
$\{B:6,\ E:5,\ A:4,\ C:4,\ D:4\}$، فتصبح قاعدة المعاملات كما يظهره الجدول 6.2.

Tid	$itemset$	Ordered frequent itemsets
1	$ABDE$	$BEAD$
2	BCE	BEC
3	$ABDE$	$BEAD$
4	$ABCE$	$BEAC$
5	$ABCDE$	$BEACD$
6	BCD	BCD

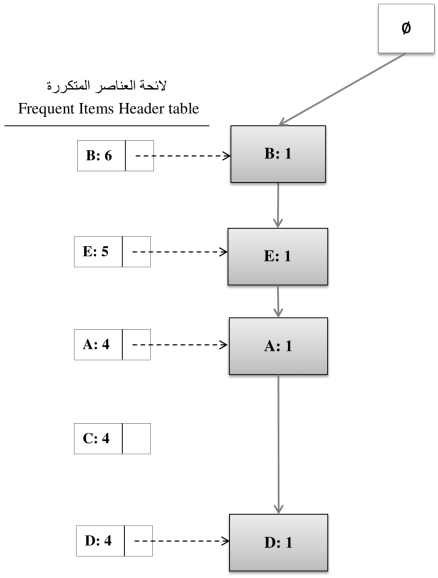
جدول 6.2: قاعدة المعاملات D بعد عملية الترتيب.

بإستغلال مجموعة العناصر المتكررة المرتبة (Ordered frequent itemsets) كما يبينه الجدول 7.2.

Tid	Ordered frequent itemsets
1	$BEAD$
2	BEC
3	$BEAD$
4	$BEAC$
5	$BEACD$
6	BCD

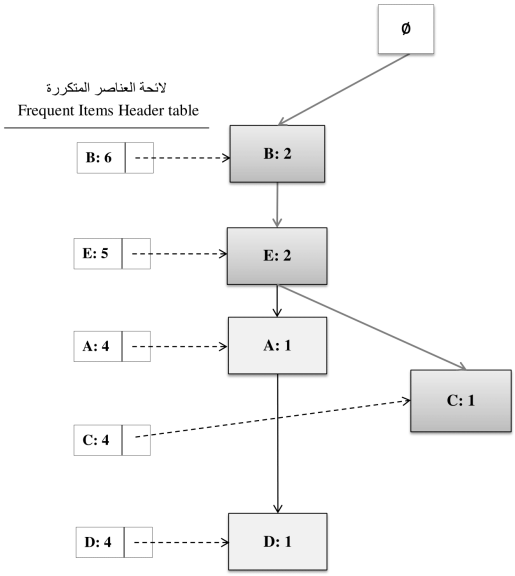
جدول 7.2: قاعدة المعاملات لإنشاء شجرة النمط المتكرر.

يتم إنشاء شجرة النمط المتكرر عبر المراحل التالية:
أولا يتم إنشاء الجذر (root) الذي يضم العنصر الخالي $\emptyset$، ثم تدرج المعاملة الأولى t_1 التي تضم العناصر $BEAD$ حيث تتشكل عقدة (Node) تحمل العنصر إضافة إلى العداد الذي يحفظ عدد مرات المرور بها، كما يظهره الشكل 4.2.



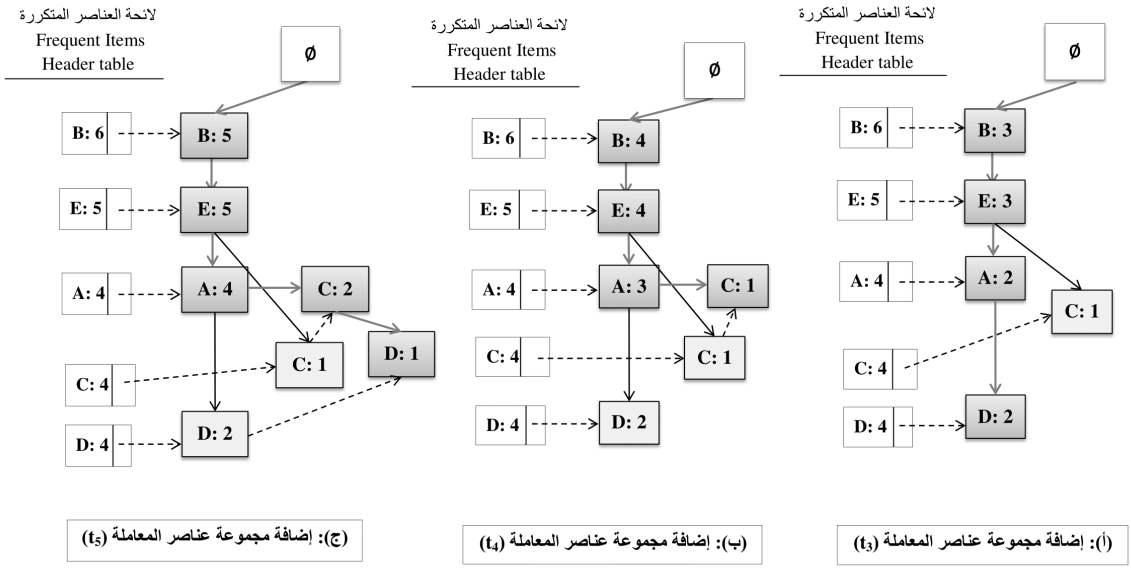
شكل 4.2: إضافة عناصر المعاملة (t_1) إلى شجرة النمط المتكرر FP-tree.

تدرج عناصر المعاملة t_2 التي تضم كلا من BEC بزيادة القيمة 1 في عداد (counter) كل عقدة تحمل نفس العنصر (same item) في المسار المتبع إنطلاقا من الجذر، وإنشاء عقدة جديدة عند إيجاد عنصر مختلف (different item) كما يبينه الشكل 5.2.



شكل 5.2: إضافة عناصر المعاملة (t_2) إلى شجرة النمط المتكرر FP-tree.

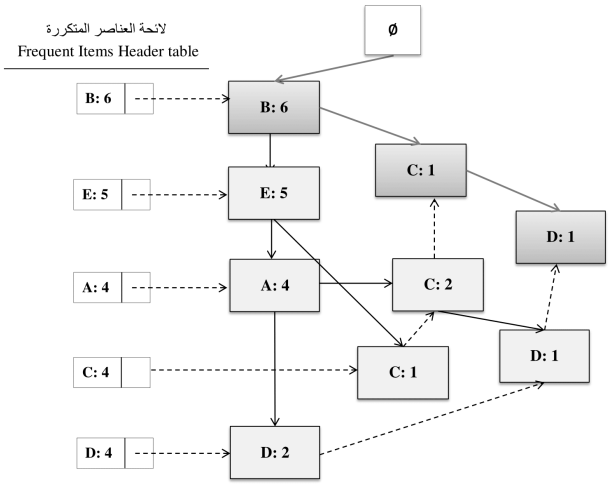
وبطريقة تكرارية يتم إدراج باقي المعاملات t_3، t_4، t_5 في شجرة النمط المتكرر (FP-tree)، الشكل 6.2 يبين العملية.



شكل 6.2: إضافة مجموعة عناصر المعاملات $(t_3\ t_4\ t_5)$ لشجرة النمط المتكرر FP-tree.

تنتهي عملية إنشاء شجرة النمط المتكرر عند إدراج آخر معاملة موجودة في القاعدة وهي المعاملة t_6، كما وتكتمل لائحة العناصر المتكررة بوضع المؤشرات (pointers) إنطلاقا من كل عنصر في اللائحة وصولا إلى جميع العقد (nodes) في الشجرة التي تحوي نفس العنصر.

الشكل 7.2 يبين البنية النهائية لشجرة النمط المتكرر بإضافة مجموعة عناصر المعاملة (t_6).



شكل 7.2: الهيكل النهائي لشجرة النمط المتكرر FP-tree.

وللتثبت من صحة الشجرة المنشأة (FP-tree validation) تتم مقارنة المعلومات المتحصل عليها من مختلف العقد في الشجرة (different tree nodes) وتعني العنصر الممثل من طرف العقدة وقيمة العداد فيها، مع المعلومات المتواجدة في لائحة العناصر المتكررة، وتعني قيمة الداعم لكل عنصر متكرر في اللائحة (the support value of each frequent item on the header table).

بالإسقاط على المثال 1 أعلاه، وبعد القيام بعملية تعداد لجميع عناصر هيكلة الشجرة (العقد) في الشكل 7.2، تم الحصول على النتيجة التالية:
$\{B : 6,\ E : 5,\ A : 4,\ C : 4,\ D : 4\}$، هذه النتيجة مطابقة للمعلومات الموجودة في لائحة العناصر المتكررة، وهذا ما يؤكد صحة هيكلة شجرة النمط المتكرر لقاعدة المعاملات.

ب) تحديد الأنماط المتكررة من خلال خوارزمية FPGrowth

في المقاربة بالإسقاط تتم عملية تحديد مجموعات العناصر المتكررة الموجودة في قاعدة المعاملات (transactional base) من خلال شجرة النمط المتكرر المنشأة (FP-tree) إضافة إلى لائحة العناصر المتكررة.
خلال عملية إنشاء شجرة النمط المتكرر، تنشأ بالموازاة روابط (links) ما بين العناصر

الموجودة في اللائحة والعقد الممثلة لتلك العناصر، بهدف تسهيل عمليات التنقل فيما بينها.

بالتالي فإن العقد الممثلة لأي عنصر يمكن الوصول إليها بإتباع سلسلة المؤشرات (pointers) إنطلاقا من الخانة الموافقة لذلك العنصر في لائحة العناصر المتكررة.
أي أن كل الأنماط التي تحتوي عنصرا ما ستكوّن مجموعات جزئية (subsets) عبر مسارات في هيكلة الشجرة تبدأ من الجذر (root) وتنتهي عند كل عقدة تمثل ذلك العنصر [12].

فبمجرد إتمام هيكلة شجرة النمط المتكرر (FP-tree Structure) فإنها تحل محل قاعدة المعاملات عند عملية إستخراج العناصر المتكررة بإستخدام خوارزمية (FPGrowth) التراجعية (recursive) التي تظهر سلسلة الأوامر الخاصة بها في الخوارزمية 4 أدناه، حيث تقوم بتحديد العناصر المتكررة عبر إسقاط كل عنصر أو مجموعة عناصر على هيكلة الشجرة بطريقة تراجعية مرورا بكل جزء من الشجرة (FP-tree) يحتوي عقدة تمثل ذلك العنصر، وتتوقف الخوارزمية عندما يصبح الجزء من شجرة النمط المتكرر الذي يُبحث فيه عن العنصر عبارة عن مسار وحيد (single path) وهو ما يمثل الحالة البسيطة أو الأساسية في التراجع.

بعدها يتم تعداد كل مجموعات العناصر والتي هي مجموعات جزئية للمسارات المستخرجة، ويحدد الداعم (support) الخاص بها من خلال أصغر قيمة داعم للعنصر المتكرر في كل مجموعة جزئية.

- **مثال 2 إستخراج الأنماط المتكررة**

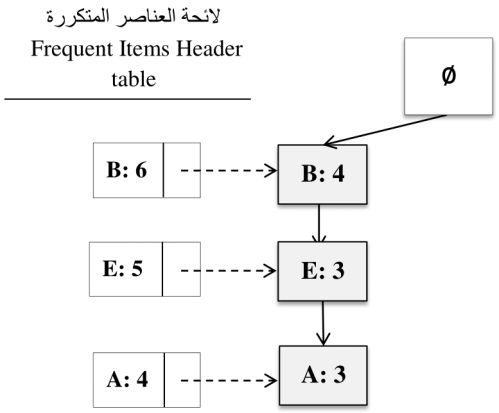
بالعودة إلى المثال 1 السابق ومن خلال شجرة النمط المتكرر (FP-tree) المنشأة، كما يوضحه الشكل 7.2: لا تحتاج عملية إستخراج الأنماط المتكررة إلى تعداد شامل (Exhaustive) للعناصر، بل تبدأ العملية وفقا للعناصر المحددة مسبقا في لائحة العناصر المتكررة بشكل تصاعدي من العنصر المتكرر ذي القيمة الأقل للداعم وصولا إلى العنصر المتكرر الذي يحمل أكبر قيمة للداعم، بإستعمال مبدأ "فرَق تسد" (divide and conquer).

أول عنصر تبتدأ به العملية من لائحة العناصر المتكررة هو العنصر D أين يتم البحث عن جميع المسارات التي يمكن من خلالها إنطلاقا من الجذر (root) الوصول إلى العقد التي تمثل ذلك العنصر، مع أخذ القيمة الأصغر للعداد (counter) في ذلك المسار:

$$D :< \{B, E, A : 2\} >, < \{B, E, A, C : 1\} >, < \{B, C : 1\} >$$

وهو ما يسمى بقاعدة الأنماط الشرطية للعنصر D
(Conditional Pattern Base "CPB" for item D).

بالجمع ما بين العناصر المشتركة وبعد حذف العناصر غير المتكررة والإبقاء فقط على العناصر المشتركة المتكررة، يتم إنشاء شجرة الأنماط المتكررة الشرطية للعنصر D (Conditional FP-tree for item D) من خلال العناصر المتكررة $D : \{B, E, A : 3\}, \{B : 4\}$ كما يظهره الشكل 8.2.



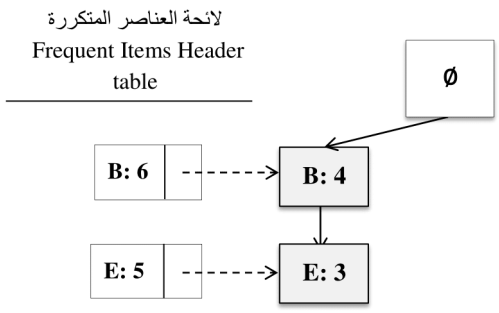
شكل 8.2: شجرة الأنماط المتكررة الشرطية للعنصر D.

ثم تأتي عملية توليف الأنماط المتكررة (Frequent Patterns combination) بإضافة العنصر D في كل توليفة من أجل تشكيل الأنماط المتكررة الشرطية (Conditional Frequent Patterns) مع الأخذ بالإعتبار عدم إحتساب العنصر الخالي لتصبح مجموعات العناصر المتكررة كالتالي:

$$\{B, E, A, D : 3\}, \{E, A, D : 3\}, \{B, A, D : 3\}, \{B, E, D : 3\},$$
$$\{A, D : 3\}, \{E, D : 3\}, \{B, D : 4\}$$

وبالإنتقال إلى العنصر الموالي، فإن قاعدة الأنماط الشرطية للعنصر C ("CPB" for item C) تكون كالتالي:

$C :< \{B, E, A : 2\} >, < \{B, E : 1\} >, < \{B : 1\} >$ و يتم إنشاء شجرة الأنماط المتكررة الشرطية للعنصر C (Conditional FP-tree for item C) من خلال العناصر المتكررة $C : \{B : 4\}, \{E : 3\}$ كما يظهره الشكل 9.2.



شكل 9.2: شجرة الأنماط المتكررة الشرطية للعنصر C.

لتصبح مجموعات العناصر المتكررة كما يأتي:
$\{B, E, C : 3\}$ $\{E, C : 3\}$ $\{B, C : 4\}$

يتم تلخيص مخرجات خوارزمية النمط المتكرر (FPGrowth) في الجدول 8.2 التالي:

الأنماط المتكررة المنشأة (F.P generated)	شجرة الأنماط المتكررة الشرطية (Conditional FP-tree)	قاعدة الأنماط الشرطية (Conditional Pattern Base)	العنصر item
$\{B, D : 4\}, \{E, D : 3\}$ $\{A, D : 3\}, \{B, E, D : 3\}$ $\{B, A, D : 3\}, \{E, A, D : 3\}$ $\{B, E, A, D : 3\}$	$\{B, E, A : 3\}$ $\{B : 4\}$	$< \{B, E, A : 2\},$ $\{B, E, A, C : 1\} >$ $< \{B, C : 1\} >$	D
$\{B, C : 4\}, \{E, C : 3\}$ $\{B, E, C : 3\}$	$\{B, : 4\}, \{E : 3\}$	$< \{B, E, A : 2\}, \{B, E : 1\} >$ $< \{B : 1\} >$	C
$\{B, A : 4\}, \{E, A : 4\}$ $\{B, E, A : 4\}$	$\{B : 4\}, \{E : 4\}$	$\{B, E : 4\}$	A
$\{B, E : 5\}$	$\{B : 5\}$	$\{B : 5\}$	E

جدول 8.2: مخرجات خوارزمية النمط المتكرر (FPGrowth).

ج) خوارزمية نمو النمط المتكرر FPGrowth

توضح الخوارزمية 4 أدناه آلية عمل الخوارزمية FPGrowth من خلال سلسلة الأوامر البرمجية [20].

خوارزمية 4 سلسلة الأوامر البرمجية لخوارزمية نمو النمط المتكرر (FPGrowth).

خوارزمية	FPGrowth(R,P,F, μ)
المدخلات	R هيكلة شجرة الأنماط المتكررة، P فئات التكافؤ، μ العتبة الدنيا للداعم
المخرجات	F مجموعة العناصر المتكررة
المبدأ	R⟵ FP-tree(D), P⟵ ∅, F ⟵ ∅
	FPGrowth(R,P,F,μ)
1	حذف جميع العناصر غير المتكررة من R
2	**if** IsPath(R) **then**
	// إضافة مجموعة جزيئة من R في F
3	**foreach** $Y \subseteq R$ do
5	$X \leftarrow P \cup Y$
6	$sup(X) \leftarrow min_{x \in Y}\{cnt(x)\}$
7	$F \leftarrow F \cup \{(X, sup(X))\}$
8	**else**
	// i معالجة شجرة الأنماط المتكررة الشرطية من أجل كل عنصرمتكرر
9	**foreach** $i \in R$ in increasing order of $sup(i)$ **do**
10	$X \leftarrow P \cup \{i\}$
11	$sup(X) \leftarrow sup(i)$
	// i مجموع العدادات ($cnt(i)$) لكل العقد الممثلة لـ
12	$F \leftarrow F \cup \{(X, sup(X))\}$
13	$R_X \leftarrow \emptyset$
	// X إسقاط شجرة الأنماط المتكررة من أجل المجموعة
14	**foreach** $path(i)$ **do**
15	$cnt(i) \leftarrow$ count of i in $path$
16	Insert $path$, excluding i into FP-tree R_X with count $cnt(i)$
17	**if** $R_X \neq \emptyset$ **then** FP-Growth(R_X, X, F, μ)

د) التعقيد الحسابي للخوارزمية algorithm complexity

في أسوء الحالات، التعقيد الحسابي لخوارزمية نمو النمط المتكرر (FPGrowth) هو: $O(|D|2^{|I|})$، بإعتبار جميع العناصر متكررة. وعلى الرغم من الهيكلة المتراصة والمتكاملة لشجرة النمط المتكرر (FP-tree compact structure)، فإنه يمكن أحيانا أن يتجاوز حجمها حجم قاعدة المعاملات الأصلية، فيصبح من الصعب إحتواؤها كاملة

في الذاكرة المركزية RAM بغرض إستعمالها، كما أن عملية بنائها يمكن أن تستغرق وقتا مطولا وتتطلب إستهلاكا كبيرا للموارد (resources consumption).

9.2 إنشاء قواعد الارتباط Association Rules Generation

تتم عملية إنشاء قواعد الإرتباط (Association Rules) إنطلاقا من مجموعة العناصر المتكررة الكلية F.

لتكن Z مجموعة عناصر من المجموعة F، يُعبَر عن قاعدة الإرتباط بالشكل التالي: $X \to Y$، حيث X هي مجموعة جزئية من مجموعة العناصر Z (وتمثل الطرف الأيسر من القاعدة والذي يسمى بالمقدمة (premise)) بحيث: $(X \subset Z) \in F$، ويمثل Y الطرفُ الأيمن من القاعدة والذي يسمى بالنتيجة (consequence) المجموعة الناتجة عن الفرق بين المجموعتين Z و X بحيث: $(Y = Z \backslash X) \in F$ ، يُعبَر عن المجموعة Y أيضا بـ: مكمَلة المجموعة X بالنسبة لمجموعة العناصر Z،
(Y is the complement of the subset X in the set Z).

بالتالي، القاعدة $X \to Y$ بالتأكيد قاعدة متكررة (frequent rule) لأن طرفيها عبارة عن مجموعات عناصر متكررة (frequent itemsets)، يعني أن:

$$s = sup(X \to Y) = |t(XY)| = sup(XY) = sup(Z) \geq \mu$$

يتبقى فقط التحقق من أن معامل الثقة c الخاص بالقاعدة (the rule confidence) يستوفي العتبة الدنيا لمعامل الثقة γ المحددة من قبل المستخدم، عبر حساب قيمته كالتالي:

$$c = conf(X \to Y) = \frac{sup(X \cup Y)}{sup(X)} = \frac{sup(XY)}{sup(X)} = \frac{sup(Z)}{sup(X)}$$

فإذا كان معامل الثقة أكبر من أو يساوي العتبة الدنيا لمعامل الثقة $(c \geq \gamma)$، فإن قاعدة الإرتباط تكون قاعدة قوية (strong rule).

بالمقابل، إذا كان معامل ثقة القاعدة $X \to Y$ أقل من العتبة الدنيا المحددة لمعامل الثقة γ أي: $c\ (X \to Y) < \gamma$ فإنه مما لاشك فيه أن كل قواعد الإرتباط من الشكل $(W \to Z \backslash W)$، حيث تعبر W عن المجموعات الجزئية (subsets) للمجموعة X $(W \subset X)$، تحمل المجموعات W قيما منخفضة لمعامل الثقة أي أقل من العتبة الدنيا γ، يكون هذا الأمر مفيدا في تفادي التحقق من معامل ثقة المجموعات الجزئية للمجموعة X [20].

ا) خوارزمية إنشاء قواعد الإرتباط Association Rules Generation Algorithm

توضح الخوارزمية 5 سلسلة الأوامر البرمجية المتبعة للتنقيب عن قواعد الإرتباط [20]

خوارزمية 5 سلسلة الأوامر البرمجية لخوارزمية قواعد الارتباط (AssociationRules).

خوارزمية	AssociationRules(F, γ)
المدخلات	مجموعة العناصر المتكررة F، العتبة الدنيا لمعامل الثقة γ // F, γ
المخرجات	عرض قواعد الإرتباط القوية (display strong Association Rules)
	AssociationRules(F, γ)
1	**foreach** $Z \in F, such\ that\ \|Z\| \geq 2$ **do**
2	$A \leftarrow \{X \| X \subset Z, X \neq \emptyset\}$
3	**while** $A \neq \emptyset$ **do**
4	$X \leftarrow$ maximal element in A
5	$A \leftarrow A \backslash X$ // تحديث مجموعة السوابق A بحذف المجموعة X منها
6	$c \leftarrow sup(Z)/sup(X)$
7	**if** $c \geq \gamma$ **then**
8	print $X \rightarrow Y, sup(Z), c$ // عرض قاعدة الإرتباط
9	**else**
10	$A \leftarrow A \backslash \{W \| W \subset X\}$ // حذف كل المجموعات الجزئية لـX من مجموعة السوابق A

فمن أجل كل مجموعة عناصر متكررة $Z \in F$ والتي تحتوي على عنصرين فأكثر، تبدأ عملية تهيئة مجموعة السوابق المحتملة A (set of possible antecedents) بإضافة كل المجموعات الجزئية المحتملة للمجموعة Z ماعدا المجموعة الخالية والمجموعة الكلية Z نفسها، ومن أجل كل مجموعة X من مجموعة السوابق المحتملة A يتم التحقق مما إذا كان معامل الثقة الخاص بالقاعدة $X \rightarrow Z \backslash X$ مساويًا أو أكبر من العتبة الدنيا لمعامل الثقة γ، فإذا كان الأمر كذلك تتم طباعة تلك القاعدة كمخرجات للخوارزمية، وإذا لم يتحقق ذلك يتم حذف جميع المجموعات الجزئية W المحتواة في المجموعة X $(W \subset X)$ من مجموعة السوابق المحتملة.

ب) مثال

بالنظر إلى مجموعة العناصر المتكررة $Z = ABDE(3)$ والتي تعتبر أطول مجموعة عناصر متكررة وفقا لجميع الخوارزميات المدروسة سابقا
(Apriori / Eclat / FPGrowth) والتي تحمل داعما مساويا للقيمة 3 كما يظهر داخل الأقواس.
بإعتبار العتبة الدنيا لمعامل الثقة γ مساوية للقيمة 0.9 أي أن $(\gamma = 0.9)$، ومن أجل

إنشاء قواعد إرتباط قوية (strong association rules)، تتم أولا عملية تهيئة مجموعة السوابق المحتملة A (set of possible antecedents) بمجموعات العناصر التالية:

$$\mathsf{A} = \{ABD(3), ABE(4), ADE(3), BDE(3), AB(3), AD(4), AE(4), BD(4), BE(5), DE(3), A(4), B(6), D(4), E(5)\}$$

تبدأ العملية بالمجموعة الأولى X من مجموعة السوابق المحتملة، بمعنى $X = ABD$، وبإسقاط علاقة قاعدة الإرتباط الموضحة سابقا $X \rightarrow Y$ حيث $Y = Z \backslash X$ على المجموعة الحالية، تصبح القاعدة من الشكل $ABD \rightarrow (ABDE \backslash ABD)$ أي $ABD \rightarrow E$.

يتم حساب معامل الثقة c لهذه القاعدة كما يأتي:

$$c = conf(ABD \rightarrow E) = \frac{sup(ABD \cup E)}{sup(ABD)} = \frac{sup(ABDE)}{sup(ABD)} = \frac{sup(3)}{sup(3)} = 1.0$$

بما أن $c(ABD \rightarrow E) \geq \gamma$ فإن هذه القاعدة قاعدة إرتباط قوية، يتم عرض هذه القاعدة ثم يتم الإنتقال إلى المجموعة الموالية $X = ABE$، وتكون قاعدة الإرتباط لهذه المجموعة كما يأتي $ABE \rightarrow D$.

تتم عملية حساب معامل الثقة c لهذه القاعدة كالتالي:

$$c = conf(ABE \rightarrow D) = \frac{sup(ABE \cup D)}{sup(ABE)} = \frac{sup(ABED)}{sup(ABE)} = \frac{sup(3)}{sup(4)} = 0.75$$

وبما أن $c(ABE \rightarrow D) < \gamma$ فإن القاعدة هذه ليست قاعدة إرتباط قوية، بالتالي يتم إزالة جميع المجموعات الجزئية لمجموعة العناصر ABE من مجموعة السوابق المحتملة A، لتصبح المجموعة المحدَثة لمجموعة السوابق (updated set) كالتالي:

$$\mathsf{A} = \{ADE(3), BDE(3), AD(4), BD(4), DE(3), D(4)\}$$

وبالمرور بالمجموعات المتبقية، يتم التوصل إلى أنها تكوَن قواعد إرتباط قوية، كون قيمة معامل الثقة الخاص بها أكبر من العتبة γ، عدا المجموعة $X = BD$ التي تكوَن قاعدة الإرتباط $BD \rightarrow AE$، فبحساب قيمة معامل الثقة يتم الحصول على:

$c\ (BD \rightarrow AE) = 0.75 < \gamma$، بالتالي تحذف جميع المجموعات الجزئية الخاصة بها.

وفي النهاية، المجموعة النهائية لقواعد الإرتباط القوية التي تعرض كمخرجات هي كالآتي:

$$ABD \rightarrow E, conf = 1.0$$
$$ADE \rightarrow B, conf = 1.0$$
$$BDE \rightarrow A, conf = 1.0$$
$$AD \rightarrow BE, conf = 1.0$$
$$DE \rightarrow AB, conf = 1.0$$

10.2 خاتمة

في هذا الفصل تمّ التطرق إلى مختلف المفاهيم المتعلّقة بإستخراج مجموعات العناصر المتكررة الكلية وطرق إكتشافها من داخل قواعد المعاملات ذات الأحجام الكبيرة بإستعمال المقاربات والخوارزميات الأكثر تداولا في مجال التنقيب في البيانات.

وكان من الملاحظ أن كل خوارزمية تملك بعض المزايا والعيوب، ومن العيوب المشتركة لها التكلفة العالية في أسوء الحالات (worst cases) للزمن المستغرق في إستخراج مجموعة العناصر المتكررة.

لكن هذا الأمر راجع إلى كون هذه الخوارزميات تقوم بالتعداد الكلي لمجموعة العناصر المتكررة بهدف إستخلاص قواعد الإرتباط المهمة لاحقا.

ولأجل تفادي ذلك، يقوم الباحثون منذ بداية الألفية الجديدة وإلى اليوم بإقتراح ووضع خوارزميات تعتبر أحيانا إمتدادا للخوارزميات المدروسة سابقا، لكنها تهدف إلى إستخلاص مجموعات عناصر ذات تمثيل متراصّ (compressed representation) من أجل إستخلاص قواعد الإرتباط المهمة بأقل تكلفة ممكنة سواءً في الوقت المستغرق أو في إستغلال المساحة التخزينية للذاكرة، وهذا ما سيتم الإشارة إليه في الفصل الموالي.

الفصل الثالث

تلخيص مجموعات العناصر المتكررة
Summarizing Frequent Itemsets

1.3 مقدمة

عملية إستخلاص قواعد الإرتباط إنطلاقا من مجموعات العناصر لازالت تشغلُ الباحثين في مجال التنقيب في البيانات، بِخاصة الجزء المتعلَق بالبحث عن مجموعات العناصر المتكررة داخل قواعد المعاملات، لكون الخوارزمياتِ الأولى في الميدان والتي ظهرت في بداية تسعينات القرن الماضي إهتمت بالتعداد الكلي لمجموعات العناصر المتكررة حتى يمكن إستخلاص قواعد الإرتباط منها.

لكن منذ بداية الألفية ظهر إتجاه جديد يبحث عن إمكانية إستخلاص تلك القواعد إنطلاقا من حيَزٍ أقل لمجموعات العناصر المتكررة، بعبارة أخرى إيجاد تمثيلات متراصَةٍ تُعتبرُ نواة الحيَز الكلي لمجموعات العناصر المتكررة، وتحمل بشكل ضمني (Implicitly) هذه المجموعات من العناصر المتكررة، بهدف تجنَبِ التكلفة العالية في الزمن المستغرق وفي مساحة الذاكرة لخوارزميات التعداد الكلي لمجموعات العناصر المتكررة.

نتيجة لذلك، ظهرت تمثيلات مجموعات العناصر المتكررة المغلقة (Closed Frequent Itemsets)، ومجموعات العناصر المتكررة القصوى (Maximal Frequent Itemsets) من خلال جُملةٍ من الخوارزميات التجريبية (Heuristics)، والتي أثبتت فعاليتها في العديد من وِرش العمل الخاصة بالتنقيب في البيانات(Data Mining Workshops).

سيتِم في هذا الفصل إلقاء نظرة حول المفاهيم الأساسية لهذه التمثيلات مرورا ببعض خوارزميات إستخراج مجموعات العناصر المتكررة القصوى (Maximal Frequent Itemset)، ليتِم في الأخير عقدُ مقارنةٍ بين خوارزميتي إستخراج العناصر المتكررة القصوى الموجودتين في منصة (SPMF) مفتوحة المصدر بإستعمال عِدَة أنواع من قواعد المعاملات.

2.3 التمثيلات المتراصّة Compressed Representations

تكوِّن مجموعات العناصر المتكررة عادة حيزا كبيرا، ويزداد بشكل أسي (Exponentially) بتزايد عدد العناصر (Items)، وبالأخص قد يؤدي تحديد قيمة صغيرة للعتبة الدنيا للداعم (Low minimum support value) إلى ظهور عدد لا يمكن حصره من مجموعات العناصر المتكررة (Intractable number of Frequent Itemsets).

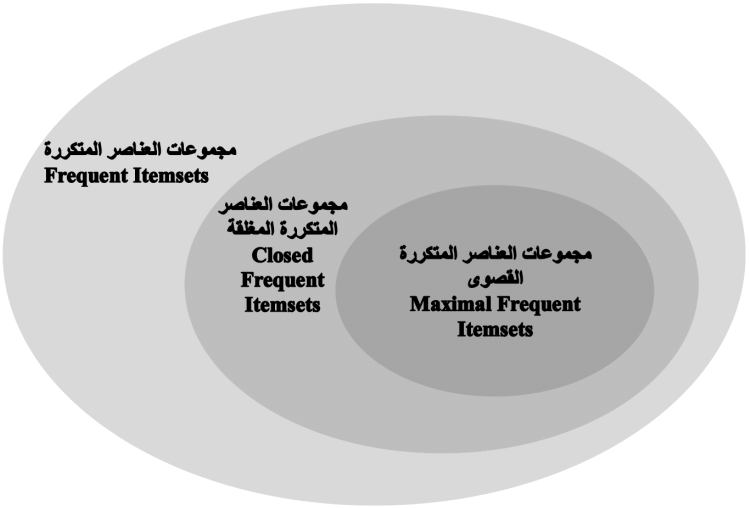
سيتم في هذا الفصل دراسة طريقة بديلة (Alternative approach) والتي تعنى بالتمثيلات المتراصّة (Compressed representations) لمجموعات العناصر المتكررة التي تلخص خصائصها الأساسية (Essential characteristics).

إستعمال التمثيلات المتراصّة لمجموعات العناصر المتكررة مفيد في تقليص متطلبات عمليات الحوسبة والتخزين إضافة إلى تسهيل عملية تحليل مجموعات العناصر المتكررة المتحصل عليها.

من بين هذه التمثيلات يوجد التمثيل الخاص بمجموعة العناصر المتكررة المغلقة (Closed Frequent Itemset(CFI))، إلى جانب التمثيل الخاص بمجموعة العناصر المتكررة القصوى (Maximal Frequent Itemset(MFI))، والتي ستتركز عليها الدراسة لاحقا [20].

فمن المهم إذن معرفة العلاقة بين كلٍ من مجموعات العناصر المتكررة الكلية (Frequent Itemsets(FI)) ومجموعات العناصر المتكررة المغلقة (CFI) ومجموعات العناصر المتكررة القصوى (MFI).

تعتبر كلّ من التمثيلات الخاصّة بمجموعات العناصر المتكررة المغلقة والقصوى مجموعات جزئية لمجموعة العناصر المتكررة الكلية، لكن مجموعات العناصر المتكررة القصوى تملك تمثيلا أكثر كثافة (more compact representation)، وهذا يرجع إلى كونها مجموعة جزئية لمجموعة العناصر المتكررة المغلقة ($MFI \subseteq CFI \subseteq FI$)، الشكل 1.3 يوضح العلاقة بين مختلف تمثيلات مجموعات العناصر المتكررة [16].



شكل 1.3: العلاقة بين مختلف تمثيلات مجموعات العناصر المتكررة [16].

بالتالي، يمكن إستعادة جميع مجموعات العناصر المتكررة إنطلاقا من مجموعات العناصر المتكررة القصوى على النحو الآتي: أولا ينبغي إستخلاص كل المجموعات الجزئية الممكنة (all possible subsets) من داخل مجموعات العناصر المتكررة القصوى، وبعد ذلك وعبر مسح وحيد لقاعدة المعاملات يمكن حساب الدّاعم الخاص بكل مجموعة جزئية [16].

3.3 مجموعة العناصر المتكررة المغلقة Closed Frequent Itemset

يقال عن مجموعة عناصر متكررة X أنها مغلقة إذ لم توجد أي مجموعة عليا (Superset) Y تحتويها وتحمل نفس قيمة الداعم بالضبط كالمجموعة X. بعبارة أخرى:

$$X\ is\ closed\ FI : \nexists\ Y\ such\ that\ Y \supset X\ and\ sup(Y) = sup(X)$$

مثـال

لتكن قاعدة المعاملات TDB_1 التي تحتوي على معاملتين t_1 و t_2 بحيث:

$$t_1 = \{a_1, ..., a_{50}\}$$
$$t_2 = \{a_1, ..., a_{100}\}$$

بفرض أن العتبة الدنيا للداعم تساوي القيمة 1، كم عدد مجموعات العناصر المغلقة الموجودة في TDB_1؟

$\leftarrow$ بتطبيق القاعدة أعلاه، يمكن القول أنه توجد مجموعتان من مجموعات العناصر المتكررة المغلقة هما:

$$P_1 = \{a_1, ..., a_{50}\} : 2$$
$$P_2 = \{a_1, ..., a_{100}\} : 1$$

هذا يعني أن المجموعات المغلقة تعتبر تمثيلا مكثفا دون فقدان المعلومات الخاصة بالمجموعات المتكررة (Closed itemsets are a lossless representation)، أي أنها تقوم بتقليص أعداد المجموعات المتكررة مع حفظ معلومات الداعم لكل مجموعة متكررة، فيمكن القول مثلا أن مجموعة العناصر الجزئية $(a_2, ..., a_{40})$ تحمل داعما يساوي القيمة 2 (تم إستنتاج ذلك من مجموعة العناصر P_1)، أو أن مجموعة العناصر الجزئية $(a_5, ..., a_{51})$ تحمل داعما مساويا للقيمة 1 (تم إستنتاج ذلك من مجموعة العناصر P_2)، وغير ذلك من الأمثلة [10].

4.3 مجموعة العناصر المتكررة القصوى Maximal Frequent Itemset

تحمل مجموعة X إسم مجموعة عناصر متكررة قصوى (Maximal Frequent Itemset) إذا كانت هذه الأخيرة مجموعة متكررة، أي أن قيمة الداعم الخاص بها تكون أكبر من أو تساوي قيمة العتبة الدنيا المحددة للداعم μ، ولا توجد لها أي مجموعة عليا (superset) مباشِرة Y تحتويها وتكون متكررة هي الأخرى. يمكن التعبير عن ذلك كالآتي:

$$X \; is \; a \; maximal \; FI : \nexists \; Y \; such \; that \; Y \supset X \; and \; sup(Y) \geq \mu$$

فيكمن الفرق بين تمثيل العناصر المتكررة المغلقة وتمثيل العناصر المتكررة القصوى أن هذه الأخيرة لا تحمل قيمة الداعم الفعلية للمجموعات الجزئية لمجموعة العناصر المتكررة القصوى (MFI representations do not care the real support of the sub-patterns)، فبالعودة إلى مثال قاعدة المعاملات TDB_1 والتي تحتوي على المعاملتين:

$$t_1 = \{a_1, ..., a_{50}\}$$
$$t_2 = \{a_1, ..., a_{100}\}$$

وبأخذ قيمة العتبة الدنيا للداعم مساوية للقيمة 1 $(\mu = 1)$، كم يكون عدد مجموعات العناصر المتكررة القصوى المتواجدة في قاعدة المعاملات TDB_1؟

← توجد مجموعة عناصر متكررة قصوى وحيدة في هذه القاعدة ألا وهي المجموعة:

$$P = \{a_1, ..., a_{100}\} : 1$$

والسبب يعود إلى أنه على الرغم من وجود المجموعة $(a_1, ..., a_{50})$ بقيمة داعم مساوية للقيمة 2، إلا أن مجموعات العناصر المتكررة القصوى لا تهتم بمعلومات قيم الداعم، فيكفي القول بأن للمجموعة $(a_1, ..., a_{50})$ مجموعة متكررة عُليا تحتويها (frequent superset) وهي المجموعة $(a_1, ..., a_{100})$، هذا يعني أن تمثيل مجموعات العناصر المتكررة القصوى يُعتبر

تمثيلا مضغوطا لمجموعات العناصر مع فُقدان المعلومات الخاصة بالداعم
(maximal itemsets are a lossy compression)، بالتالي يمكن فقط معرفة أن مجموعة عناصر مَا تكون متكررة أم لا، فمثلا المجموعة العناصر $(a_1..., a_{50})$ متكررة كونها مجموعة جزئية لمجموعة العناصر المتكررة القصوى $(a_1, ..., a_{100})$، لكن لا يمكن معرفة قيمة الداعم الفعلية لها، وهكذا مع جميع الأمثلة الأخرى [10].

5.3 التعقيد الحسابي لعملية إيجاد مجموعات العناصر المتكررة القصوى

على الرّغم من تعدّدِ الخوارزميات المقترحة للتنقيب في البيانات وإستخراج مجموعات العناصر المتكررة القصوى (MFI)، إلا أن الدراسات المتعلِقة بتحديد طبيعة التعقيد الحسابي لمشاكل التنقيب في البيانات بحد ذاتهِ لازالت غير كافية.

تملك مشكلة التعقيد الحسابي لمجموعات العناصر المتكررة القصوى جانبين يتمثلان في تعداد (counting) جميع مجموعات العناصر المتكررة القصوى المختلفة الموجودة داخل قاعدة المعاملات أيّا كانت القيمة الدنيا للدّاعم (μ)، وأيضا في إيجاد (finding) هذه المجموعات من العناصر المتكررة القصوى.

تكون عملية التعداد ضمن P-Complete، أمّا عملية التنقيب عن مجموعات العناصر المتكررة القصوى فتكون ضمن NP-hard، إضافة إلى أن عملية التحقق من وجود مجموعة عناصر متكررة قصوى داخل قاعدة معاملات يكون ضمن P [18].

بالتّالي، فإنه لا توجد خوارزمية فعّالة (Efficient algorithm) لحل المشكلات بهذا النوع من التعقيد، بل تبقى مجرّد خوارزميات تجريبية (Heuristics) بدون ضمانات على كونِها حلولا مُثلى (optimal)، ولا يزال مشكل التعقيد الحسابي لمجموعات العناصر المتكررة القصوى مفتوحا إلى حد السّاعة [18].

وفيما يأتي ستتم دراسة موجزة لبعض الخوارزميات الموجودة حاليا لإستخراج مجموعات العناصر المتكررة القصوى.

1.5.3 خوارزمية MaxMiner

قدّم Roberto J.Byardo خوارزمية MaxMiner [11] سنة 1998 والتي تهدف إلى البحث عن مجموعة العناصر المتكررة القصوى (MFI) فقط، تمَ إستقاء هذه الخوارزمية من خلال الخوارزمية الشهيرة Apriori.

فمن أجل تخفيض حيّز البحث لا تقوم خوارزمية MaxMiner فقط بتقليم المجموعات الجزئية غير المتكررة بل وتستعمل أيضا تقنية النظر بخطوة إلى الأمام (Lookahead) من أجل القيام بعملية تقليم للمجموعات العليا غير المتكررة (Superset Prunning).
فمن أجل كلّ مجموعة عناصر X تقوم الخوارزمية بإيجاد جميع العناصر الفردية j التي يشكل إتحادها مع المجموعة X أي $(X \cup j)$ مجموعة عناصر متكررة، علما أنّ العنصر j لا ينتمي إلى مجموعة العناصر X $(j \notin X)$، بالتالي يمكن الإستنتاج بأنّ كل المجموعات الجزئية المنبثقة منها هي مجموعات عناصر متكررة.

بالرّغم من أنّ عملية تقليم المجموعات العليا تقوم بتخفيض زمن البحث بشكل كبير إلا أنّ خوارزمية MaxMiner لازالت تحتاج إلى العديد من عمليات مسح لقاعدة المعاملات للوصول إلى العناصر المتكررة القصوى [8].

2.5.3 خوارزمية Depth Project

توجد أيضا خوارزمية Depth Project المقدّمة من طرف Agrawal، Aggrawal و Prasad [1] سنة 2000 والتي تقوم بالبحث عن مجموعة العناصر المتكررة القصوى عبر القيام بمزج تقنيتي البحث العمودي والبحث الأفقي (Depth-first / Breadth first Traversal) داخل مجموعة العناصر.

يتم في هذه الخوارزمية إستعمال طريقة التقليم لكل من المجموعات الجزئية غير المتكررة والمجموعات العليا، بحيث تكون قاعدة المعاملات فيها ممثلة على شكل "خارطة ثنائية" (Bitmap)، يكون كل سطر (row) من الخارطة الثنائية متناسبا مع معاملة ما في قاعدة المعاملات ويسمى "شعاعا ثنائيا" (Bitvector)، كما يعبّر كل عمود (column) على عنصر مَا (item) من مجموعة العناصر.

يكون عدد الأسطر مساويا لعدد المعاملات (transactions) الموجودة في قاعدة المعاملات كما يكون عدد الأعمدة مساويا لعدد العناصر (Items) في القاعدة.

في كل سطر يوجد الرمز "1" في الموضع i إذا كانت المعاملة الموافقة لذلك السطر تشتمل على العنصر i، بالمقابل يوجد الرمز "0" في ذلك الموضع إذا لم يكن الأمر كذلك.

تُظهر الجداول 1.3 و 2.3 مثالا عن قاعدة المعاملات والتمثيل الخاص بالخارطة الثنائية (Bitmap) لهذه القاعدة.

t	$i(t)$
1	$ABDE$
2	BCE
3	$ABDE$
4	$ABCE$
5	$ABCDE$
6	BCD

جدول 1.3: قاعدة المعاملات.

D	A	B	C	D	E
1	1	1	1	1	1
2	0	1	1	0	1
3	1	1	0	1	1
4	1	1	1	0	1
5	1	1	1	1	1
6	0	1	1	1	0

جدول 2.3: تمثيل الخارطة الثنائية (Bitmap) لقاعدة المعاملات.

يعبّر عن داعم مجموعات العناصر (itemsets support) بعدد الأسطر التي تحمل الرمز "1" في المواضع الموافقة لعناصر تلك المجموعات، فعلى سبيل المثال يكون الدّاعم الخاص بمجموعة العناصر ABD مساويا للقيمة 3، كون الأسطر 1، 3 و5 تحمل معا الرمز "1" في المواضع الخاصة بكل من العناصر A, B و D على الترتيب.

فبطرق حسابية مصممة بعناية، تصبح هذه الخوارزمية قادرة على تقليص تكلفة إيجاد قيمة الداعم بشكل ملحوظ.

أظهرت النتائج التجريبية أن خوارزمية Depth Project تفوّقت على خوارزمية MaxMiner من حيث الأداء [8].

3.5.3 خوارزمية MAFIA

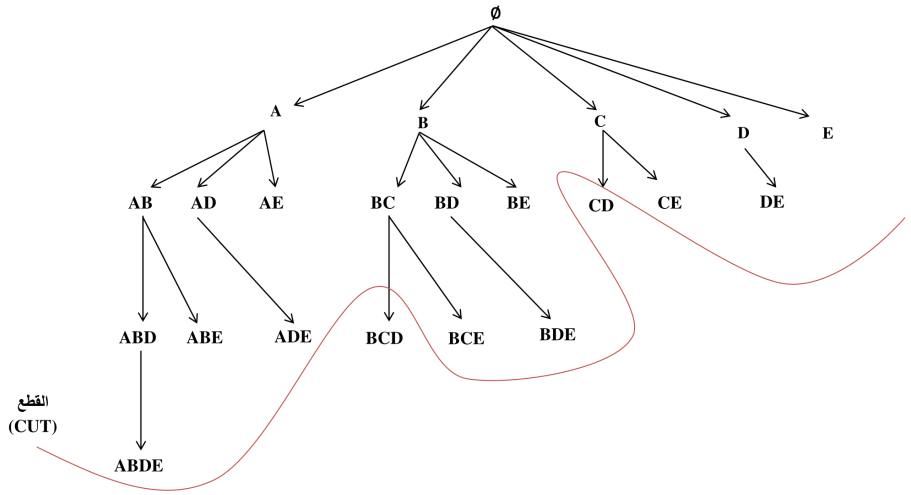
من خلال أوراقهم البحثية قدَم كل من Doug Burdick، Manuel calimlim و Johannes Gehrke [4] سنة 2001 خوارزمية MAFIA والتي تعني خوارزمية العناصر المتكررة القصوى (MAximal Frequent Itemset Algorithm)، فتقريبا وبنفس طريقة خوارزمية Depth Project يتم تمثيل قاعدة المعاملات في شكل خارطة ثنائية (Bitmap)، ويتم حساب الداعم (support) الخاص بمجموعة عناصر مَا من خلال الأعمدة (columns) في هذه الخارطة الثنائية من خلال تعداد ظهور الرمز"1" في الأعمدة الخاصة بمجموعات عناصر معيَنة، تسمى الخارطة الثنائية في هذه الخوارزمية بالخارطة الثنائية العمودية (vertical bitmap).

فبأخذ الخارطة الثنائية (Bitmap) في الجدول 2.3، تكون الأشعة الثنائية (Bitvectors) الخاصة بالعناصر A، B و D هي: 101110، 111111 و 101011 على الترتيب.
ومن أجل الحصول على الشعاع الثنائي (Bitvector) لأي مجموعة عناصر، يكفي تطبيق العملية المنطقية "and" بين الأشعة الثنائية للعناصر المكوِنة لتلك المجموعة فيكون الشعاع

الثنائي لمجموعة العناصر AB كالتالي:
$101110 \quad and \quad 111111$ أي 101110 ، بينما يمكن حساب الشعاع الثنائي الخاص بالمجموعة ABD من خلال الأشعة الثنائية لكل من AB و D بتطبيق العملية المنطقية AB and D كما يلي:
$101110 \quad and \quad 101011 = 101010$، أي أن مجموعة العناصر هذه تحمل داعما مساويا للقيمة 3 نتيجة ظهور الرمز "1" ثلاث مرات في شعاعها الثنائي (its Bitvector) [8].

تُعتبر خوارزمية MAFIA من خوارزميات المقاربة نحو العمق (depth-first algorithm) حيث أنها تقوم بتمثيل مجموعات العناصر في شكل شجرة مجموعات العناصر المرشحة (candidate itemset tree) وبإستخدام عملية القطع (cut)، تجوب الخوارزمية بشكل عمودي هذه الشجرة بهدف وضع حدود (barriers) ما بين كل مجموعة عناصر متكررة والمجموعة العليا (superset) غير المتكررة التي تليها مباشرة، لتكون بالتالي جميع مجموعات العناصر المتواجدة فوق حدود القطع (cut) بشكل مباشر مجموعات عناصر متكررة (FI)، أي أنه لن تكون هناك مجموعات عناصر متواجدة تحت القَطع يمكن لها أن تكون متكررة كونها تحتوي عناصر تمّ التوصل إلى أنها عناصر غير متكررة (infrequent).

يُظهر المثال في الشكل 2.3 كيفية عمل خوارزمية MAFIA على قاعدة المعاملات المُبينة في الجدول 1.3، وتحديد العتبة الدنيا للداعم بالقيمة 3 $(3 = \mu)$ [4].



شكل 2.3: طريقة عمل خوارزمية MAFIA [4].

4.5.3 خوارزمية FPmax

قام كل من الباحثين Gosta Grahne و Jianfei Zhu [8] من جامعة Concordia الكندية سنة 2003، بتقديم خوارزمية تعتبر إمتدادا لخورزمية FPGrowth، أطلق عليها إسمFPmax كونها تهتم فقط بإستخراج العناصر المتكررة القصوى بإستخدام هيكلة جديدة تسمى شجرة العناصر المتكررة القصوى ((Maximal Frequent Itemset-tree(MFI-tree).

على غرار خوارزمية FPGrowth، فإن خوارزمية FPmax تعتبر أيضا خوارزمية تراجعية (Recursive Algorithm).
تعمل خوارزمية FPmax وفقا للفرضية التي تنص على أن مجموعة عناصر متكررة قصوى لا يمكن لها أن تكون مجموعة جزئية لأي مجموعة متكررة يتم انشاؤها لاحقا (Future Frequent Itemset)، هذا يعني أن أي مجموعة عناصر متكررة إمّا أن تكون هي نفسها مجموعة عناصر متكررة قصوى (MFI) أو أن تكون مجموعة جزئية لمجموعة عناصر متكررة قصوى موجودة سابقا في هيكلة شجرة العناصر المتكررة القصوى (MFI-tree) التي تستعملها الخوارزمية كما يلي:

شجرة العناصر المتكررة القصوى هي هيكلة خاصة تشبه إلى حد بعيد هيكلة شجرة النمط المتكرر (FP-Tree)، وتستعمل لجعل عمليات التحقق من المجموعات الجزئية (Subset Cheking) أكثر فعالية.
تعرَّف شجرة العناصر المتكررة القصوى (MFI-Tree) من خلال العقدة التي تحتوي العنصر الخالي والتي تسمى بالجذر (root)، إضافة إلى قائمة مرتبطة (Linked List) تسمى باللائحة (Head)، تحتوي على العناصر بترتيب مماثل للائحة العناصر المتكررة (FI Header Table) الخاصة بشجرة النمط المتكرر (FP-Tree).

تبدأ عملية إنشاء هيكلة شجرة العناصر المتكررة القصوى (MFI-Tree) إنطلاقا من العنصر الأقل تكرارا في اللائحة (Head)، بحيث يتم إستخراج مجموعة العناصر المتكررة الشرطية (Conditional Frequent Itemsets) الخاصة به من خلال شجرة الأنماط الشرطية (Conditional FP-tree) لذلك العنصر ، لتتم بعدها عملية التحقق مما إذا كانت هذه المجموعة عبارة عن مجموعة جزئية لأي من مجموعات العناصر المتكررة القصوى الموجودة سابقا في شجرة العناصر المتكررة القصوى (MFI-Tree) من خلال الإجراء (Function) المسمى (Subset Cheking)، فإذا كانت هذه المجموعة تمثل مجموعة جزئية لمجموعة عناصر متكررة قصوى موجودة سابقا، فلن يتم إضافتها إلى هيكلة شجرة العناصر المتكررة القصوى (MFI-Tree)، وإذا لم يكن الأمر كذلك فسيتم إضافتها إلى الهيكلة.

ثم يتم الإنتقال إلى العناصر الموالية في اللائحة (Head) بحيث يتم إدراج مجموعة العناصر المتكررة الشرطية الخاصة بكل منها في شجرة العناصر المتكررة القصوى (MFI-Tree) مالم تكن مجموعة جزئية لمجموعة عناصر متكررة قصوى موجودة مسبقا في هيكلة الشجرة.

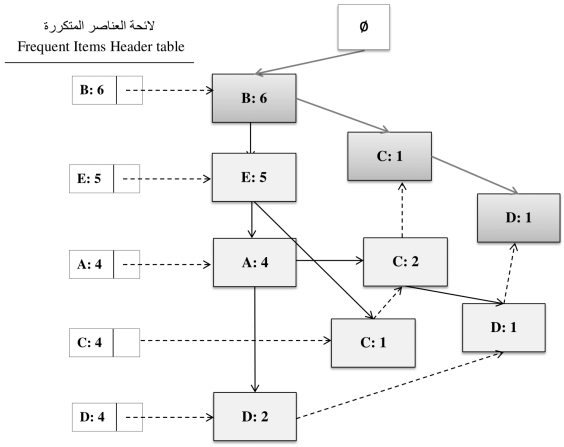
تحتفظ هيكلة شجرة العناصر المتكررة القصوى (MFI-Tree) في الأخير، فقط بمجموعات العناصر المتكررة القصوى (MFI) [8].

المثال الموالي يمثل آلية عمل خورازمية FPMax إعتمادا على الجدول 3.3 ومن خلال الشكل 3.3 [8].

العنصر item	قاعدة الأنماط الشرطية (Conditional Pattern Base)	شجرة الأنماط المتكررة الشرطية (Conditional FP-tree)
D	$< \{B, E, A : 2\}, \{B, E, A, C : 1\} > < \{B, C : 1\} >$	$\{B, E, A : 3\}$ $\{B : 4\}$
C	$< \{B, E, A : 2\}, \{B, E : 1\} > < \{B : 1\} >$	$\{B, : 4\}, \{E : 3\}$
A	$\{B, E : 4\}$	$\{B : 4\}, \{E : 4\}$
E	$\{B : 5\}$	$\{B : 5\}$

جدول 3.3: قاعدة الأنماط الشرطية وشجرة الأنماط المتكررة الشرطية

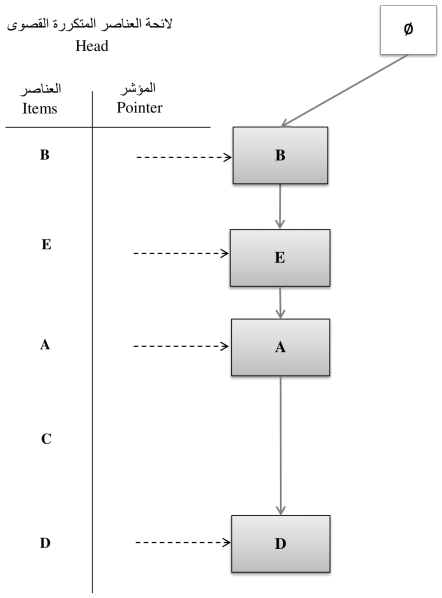
.



شكل 3.3: شجرة النمط المتكرر FP-tree.

تبدأ عملية إنشاء شجرة العناصر المتكررة القصوى (MFI-Tree) إنطلاقا من العنصر الأقل تكرارا نحو العنصر الأكثر تكرارا، كما يلي:

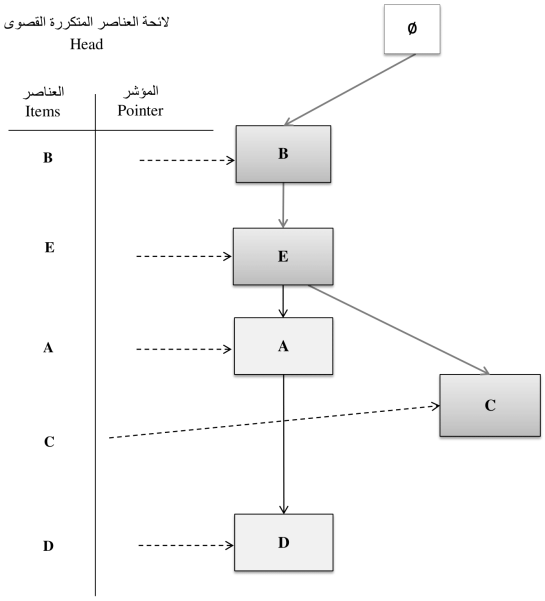
- العنصر D: تحتوي شجرة الأنماط المتكررة الشرطية (Conditional FP-tree) الخاصة به على مسار وحيد (One single Path)، بالتالي يتم الحصول على مجموعة العناصر المتكررة الشرطية $\{B, E, A, D\}$، حيث يظهر بشكل جلي أنها مجموعة عناصر متكررة قصوى (MFI) كون شجرة العناصر المتكررة القصوى (MFI-Tree) لا تحتوي على مجموعة عناصر متكررة قصوى مسبقة، فيتم إضافة هذه المجموعة إلى الجذر (root) مباشرة كما يوضحه الشكل 4.3.



شكل 4.3: إضافة العنصر D ضمن شجرة العناصر المتكررة القصوى MFI-Tree.

تأخذ لائحة العناصر المتكررة القصوى (Head) لشجرة العناصر المتكررة القصوى (MFI-Tree) نفس تمثيل لائحة العناصر المتكررة لشجرة النمط المتكرر (FP-Tree)، لكنها تقوم بإهمال قيم الداعم للعناصر.

- العنصر C: يتم الإنتقال إلى العنصر الموالي C حيث يتم الحصول على مجموعة العناصر المتكررة الشرطية $\{B, E, C\}$، وبما أن العنصر C الموجود في اللائحة، لا يملك أيَّة روابط تؤشر نحو العقد في شجرة العناصر المتكررة القصوى (MFI-Tree)، هذا يعني أنه لا توجد مجموعة عليا (Superset) تحتوي هذه المجموعة من العناصر المتكررة الشرطية، فيتم إدراج مجموعة العناصر المتكررة القصوى $\{B, E, C\}$ في داخل شجرة العناصر المتكررة القصوى (MFI-Tree) كما يظهره الشكل 5.3.

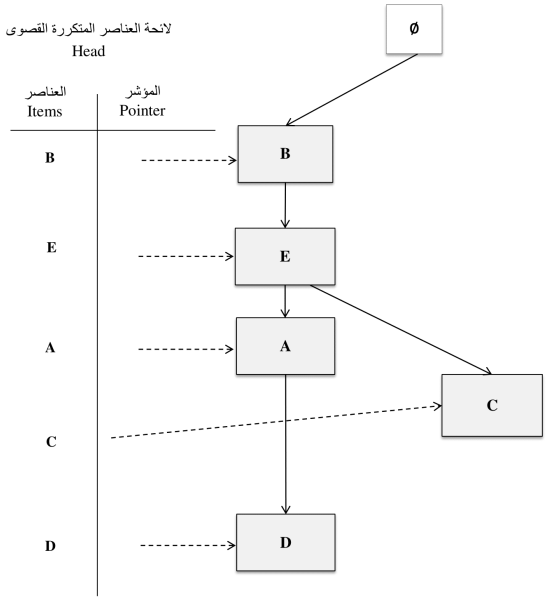


شكل 5.3: إضافة العنصر C ضمن شجرة العناصر المتكررة القصوى MFI-Tree.

- العنصر A: مجموعة العناصر الشرطية الوحيدة الخاصة به هي $\{B, E, A\}$، بعد القيام بعملية التحقق يظهر أن هذه المجموعة هي مجموعة جزئية لمجموعة عناصر متكررة قصوى سابقة، وبالتالي لن يتم إضافة هذه المجموعة ضمن شجرة العناصر المتكررة القصوى.

- العنصر E: يتم المرور إلى العنصر الأخير E، العنصر الأكثر تكرارا في اللائحة، حيث يلاحَظ أن مجموعة العناصر الشرطية $\{B, E\}$ الخاصة به، تعد مجموعة جزئية لمجموعة عليا (Superset) في شجرة العناصر المتكررة القصوى (MFI-Tree)، فلا يمكن إذن إضافة هذه المجموعة الشرطية إليها.

يمثل كل فرع من شجرة العناصر المتكررة القصوى (MFI-Tree(MFI)) مجموعة عناصر متكررة قصوى، بالتالي تكون مجموعات العناصر المتكررة القصوى كالآتي: $\{B, E, C\}$ $\{B, E, A, D\}$ كما يبينه الشكل 6.3.



شكل 6.3: الشكل النهائي لهيكلة شجرة العناصر المتكررة القصوى MFI-Tree.

الخوارزمية 6 أدناه توضح آلية عمل خوارزمية الأنماط المتكررة القصوى (**FPMax**) من خلال سلسلة الأوامر البرمجية [8].

خوارزمية 6 سلسلة الأوامر البرمجية لخوارزمية الأنماط المتكررة القصوى FPMax.

خوارزمية	FPMax (T)
المدخلات	T هيكل شجرة العناصر المتكررة $MFIT$ هيكل شجرة العناصر المتكررة القصوى $Head$ لائحة العناصر
المخرجات	الهيكل النهائي لشجرة العناصر المتكررة القصوى تحتوي على $MFIT$ جميع مجموعة العناصر المتكررة القصوى
1	**if** P only contains a single path P
2	Insert $Head \cup P$ into $MFIT$
3	**else foreach** i In Header-Table of T
4	Append i to $Head$;
5	Construct the Head-pattern base
6	$Tail$ = { frequent items in base }
7	$Subsetcheking(Head \cup Tail)$;
8	**if** $Head \cup Tail$ is not $MFIT$
9	Construct the FP-Tree T_{Head};
10	Call FPMAX (T_{Head})
11	remove i from $Head$.

5.5.3 خوارزمية LCM Max

خوارزمية LCM هي إختصار لـ Linear time Closed itemset Miner اقترحت في سنة 2004 [17] من قبل الباحثين Uno Takeaki، Tatsuya Asai، Yuzo Uchida و Hiroki Arimura بالمعهد الوطني للمعلوماتية (National Institute of Informatics) بطوكيو (Tokyo) كواحدة من عدة خوارزميات لتعداد العناصر المتكررة [8]:

LCM: لتعداد مجموعات العناصر المتكررة المغلقة من قاعدة المعاملات.

LCM-freq: للتعداد الكلي لمجموعات العناصر المتكررة من قاعدة المعاملات.

LCM Max: لتعداد مجموعات العناصر المتكررة القصوى من قاعدة المعاملات.

يتم الحصول على خوارزمية LCM Max من خلال إضافة التحقق الصريح (Explicit Check) لخوارزمية LCM عمَا إذا كانت مجموعة العناصر الحالية عبارة عن مجموعة عناصر متكررة قصوى (Maximal FI).

تتم عملية تعداد العناصر المتكررة القصوى بإستخدام مجموعات العناصر المتكررة المغلقة (CFI)، فالفكرة الأساسية لهذه الخوارزمية تعتمد على كون أي مجموعة عناصر متكررة قصوى هي بالتأكيد مجموعة عناصر متكررة مغلقة، بالتالي يتم تعداد مجموعات العناصر المغلقة (CFI) وإستخلاص فقط تلك التي تكون مجموعات عناصر متكررة قصوى (MFI).

بزيادة الشرط التالي في خوارزمية LCM يتم الحصول على خوارزمية LCM Max: من أجل كل عناصر متكررة مغلقة X: تكون X مجموعة عناصر متكررة قصوى إذا وفقط إذا كان $X \cup \{i\}$ يعبر عن مجموعة عناصر غير متكررة (infrequent) من أجل كل $i \notin X$، هذه الإضافة لا تؤثر على المساحة التخزينية للذاكرة، لكنها تزيد في الزمن المستغرق للخوارزمية لأنه في كل تكرار يتم القيام بهذا التحقق.

فخوارزمية LCM Max تبحث في طرق إيجاد مجموعات العناصر المتكررة القصوى دون إستخدام هيكلة كبيرة للبيانات، من خلال بعض التحسينات منها: تخفيض حجم قاعدة المعاملات كلما تقدمت الخوارزمية من مرحلة إلى أخرى مما يفيد في خفض إستهلاك المساحة التخزينية للذاكرة وتحسين زمن الإستجابة (Response time) من المجموعات القصوى جعلت منها فعالة في معالجة بعض قواعد المعاملات المتناثرة (Sparse Databases).

6.5.3 خوارزمية GenMax

أقتُرحت خوارزمية GenMax من قبل الباحثين Mohammed.J Zaki وKaram Gouda سنة 2005 [7]، تنص فكرة الخوارزمية على أنّ عملية البحث عن العناصر المتكررة القصوى تتطلب خطوة إضافية بعد تحديد مجموعات العناصر المتكررة، بحيث يتم إنشاء مجموعة M من أجل وضع مجموعات العناصر المتكررة القصوى فقط داخلها، تكون هذه المجموعة في البداية مجموعة خالية $(M = \emptyset)$ وفي كل مرة يتم جلب مجموعة عناصر متكررة X من داخل مجموعة العناصر المحتملة P، تتم عملية التحقق من المجموعات الجزئية والمجموعات العليا لها قبل وضعها داخل المجموعة M، ففي عملية تحقق المجموعات الجزئية يتم التأكد من أنّ المجموعة X المحتملة لأن تكون مجموعة عناصر متكررة قصوى ليست متضمَنة في أيّ مجموعة من مجموعات العناصر المتكررة القصوى الموجودة داخل المجموعة M، أمّا فيما يخص تحقق المجموعات العليا فيتمّ إضافة العنصر الموالي في مجموعة العناصر المحتملة P إلى المجموعة X، ليتمّ التحقق بعدها ممّا إذا كانت المجموعة الناتجة عبارة عن مجموعة متضمَنة داخل المجموعة M فإذا كانت كذلك يتمّ إزالتها، والاّ فيتم إدراجها داخلها.

تعتمد خوارزمية GenMax التراجعية على عملية تقاطع معرِّفات المعامَلات (Tidsets Intersection) المستوحاة من خوارزمية Eclat، الخوارزمية المرجِع للمقاربة العمودية، وهذا ما يجعلها لا تضع أيّ مجموعة عناصر محتملة لأن تكون مجموعة عناصر متكررة قصوى داخل المجموعة M إلاّ بعد التحقق الكامل من أنّها فعلا مجموعة عناصر متكررة قصوى.

هذا التحقق يجعل من المجموعة M تحتوي فقط على مجموعات العناصر المتكررة القصوى النهائية، بالتّالي لا يمكن حذف أيّ مجموعة عناصر تمّت إضافتها إليها [20].

مثال

يشرح المثال الموضح في الشكل 7.3 طريقة عمل خوارزمية GenMax، فبإستعمال قاعدة المعاملات المبينة في الجدول 1.3، وبتحديد قيمة العتبة الدنيا للداعم بالقيمة 3 $(3 = \mu)$ تكون في البداية المجموعة M التي تحتوي مجموعة العناصر المتكررة القصوى عبارة عن مجموعة خالية $(M = \emptyset)$.
تبدأ العملية من خلال مجموعة العناصر الفردية المتكررة الموجودة في مجموعة العناصر الكلية، والتي تعتبر مجموعة العناصر المحتملة للمجموعة الخالية $(\emptyset)$ كما يلي [20]:

$$P_\phi = \{< A, 1345 >, < B, 123456 >, < C, 2456 >, \\ < D1356 >, < E, 12345 >\}$$

تتم بعد ذلك عملية تقاطع معرِّفات معامَلات العنصر الأول A مع معرِّفات معامَلات العناصر الموالية الأخرى، لتصبح مجموعة العناصر المتكررة المُلحقة للعنصر A (Fre-quent Extension from A) كالتالي

$P_A = \{< AB, 1345 >, < AD, 135 >, < AE, 1345 >\}$ (تمّ حذف مجموعة العناصر AC كونها مجموعة عناصر غير متكررة).

يؤدي إختيار المجموعة المحتملة $X = AB$ إلى إنشاء المجموعة الملحقة الموالية كالتالي:

$P_{AB} = \{< ABD, 135 >, < ABE, 1345 >\}$ ثم أخيرا، يتم الوصول إلى المجموعة الموجودة أقصى اليسار في الشكل 7.3، والتي تُعتبر المجموعة الملحقة لمجموعة العناصر ABD كالتالي: $P_{ABD} = \{ABDE, 135\}$

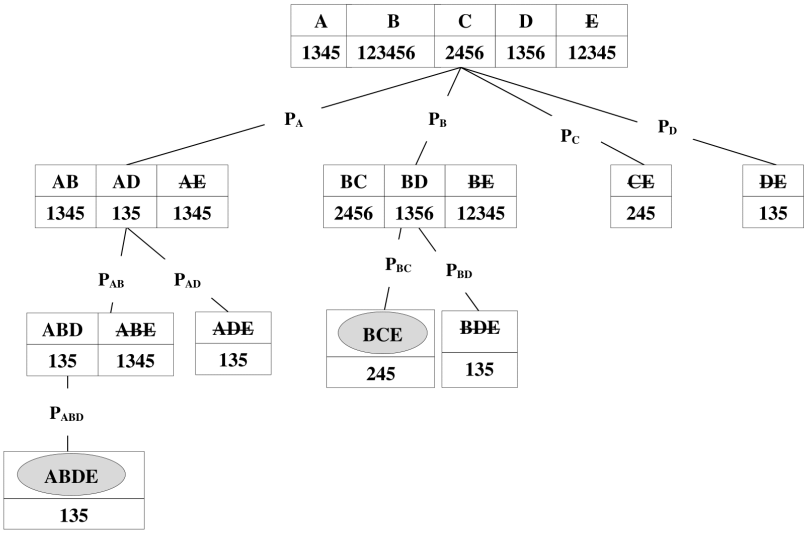
لتوضع المجموعةُ $ABDE$ بعد هذه الخطوة داخل مجموعة العناصر المتكررة النهائية M كونها لا تملك أي مجموعة عناصر موالية ملحقة لها، ولكون مجموعة العناصر المتكررة القصوى النهائية لازالت مجموعة خالية في هذه المرحلة، لتصبح هذه الأخيرة كالتالي: $M = \{ABDE\}$.

ثم تتراجع عملية البحث بمستوى واحد (one level backtrack) من أجل القيام بعملية التحقق من المجموعة ABE والتي تعتبر أيضا إحدى المجموعات المحتملة لأن تكون مجموعة عناصر متكررة قصوى، ليلاحَظ بأن هذه الأخيرة مُتضمَنة داخل مجموعة العناصر المتكررة القصوى $ABDE$، بالتالي تتم عملية حذفها.

وبنفس الطريقة تتم معالجة المجموعة الملحقة للمجموعة AD والتي هي كالتالي: $P_{AD} = \{< ADE, 135 >\}$، ليتم أيضا حذفها كونها مجموعة مُتضمَنة مُسبقا داخل المجموعة المتكررة القصوى $ABDE$، وكذلك الأمر بالنسبة للمجموعة AE.

بعد هذه الخطوة، تكون جميع مجموعات العناصر المتكررة القصوى المبتدئة بالعنصر A متواجدة داخل المجموعة M، ليتم بعدها الإنتقال والمتابعة مع مجموعات العناصر المبتدئة بالعنصر B، وبعد الوصول إلى عمق أقصى اليسار أين توجد مجموعة العناصر BCE التي لا يمكن إلحاق أي مجموعة عناصر أخرى بها، ولكون هذه المجموعة ليست مُتضمَنة في أي مجموعة عناصر متكررة قصوى داخل المجموعة M يتم إدراجها كمجموعة عناصر متكررة قصوى في M، لتصبح بذلك هذه الأخيرة كالتالي $M = \{ABDE, BCE\}$.

بعدها تُحذف كل مجموعات العناصر المُتبقية كونها مُتضمَنة في إحدى مجموعتي العناصر المتكررة القصوى $ABDE$ أو BCE.



شكل 7.3: طريقة عمل خوارزمية GenMax.
المجموعات المحاطة بدوائر رمادية تمثل المجموعات المتكررة القصوى MFI [20].

7.5.3 خوارزمية Charm-MFI

فكرة إنتقاء مجموعات العناصر المتكررة القصوى من بين مجموعات العناصر المتكررة المغلقة المستخدمة في خوارزمية Charm-MFI المقترحة من قبل الباحث Laszlo Szathmary [16] سنة 2006، هي كالتالي:

عند كل خطوة i يتم وسم جميع العناصر المتكررة المغلقة ذات الطول (i-long CFI) على أنها مجموعات "قصوى" (Maximal)، وعند التكرار الموالي أي $i+1$ يتم التحقق مما إذا كانت مجموعة العناصر المتكررة المغلقة ذات الطول $i+1$ تحمل مجموعة جزئية من مجموعات الطول i (i-long subset)، فإذا كان الأمر كذلك فإن المجموعة الجزئية ذات الطول i ليست مجموعة عناصر متكررة قصوى لأنها تملك مجموعة عليا متكررة (Frequent Superset)، بالتالي يتم وسم تلك المجموعة على أنها "ليست قصوى" (not Maximal).

بعد مرور الخوارزمية على جميع مجموعات العناصر المغلقة، تكون مجموعات العناصر المتكررة القصوى النهائية هي المجموعات التي بقيت تحمل وسم مجموعات قصوى (Maximal).

بالتالي يمكن إستنتاج أن أطول مجموعة عناصر متكررة مغلقة تكون دائما مجموعة عناصر متكررة قصوى .

يمكن إضافة هذه الفكرة في أي خوارزمية تبحث عن المجموعات العناصر المغلقة لتجعل منها خوارزمية لإيجاد مجموعة العناصر المتكررة القصوى شرط أن تكون مجموعات العناصر المتكررة المغلقة في البداية مرتّبة ترتيبا تنازليا حسب طول كل مجموعة عناصر [16].

الخوارزمية 7 أدناه توضح آلية عمل خوارزمية Charm-MFI من خلال سلسلة الأوامر البرمجية [16].

خوارزمية 7 سلسلة الأوامر البرمجية لخوارزمية Charm-MFI.

خوارزمية	CharmMFI (CFI)
المدخلات	مجموعة العناصر المتكررة المغلقة CFI
المخرجات	مجموعة العناصر المتكررة القصوى MFI
1	**maxItemsetLength** $\longleftarrow$ (Length of the largest CFI);
2	$T_1 \longleftarrow readTable(1)$;
3	**for**$(i \longleftarrow 1; i < maxItemsetLength; ++i)$
4	{
5	$T_{i+1} \longleftarrow readTable(i+1)$;
6	findMaximalFrequentItemsets (T_{i+1}, T_i);
7	}
8	**Procedure** findMaximalFrequentItemsets
	// T_{i+1}: جدول مجموعة العناصر المتكررة المغلقة بطول $i+1$
	// T_i: جدول مجموعة العناصر المتكررة المغلقة بطول i
9	loop over the rows of $T_{i+1}(sup)$
10	{
11	$S \longleftarrow Subsets(T_i, sup)$; //إيجاد المجموعات الجزئية بالدعم sup في T_i
12	loop over the elements of $S(sub)$:
13	$sub.maximal \longleftarrow false$;
14	}

6.3 خاتمة

مع نهاية الفصل تمَ التعرف على التمثيلات المتراصَة (Compressed Representations) لمجموعات العناصر المتكررة، كتمثيلات مجموعات العناصر المتكررة المغلقة (CFI) وتمثيلات مجموعات العناصر المتكررة القصوى (MFI)، والعلاقة الموجودة بين مختلف هذه التمثيلات.

كما تمَ تسليط الضوء على التعقيد الحسابي لعملية إيجاد مجموعات العناصر المتكررة القصوى، إضافة إلى إلقاء نظرة حول بعض الخوارزميات الحالية التي تبحث في عمليات إيجاد مجموعات العناصر المتكررة القصوى.

وسيتم في الفصل الموالي عرض دراسة تطبيقية تتمثل في المقارنة بين خوارزميتين من خوارزميات إيجاد مجموعات العناصر المتكررة المتكررة القصوى من حيث الزمن المستغرق للتنفيذ وسعة إستهلاك الذاكرة.

الفصل الرابع

دراسة تجريبية

Experimental Analysis

1.4 مقدمة

بعد التعرف في الفصل السابق على التمثيلات المتراصة لمجموعات العناصر المتكررة، وإلقاء نظرة عامة حول الخوارزميات الحالية المتبعة لإيجاد مجموعات العناصر المتكررة القصوى، سنقوم بإجراء دراسة تجريبية تتمثل في مقارنة بين إثنتين من هذه الخوارزميات والموجودتين ضمن منصة SPMF مفتوحة المصدر، حيث ستتركز هذه المقارنة حول عاملي الزمن المستغرق لتنفيذ الخوارزمية وسعة المساحة التخزينية للذاكرة المستهلكة أثناء التنفيذ عبر مجموعات مختلفة من قواعد المعاملات.

2.4 التعريف بمنصة SPMF

SPMF هي إختصار لـ Sequential Pattern Mining Framework وتعني منصّة التنقيب عن الأنماط المتسلسلة، تعدّ من أشهر المكتبات مفتوحة المصدر المكتوبة بلغة الجافا، أُنشأت من قبل الباحث Philippe Fournier-Viger ومساعديه أواخر العام 2008 [6].

تتضمن هذه المكتبة 129 خوارزمية متخصصة في عمليات التنقيب في البيانات، منها ما يهتم بإستخراج البيانات التي تهدف إلى إستخلاص الأنماط المهمة، وأيضا إلى إستنباط قواعد الإرتباط الكامنة داخل قواعد المعاملات، إلى غير ذلك من العمليات.

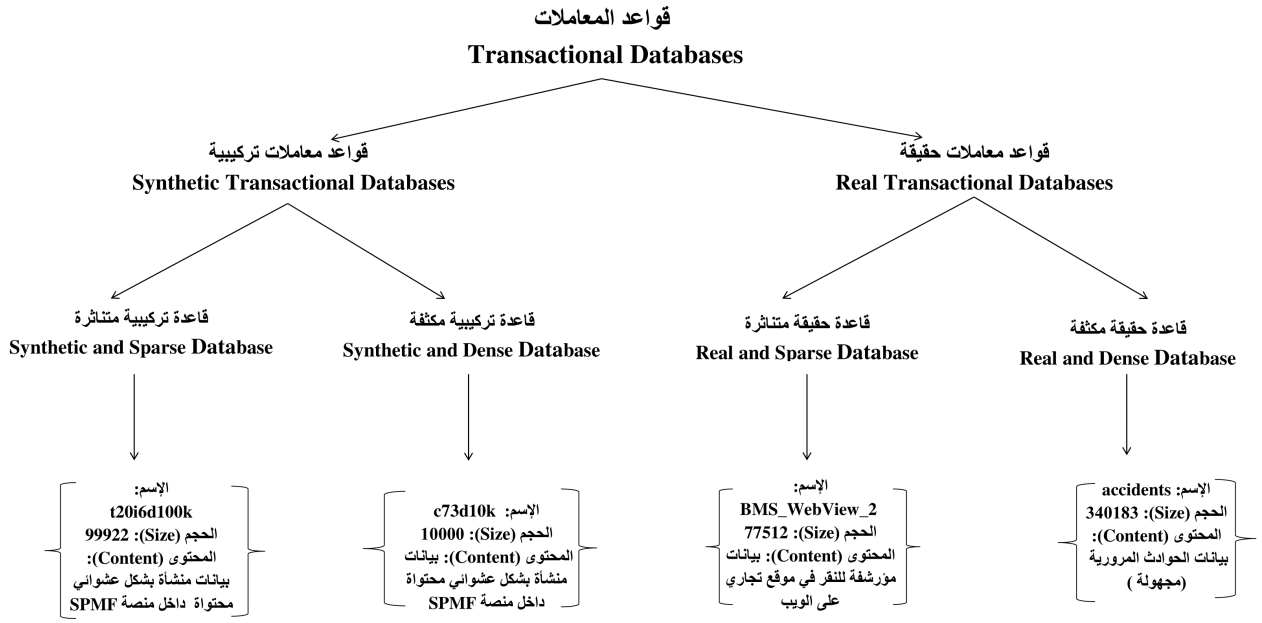
وفيما يخص عمليات إستخراج الأنماط المتكررة تحتوي منصة SPMF على 32 خوارزمية متخصصة، تتخللها 2 من الخوارزميات لإستخراج الأنماط المتكررة القصوى

(MFI). الإصدار الحالي لهذه المكتبة هو v2.13 تم نشره في 16 مارس 2017.

3.4 قواعد المعاملات الخاضعة للتجربة

تشمل المقارنة المراد إجراؤها مجموعة من قواعد المعاملات المدمجة داخل منصة SPMF والمختارة بشكل عشوائي لتُغطي جميع الجوانب الممكنة من حيث أنواع قواعد المعاملات الموجودة، فنجد قواعد المعاملات الحقيقية بنوعيها: المكثفة والمتناثرة والتي تمّ إستقاؤها من الواقع، إضافة إلى قواعد المعاملات التركيبية المكثفة والمتناثرة كذلك والتي تمّ إنشاؤها بشكل عشوائي

يوضح الشكل 1.4 أنواع قواعد المعاملات الخاضعة للتطبيق التجريبي.



شكل 1.4: قواعد المعاملات الخاضعة للتجربة.

4.4 الخطوات التجريبية Experiment Protocol

إتبعنا من أجل تحقيق التطبيق التجريبي سلسلة من الخطوات كما تظهره الخوارزمية 8 أدناه، حيث إختصت عملية المقارنة حول عاملين هما: الزمن المستغرق الذي يقاس بالثواني (seconds)، والمساحة التخزينية للذاكرة والتي تقاس بالميغابايت (MigaByte).

خوارزمية 8 سلسلة الأوامر البرمجية للخوارزمية التجريبية (Experiment).

خوارزمية	Experiment (Algorithms-set, Datasets, μ, max-times)
المدخلات	Algorithms-set مجموعة الخوارزميات المعتبرة في الدّراسة Datasets قواعد المعاملات المعتبرة في الدّراسة max-times العتبة الدنيا للدّاعم μ، عدد التكرارات
المخرجات	time results files, memory results files
1	**foreach** algorithm in Algorithms-set **do** // كل الخوارزميات
2	**foreach** dataset in Datasets **do** // كل قواعد المعاملات
3	**for** μ from 0.9 down to 0.02 **do** // كل القيم الممكنة للداعم
4	**repeat** max-times // بأي عدد $max - times$ يمكن تحديد قيمة
5	instantiate the algorithm
6	Run the algorithm with the specified parameters (dataset, μ)
7	get the average time results
8	record it in time results files
9	get the average memory results
10	record it in memory results files
11	**if** there is Exception **then**
12	write a specific value in the results files // كتابة قيمة معيّنة
13	continue // في ملف النتائج ثم المتابعة

يستحسن إلحاق قيمة مرتفعة للمتغير max-times مثلا 1000 أو 1500 من أجل الحصول على نتائج أكثر دقة، لكن نظرا لضيق الوقت فلقد قمنا بتكرار التنفيذ 100 مرة فقط.

تجدر الإشارة إلى أن التجارب إقتصرت فقط على خوارزميتي FPmax و Charm-MFI لكون هاتين الأخيرة هما خوارزميتي إيجاد مجموعات العناصر المتكررة القصوى الوحيدتين اللتان تملكان برامج تنفيذية (Implementation) في منصة SPMF، أما باقي الخوارزميات فإما لازالت بدون برنامج تنفيذي بعد، أو تمّ إيقاف برنامجها التنفيذي مؤقتا من المنصة بسبب بعض الأخطاء أو الأجزاء غير المكتملة فيه.

من أجل تتبع الزمن المستغرق تم إستخدام الإجراءين getstartTimestamp() و getEndTime() الموجودين مسبقا في البرنامج التنفيذي لكل خوارزمية منهما داخل SPMF، واللذين يستعملان وظيفة النظام system.currenttimemillis() التي تقوم برصد الزمن الحالي للنظام وقتَ التنفيذ بالميللي ثانية ثم بعد ذلك تم تحويل هذه القيم للثواني، أما بالنسبة لسعة الذاكرة، فتم إستخدام الإجراء getMaxMemory() الموجودة في الأداة MemoryLogger والذي يستعمل لأجل تسجيل أقصى حجم للذاكرة المستعملة أثناء تنفيذ الخوارزمية.

بالنسبة إلى العتاد فقد قمنا بإستخدام جهاز كومبيوتر core i3 بمعالج 2.30 GHz وذاكرة بحجم 6 GB يعمل بنظام تشغيل Windows 10 x64-based Processor، أما بالنسبة إلى إصدار جافا فهو JAVA Version 8 Update 112 ,Copyright (c) 2016.

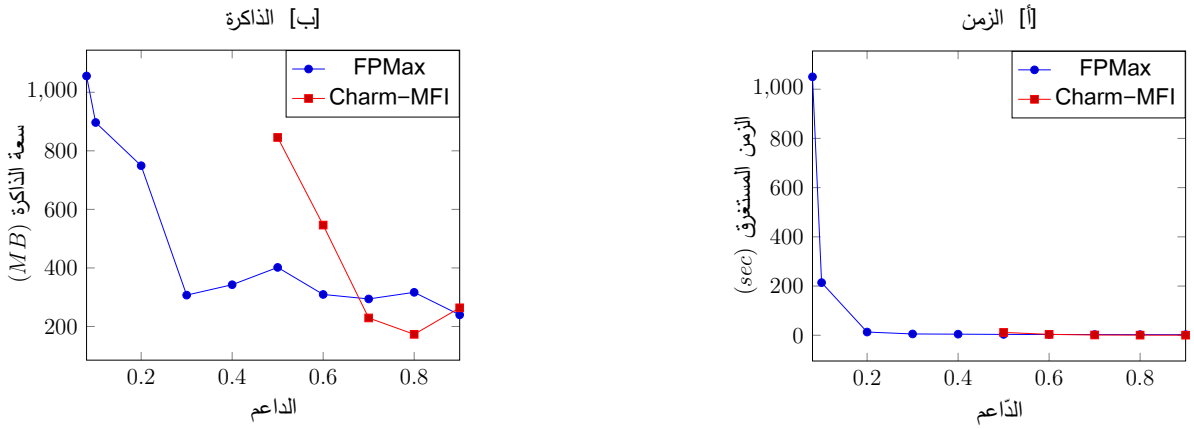
تمّ تسجيل البيانات الخاصة بكل من الزمن المُستغرق وسعة إستهلاك الذاكرة داخل ملفات نصية بهدف إستعمالها لإنشاء التمثيلات البيانية، وتمّ تعريف قيمة مُعيّنة تُبين وجود خطأ (Exception) كتجاوز سعة الذاكرة مثلا (Memory Overflow) والذي يُعتبر من الأخطاء الشائعة في مثل هذه التجارب التطبيقية.

5.4 مقارنة النتائج

فيما يلي تظهر التمثيلات البيانية للنتائج التجريبية المتحصّل عليها:

– قواعد المعاملات المكثفة (Condensed):

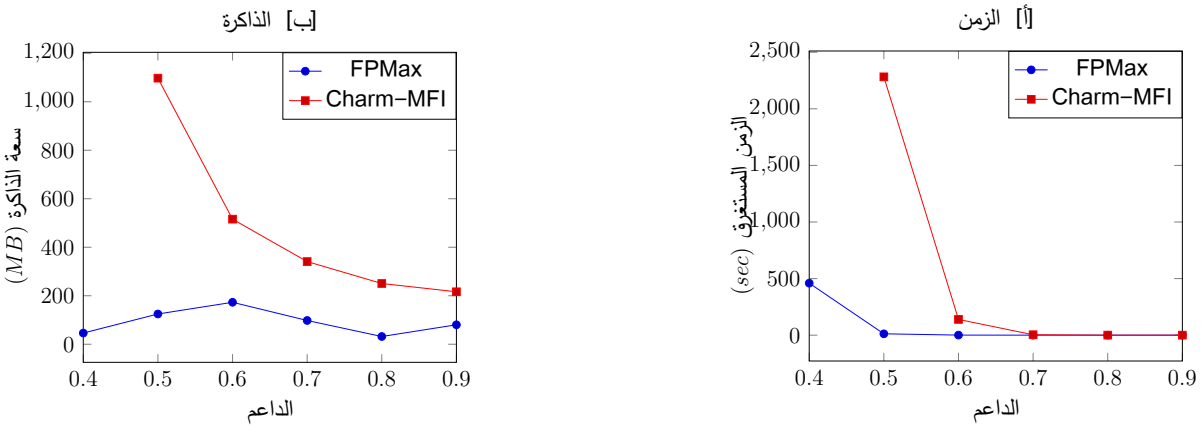
- أظهر التمثيل البياني الشكل 2.4 للنتائج التجريبية لقاعدة المعاملات الحقيقية accidents بالنظر إلى عامل الزمن المستغرق كما هو مبين في الجزء [أ] للشكل أنّ كلاّ من خوارزميتي FPMax و Charm-MFI إستغرقتا نفس المدّة من الزمن عند التنفيذ والتي تقدر بـ 3_{sec} تقريبا وهذا عندما يكون الدّاعم في المجال $[0.5, 0.9]$، لينقطع التمثيل البياني لخوارزمية Charm-MFI ويتواصل تمثيل خوارزمية FPMax نحو القيم المنخفضة للدّاعم أين وصلا حتى القيمة 0.08 بإرتفاع طفيف للزمن المستغرق $[4_{sec}, 13_{sec}]$ عندما يكون الدّاعم للقيم $[0.4, 0.2]$ ليصل التمثيل البياني للزمن أعلى قيمة له والتي تجاوزت 1000_{sec} عند القيمة 0.08 للدّاعم.



شكل 2.4: النتائج التجريبية لقاعدة المعاملات الحقيقية المكثفة accidents
[أ] الزمن [ب] الذاكرة.

أمّا بالنظر إلى عامل إستهلاك الذاكرة كما يظهره الجزء [ب] فنلاحظ أنه عندما تكون قيم الدّاعم محصورة في المجال $[0.7, 0.9]$ فإنّ خوارزمية FPMax إستهلكت ما يقارب $300MB$ من الذاكرة، بينما كان إستهلاك خوارزمية Charm-MFI أقل من ذلك حيث قدّر إستهلاكها بحوالي $220MB$ ليرتفع تمثيلها البياني مباشرة بعد الخروج من ذلك المجال ويصل إلى قيمة إستهلاك مقدرة بـ $845MB$ عند القيمة التي يكون فيها الدّاعم لـ0.5، ثم ينقطع التمثيل البياني لخوارزمية Charm-MFI بعد هذه القيمة للدّاعم ويتواصل تمثيل خوارزمية FPMax بالتدريج حيث لم تنخفض سعة إستهلاكها للذاكرة عن $300MB$ إلى أن تصل إلى 0.08 حيث تجاوزت سعة إستهلاك الذاكرة حاجز $1000MB$.

• أظهر تمثيل النتائج التجريبية لقاعدة المعاملات التركيبية c73d10k كما هو موضّح في الشكل 3.4 من ناحية الزمن المستغرق الجزء [أ] من الشكل، أنّ كِلا الخوارزميتين إستغرقتا وقت التنفيذ نفسه تقريبا في المجال الذي يكون فيه الدّاعم يحمل القيمة $[0.7, 0.9]$ بزمن مقدّر بـ 2_{sec} ليرتفع التمثيل البياني لخوارزمية Charm-MFI بشكل تدريجي ثم ملحوظ مباشرة بعد القيمة 0.6 للدّاعم حيث إقترب الزمن المستغرق من حدود 2300_{sec}، بالمقابل تواصل إرتفاع تمثيل خوارزمية FPMax بشكل تدريجي أين لم يتجاوز الزمن 460_{sec} عند أقل قيمة للدّاعم وصلت إليها الخوارزمية والمقدرة بـ 0.4



شكل 3.4: النتائج التجريبية لقاعدة المعاملات التركيبية المكثفة c73d10k
[أ] الزمن [ب] الذاكرة.

أمّا من ناحية إستهلاك الذاكرة الجزء [ب] من الشكل فنلاحظ جليا التفاوت ما بين قيم الإستهلاك لكل خوارزمية، حيث بدأت خوارزمية Charm-MFI الإستهلاك لقيمة $216MB$ عند قيمة الدّاعم المساوية لـ 0.9 وواصلت الإرتفاع حتى قاربت إستهلاك $1100MB$ عند القيمة 0.5 للدّاعم بالمقابل فإنه لم يتجاوز إستهلاك خوارزمية FPMax للذاكرة $180MB$ حيث إبتدأت أصلا بقيمة إستهلاك مقدرة بـ $80MB$ عند الدّاعم 0.9 ووصلت إلى الحدود الدنيا للإستهلاك بـ $46MB$ عند قيمة الدّاعم المساوية لـ 0.4.

نلاحظ توقف كل من الخوارزميتين عند القيم المتوسطة، حيث إنقطع تمثيل خوارزمية Charm-MFI عند القيمة 0.5 للداعم، ولم يحد تمثيل خوارزمية FPMax عن القاعدة هو الآخر حيث إنقطع عند القيمة 0.4 للداعم.

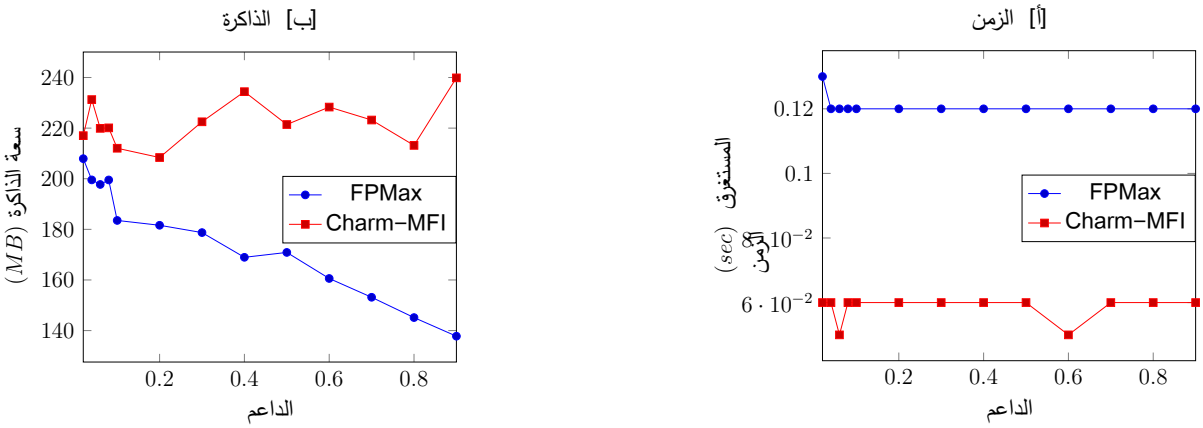
هذا الأمر يرجع إلى أنه كلما تم تخفيض قيم الدّاعم، كلما إزدادت مجموعات العناصر المتكررة القصوى المستخرجة، بالتالي يزداد إستهلاك البرنامج التنفيذي للمساحة التخزينية للذاكرة الأمر الذي يؤدّي إلى تجاوز سعة الذاكرة التخزينية.

إذن يمكن القول أن كلاً من خوارزميتي FPMax و Charm-MFI تعملان بنفس الكفاءة بالنظر إلى الزمن المستغرق للتنفيذ وهذا عند القيم الكبيرة للعتبة الدنيا للداعم، أمّا عند القيم المتوسطة والمنخفضة لعتبة الدّاعم فإنّ الزمن المستغرق للتنفيذ في خوارزمية FPMax كان أفضل من في خوارزمية Charm-MFI، بالمقابل و من ناحية سعة الذاكرة المستهلكة، فإنّ كلا الخوارزميتين أظهرتا إستهلاكا أقل نسبيا للذاكرة عند القيم المرتفعة لعتبة الدّاعم، امّا عند القيم المتوسطة والمنخفضة فيظهر جليّا تفوقُ خوارزمية FPMax كون خوارزمية Charm-MFI أظهرت إستهلاكا كبيرا للذّاكرة بداية من القيم المتوسّطة

للعتبة الدنيا للدّاعم، وبالنسبة للقيم الصغيرة لعتبة الدّاعم فلم يكتمل التمثيل البياني بسبب تجاوز سعة الذاكرة (Memory Overflow).

– قواعد المعاملات المتناثرة (Sparse):

• نلاحظ من خلال الشكل 4.4، الموضح للتمثيل البياني لنتائج التجارب على قاعدة المعاملات الحقيقة BMS WebView2 وبالنظر إلى الزمن المستغرق كما يبينه الجزء [أ] من الشكل، أنّ خوارزمية Charm-MFI إستغرقت زمنا ثابتا مقدرا بـ 0.06_{sec} في جميع قيم الدّاعم بدءا من 0.9 بل وإنخفض أحيانا إلى 0.05_{sec} عند بعض القيم للدّاعم كـ 0.6 و0.06، بينما كان الزمن المستغرق لدى خوارزمية FPMax ثابتا عند القيمة 0.12_{sec} بدءا من الداعم 0.9 ليرتفع إلى 0.13_{sec} عند القيمة 0.02.

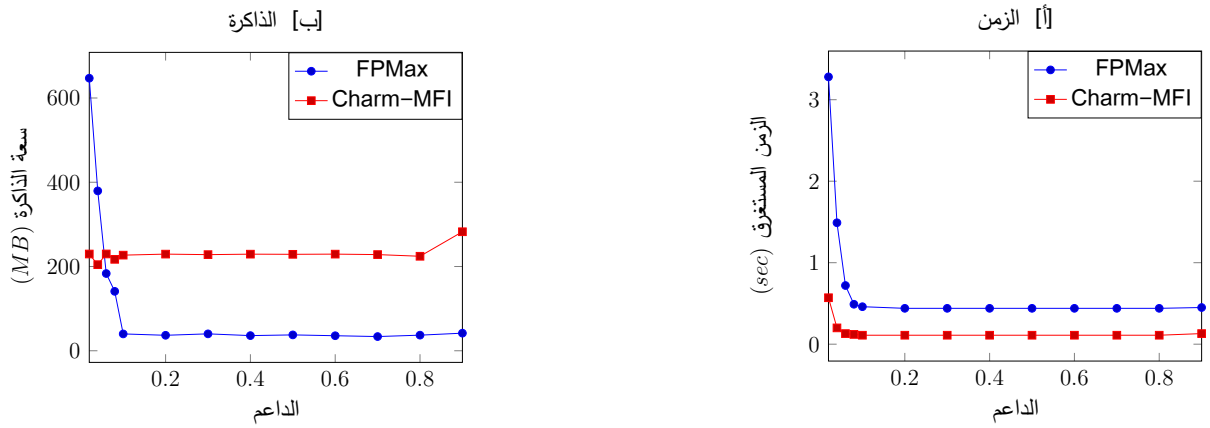


شكل 4.4: النتائج التجريبية لقاعدة المعاملات الحقيقية المتناثرة BMS Web View2.
[أ] الزمن [ب] الذاكرة

أما بالنسبة إلى سعة إستهلاك الذاكرة كما هو مبين في الجزء [ب] من الشكل فنلاحظ أن خوارزمية FPMax بدأ إستهلاكها لسعة الذاكرة من قيم يمكن إعتبارها صغيرة $137MB$ عند الدّاعم 0.9 ليرتفع تدريجيا وتصل إلى أعلى قيمة إستهلاكية لها وهي $207MB$ عند الداعم 0.02، أمّا خوارزمية Charm-MFI فكانت قيمها في تذبذب حيث إبتدأ إستهلاكها بقيم مرتفعة مقدّرة بحوالي $239MB$ عند الدّاعم 0.9 ليصل إلى $217MB$ عند الداعم المساوي لـ 0.02.

• يُظهر الشكل 5.4 التمثيل البياني الخاص بنتائج التجارب على قاعدة المعاملات التركيبية t20i6d100k من حيث الزمن الموضح من خلال الجزء [أ] أنّ خوارزمية Cahrm-MFI تميزت بتحقيق أسرع وقت في التنفيذ حيث كان ثابتا عند القيمة 0.11_{sec} في مختلف قيم الدّاعم ثم إرتفع بعدها إلى القيمة 0.20_{sec} عند قيمة الداعم 0.04 ثم

إستمرت بالإرتفاع حتى بلغت 0.57_{sec} وذلك عند قيمة الداعم المساوية لـ 0.02، أما خوارزمية FPMax فقد إستغرقت زمنا ثابتا مقدرا بـ 0.44_{sec} عبر مختلف قيم الداعم ثم إرتفع للوصول إلى أعلى قيمة 3.28_{sec} عند قيمة الداعم المساوية لـ 0.02.



شكل 5.4: النتائج التجريبية لقاعدة المعاملات التركيبية المتناثرة t20i6d100k.
[أ] الزمن [ب] الذاكرة

أمّا من ناحية سعة إستهلاك الذاكرة الجزء [ب] من الشكل فنلاحظ أن خوارزمية FPMax إستهلكت حيزا ثابتا من الذاكرة قدّر بـ $37MB$ في مختلف قيم الداعم ثم إرتفع ليصل إلى حدود $650MB$ وذلك عند القيمة 0.02 للداعم، بينما خوارزمية Charm-MFI فبدأ إستهلاكها للذاكرة من القيمة $282MB$ ثم بقي في إستقرار دون أن يتجاوز $230MB$ عبر مختلف قيم الداعم.

تبيّن التمثيلات البيانية للنتائج المتحصّل عليها أنّ خوارزمية Charm-MFI أظهرت تفوقا ملحوظا مقارنة بخوارزمية FPMax من ناحية الزمن المستغرق للتنفيذ بالنظر إلى كلا من القاعدتين سواءا الحقيقة أو التركيبية، أمّا فيما يخصّ سعة الذاكرة فيلاحظ جليّا الإستهلاك الكبير للذاكرة من قبل خوارزمية Charm-MFI.

بالتالي، يمكن القول من خلال هذه النتائج أنّ خوارزمية FPMax تفوقت على خوارزمية Charm-MFI في قواعد المعاملات المكثفة (Condensed Bases) حيث كانت أقل من حيث الزمن المستغرق للتنفيذ وذات إستهلاك أقل للمساحة التخزينية للذاكرة، أمّا بالنسبة لقواعد المعاملات المتناثرة (Sparse Bases) فإنّ خوارزمية Charm-MFI أظهرت تفوقا من حيث الزمن المستغرق للتنفيذ مقارنة بخوارزمية FPMax، ولكن بالنظر للمساحة التخزينية للذاكرة فإن خوارزمية Charm-MFI إستهلكت الكثير من سعة الذاكرة عكس خوارزمية FPMax التي كان إستهلاكها للذاكرة مقبولا.

لذا فإنّ تحديد الخوارزمية الأمثل (Optimal Algorithm) يعود حقّا لأوليات ومتطلبات العمل المراد إنجازه.

6.4 خاتمة

تم في هذا الفصل تطبيق دراسة تجريبية تمثلت في عقد مقارنة بين خوارزميتي إيجاد مجموعات العناصر المتكررة القصوى وهما خوارزمية FPmax وخوارزمية Charm-MFI المتواجدتين في منصة SPMF وإقتصرت التجارب حولهما فقط نظرا لعدم وجود برامج تنفيذية (Implements) لخوارزميات أخرى تُعنى بإيجاد مجموعات العناصر المتكررة القصوى في هذه المنصة.

شملت المقارنة مجموعة مختلفة من قواعد المعاملات: حقيقة وتركيبية / مكثفة ومتناثرة. وإختصت حول عاملين هما: الزمن المستغرق والمساحة التخزينية للذاكرة.

حيث أظهرت النتائج التجريبية تفوق خوارزمية FPMax على خوارزمية Charm-MFI من حيث الإستهلاك الأقل للمساحة التخزينية للذاكرة، أمّا بالنسبة للزمن المستغرق للتنفيذ فإنّ خوارزمية Charm-MFI تفوقت في قواعد المعاملات المتناثرة بالمقابل رجحت كفة خوارزمية FPMax في قواعد المعاملات المكثفة.

خاتمة

أصبحت البيانات تتزايد بشكل مستمر وملحوظ نتيجة التطور التكنولوجي الذي يشهده العالم حاليا، بل وشملت هذه البيانات مختلف المجالات والميادين، لذا أصبح من الضروري إستغلالها للوصول إلى المعارف المُكتنزَة داخلها.

لكن لم تتمكن الأساليب والوسائل التقليدية من تحليل هذه البيانات بسبب إختلافها وتنوعها، مما إستدعى إستخدام طرق جديدة لإستخلاص المعارف من البيانات، وأبرز هذه الطرق هي إستعمال تقنيات "التنقيب في البيانات" (Data Mining) بمختلف أدواتها ومهامها، حيث تركّزت الدراسة على واحدة من أهمّ هذه المهام وهي ما تعرف بقواعد الإرتباط (Association Rules) والتي يتمثل عملها في إيجاد علاقات بين متغيرين أو أكثر أو بمفهوم أوضح إستخراج قواعد الإرتباط من داخل هذه البيانات، لكن بصورة أعمق فلقد تمّ التركيز أكثر على أهمّ خطوة للقيام بهذه العملية والتي تتمثل في إستخراج مجموعات العناصر المتكررة (Mining Frequent Itemsets).

أولى الأفكار المطروحة كان عملية إستخراج العناصر المتكررة الكلية (All Frequent Itemsets) الموجودة في قاعدة البيانات من خلال مقاربات وخوارزميات تمّ إقتراحها من قبل مجموعة من الباحثين على مدى عقدين من الزمن، فظهرت خوارزميات المقاربة بالمستويات (Level-wise Approach) والمقاربة العمودية (Vertical Approach) بالإضافة إلى خوارزميات المقاربة بالإسقاط (Projection Approach)، لكن ومع مرور الوقت ظهر نهج جديد بحث في فكرة إستنباط قواعد الإرتباط من حيز صغير لمجموعات العناصر المتكررة، أي ما يُعرف بالتمثيلات المتراصة (Condensed Representations) لمجموعات العناصر المتكررة، والتي من ضمنها تمثيلات مجموعات العناصر المتكررة المغلقة (CFI) إضافة إلى تمثيلات مجموعات العناصر المتكررة القصوى (MFI)، جاءت هذه الفكرة من أجل تفادي التعداد الكلي لمجموعات العناصر بالتالي التقليل من تكلفة إستهلاك المساحة التخزينية للذاكرة والوقت المستغرق عند التنفيذ.

تمّ تسليط الضوء في هذه المذكرة على تمثيلات مجموعات العناصر المتكررة القصوى بإلقاء نظرة موجزة على بعض الخوارزميات الحالية الخاصّة بإستخلاص مجموعات العناصر المتكررة القصوى (MFI) بداية مع خوارزمية MaxMiner، وخوارزمية Depth-Project، فخوارزمية Mafia، ثم خوارزمية FPMax، بعدها خوارزمية LCM-Max،

إضافة إلى خوارزمية GenMax، لتختم في النهاية بخوارزمية Charm-MFI.

بعدها تمت عملية مقارنة الخوارزميتين المتواجدتين في المنصة مفتوحة المصدر SPMF وهما خوارزمية Charm-MFI وخوارزمية FPmax، تمت المقارنة على أساس عاملي الزمن المستغرق للتنفيذ وسعة إستهلاك الذاكرة لكل منهما من خلال مجموعات مختلفة من البيانات المختارة عشوائيا.

حيث ومن خلال هذه النتائج تم التوصل إلى أنّ خوارزمية FPMax فاقت خوارزمية Charm-MFI في قواعد المعاملات المكثفة نظرا إلى أنها كانت أقل من حيث الزمن المستغرق للتنفيذ وذات إستهلاك أقل للذاكرة، بالمقابل وعلى مستوى قواعد المعاملات المتناثرة فإنّ خوارزمية Charm-MFI أظهرت تفوقا من حيث الزمن المستغرق للتنفيذ مقارنة بخوارزمية FPMax إلا أن إستهلاكها للذاكرة كان أكبر منه في خوارزمية FPMax.

تعرفنا من خلال إنجاز هذه المذكرة على مجال جديد في علوم الكومبيوتر وهو مجال "إستخلاص المعارف من قواعد البيانات" وبخاصة إستخراج مجموعات العناصر المتكررة القصوى التي تعتبر من الخطوات المهمة لإنشاء قواعد الإرتباط.

من بين الصعوبات التي واجهتنا في هذه المذكرة ندرة البرامج التنفيذية (Implementations) للخوارزميات المتعلقة بإستخراج مجموعات العناصر المتكررة القصوى بشكل عام وفي منصة SPMF بشكل خاص ممّا أدّى إلى حصر التطبيقات التجريبية على خوارزميتين فقط.

نقترح تدعيم هذه المنصة بخوارزميات إستخراج العناصر المتكررة القصوى، من أجل أن تكون الدراسات التطبيقية المستقبلية أكثر تعمقًا وشمولية، إضافة إلى دمج هذه الخوارزميات في تطبيقات عملية على أرض الواقع.

كما نقترح البحث في إمكانية تقديم تمثيل لمجموعات العناصر المتكررة القصوى مع حفظ معلومات الداعم الخاص بكل مجموعة عناصر متكررة.

المراجع العلمية

[1] Ramesh C. Agarwal, Charu C. Aggarwal, and V. V. V. Prasad. Depth first generation of long patterns. In KDD, pages 108–118. ACM, 2000.

[2] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In VLDB, pages 487–499. Morgan Kaufmann, 1994.

[3] Max Boisot and Agustí Canals. Data, information and knowledge: have we got it right? Journal of evolutionary economics, 14(1): 43–67, 2004.

[4] Douglas Burdick, Manuel Calimlim, and Johannes Gehrke. MAFIA: A maximal frequent itemset algorithm for transactional databases. In ICDE, pages 443–452. IEEE Computer Society, 2001.

[5] Usama M. Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery in databases. AI Magazine, 17(3):37–54, 1996.

[6] Philippe Fournier-Viger, Jerry Chun-Wei Lin, Antonio Gomariz, Ted Gueniche, Azadeh Soltani, Zhihong Deng, and Hoang Thanh Lam. The SPMF open-source data mining library version 2. In ECML/PKDD (3), volume 9853 of Lecture Notes in Computer Science, pages 36–40. Springer, 2016.

[7] Karam Gouda and Mohammed Javeed Zaki. Genmax: An efficient algorithm for mining maximal frequent itemsets. Data Min. Knowl. Discov., 11(3):223–242, 2005.

[8] Gösta Grahne and Jianfei Zhu. High performance mining of maximal frequent itemsets. In HPDM, 6th International Workshop on High Performance Data Mining, 2003.

[9] Jiawei Han, Jian Pei, and Yiwen Yin. Mining frequent patterns without candidate generation. In SIGMOD Conference, pages 1–12. ACM, 2000.

[10] Jiawei Han, Micheline Kamber, and Jian Pei. Data Mining: Concepts and Techniques, 3rd edition. Morgan Kaufmann, 2011.

[11] Roberto J. Bayardo Jr. Efficiently mining long patterns from databases. In SIGMOD Conference, pages 85–93. ACM Press, 1998.

[12] khaled tannir. blog.khaledtannir.net l'algorithme fp-growth. URL http://blog.khaledtannir.net/2012/07/lalgorithmefp-growth-les-bases-13/.

[13] Daniel T. Larose and Thierry Vallaud. Des données à la connaissance. Bases de données. Vuibert informatique, Paris, impr. 2005.

[14] Nicolas Pasquier. Data Mining : algorithmes d'extraction et de réduction des règles d'association dans les bases de données. PhD thesis, Blaise Pascal University, Clermont-Ferrand, France, 2000.

[15] Philippe PREUX. Fouille de données notes de cours. 2011.

[16] Laszlo Szathmary. Symbolic Data Mining Methods with the Coron Platform. (Méthodes symboliques de fouille de données avec la plate-forme Coron). PhD thesis, Henri Poincaré University, Nancy, France, 2006. URL https://tel.archives-ouvertes.fr/tel-00336374.

[17] Takeaki Uno, Tatsuya Asai, Yuzo Uchida, and Hiroki Arimura. An efficient algorithm for enumerating closed patterns in transaction databases. In Discovery Science, volume 3245 of Lecture Notes in Computer Science, pages 16–31. Springer, 2004.

[18] Guizhen Yang. The complexity of mining maximal frequent itemsets and maximal frequent patterns. In KDD, pages 344–353. ACM, 2004.

[19] Fu Yongjian. Data mining: tasks, techniques and applications. IEEE Potentials, 16(4):18–20, 1997.

[20] Mohammed J. Zaki and Wagner Meira Jr. Data Mining and Analysis: Fundamental Concepts and Algorithms. Cambridge University Press, 2014.

[21] Mohammed Javeed Zaki and Karam Gouda. Fast vertical mining using diffsets. In KDD, pages 326–335. ACM, 2003.

[22] Mohammed Javeed Zaki, Srinivasan Parthasarathy, Mitsunori Ogihara, and Wei Li. New algorithms for fast discovery of association rules. In KDD, pages 283–286. AAAI Press, 1997.