تحسين خوارزمية Apriori من خلال تقليل مسح قاعدة البيانات

الباحث: د. محمد حجوز

المدرس في كلية العلوم الثانية بجامعة البعث

M.Hajjouz@gmail.com

0955552683, 0312121979

الملخص:

تعتمد خوارزمية قواعد الارتباط في تقنية التنقيب في البيانات على إيجاد علاقات مثيرة للاهتمام وارتباطات ملحوظة بين العناصر المختلفة في مجموعة كبيرة من البيانات. ومن الأمثلة الشهيرة عن التنقيب في البيانات باستخدام تقنية قواعد الارتباط هو تحليل سلة السوق. حيث تعتبر خوارزمية Apriori هي أول خوارزميات قواعد الارتباط. إلا أن هذه الخوارزمية تحتوي على مشكلتين في عملية التنقيب في البيانات: الأولى هي أنها تحتاج إلى الكثير من الوقت لمسح كامل قاعدة البيانات والثانية هي أنه ينتج عددًا كبيرًا من المجموعات المرشحة غير ذات الصلة والتي تشغل ذاكرة النظام. نقدم في هذا البحث خوارزمية محسنة عن هذه الخوارزمية لحل هاتين المشكلتين. ستعمل خوارزمية Apriori المحسّنة على تقليل عدد مرات مسح قاعدة بيانات بالكامل من خلال تقليل التوليد الزائد للعناصر الفرعية حتى نحصل في الأخير على واحدة من مجموعات العناصر المرشحة توافق الدعم الأدنى minsup. لتحقيق هذه الأهداف سنستخدم مفهوم المجموعة الرئيسة العامة وأمثلة قواعد البيانات. تعمل خوارزمية Apriori المحسّنة على تقليل استهلاك موارد النظام وتحسين كفائته.

الكلمات المفتاحية: استخراج البيانات، المجموعة الرئيسة العامة، المجموعة الرئيسة المحلية، خوارزمية Apriori، مجموعات العناصر التكرارية.

Enhanced of the Apriori algorithm by reducing database scanning

Abstract:

Data mining with association rules technique is the process of finding interesting relationships and remarkable correlations between different elements in a large set of data elements. A famous example of data capturing using correlation rules technique is market basket analysis. The Apriori algorithm is the first association rule algorithm، However, this classic algorithm has two problems with data mining. The first is that it will take a lot of time to scan the database, and the second is that it will generate a large number of irrelevant candidate groups that occupy the system memory. In this paper, we present an improved algorithm to solve these two problems. The improved Apriori algorithm will reduce the number of times to scan an entire database by reducing the over generation of sub-items until we finally have one of the minimum support matching candidate item sets. To achieve these goals, we will use the general energy concept and database examples. The improved Apriori algorithm reduces system resources occupied and improves system efficiency

Key words: Data mining, Global power set, Local power set, Apriori algorithm, Frequent itemsets.

1. المقدمة

يسمى عصرنا الحالي بعصر الانفجار المعلوماتي لما يحتوي من بيانات ضخمة في جميع المجالات والتي تزداد يوماً بعد يوم، وينتج عنه ما يسمى بقواعد البيانات الضخمة، وأصبحت الحاجة ملحة لاستخراج معلومات مخفية وذات مغزى من هذه القواعد. ومن المستحيل العثور على معلومات مفيدة ومخفية بالطرق التقليدية. مما أدى إلى ظهور تقنيات التنقيب في البيانات وما لديها من قواعد ارتباطات وعلاقات ارتباط مثيرة للاهتمام بين مجموعة كبيرة من عناصر البيانات. تعرض قواعد الارتباط شروط قيمة السمات التي تحدث معاً بشكل متكرر في مجموعة بيانات معينة. توفر قواعد الارتباط معلومات من هذا النوع في شكل عبارات "if–then". يتم حساب هذه القواعد من البيانات، وعلى عكس قواعد المنطق الشرطية، فإن قواعد الارتباط هي احتمالية بطبيعتها. وتحتوي على رقمين يعبران عن درجة عدم اليقين بشأن القاعدة. ويستخدم في قواعد الارتباط عاملين أساسيين هما:[1]

- **الدعم:** وهو عدد المعاملات التي تشمل جميع العناصر في الأجزاء السابقة واللاحقة من القاعدة. (يتم التعبير عنه أحيانًا كنسبة مئوية من إجمالي عدد السجلات في قاعدة البيانات)
- **الثقة:** وهي نسبة عدد المعاملات التي تشمل جميع العناصر اللاحقة والسابقة (أي الدعم) إلى عدد المعاملات التي تشمل جميع العناصر السابقة.

2. مشكلة البحث وأهميته:

نسعى في هذا البحث إلى تحسين خوارزمية Apriori وذلك بتخفيض عدد مرات مسح كامل قاعدة البيانات من أجل الحصول على المجموعات التكرارية وذلك من خلال تجاهل المجموعات المرشحة غير ذات الصلة من المجموعات التكرارية والتي تشغل ذاكرة النظام وتزيد مساحة التخزين.

3. خوارزمية Apriori العادية:[1]

تفيد خوارزمية Apriori العادية من خلال استخدام قواعد الارتباط في تحديد العناصر المرتبطة مع بعضها، فمثلاً في المتاجر الكبيرة (Supermarkets) بأمريكا يلجأ أصحابها إلى دراسة تحليل السوق والسلل الشرائية لمعرفة البضائع التي تشترى مع بعضها لوضعها أمام أنظار الزبون بجانب بعضها لكي لا ينسى أي غرضٍ منها وحتى يكسبه المتجر كعميل مخلص، وتعتمد هذه الخوارزمية على حساب عاملين أساسيين هما: عامل الدعم Support ويحسب عن طريق الاحتمالات، وعامل الثقة Confidence. يقوم الخبير في المؤسسة بوضع عتبة الدعم الصغرى Minimum Support Threshold، وعتبة الثقة الصغرى Minimum Confidence Threshold وكل من يتجاوز هاتين العتبتين يكون دعمه وثقته كبيرين ومن ثم تكون القاعدة قوية. فيما يلي مثال عن قاعدتين:[2]

- Buys(x, “Milk”) ⇨ Buys(x, “Bread”) [0.5%, 60%]
- Student(x, “CS”) ^ takes(x, “DB”) ⇨ grade(x, “A”) [1%, 75%]

تفيد القاعدة الأولى في معرفة هل الزبون x الذي اشترى الحليب سيشتري الخبز معه؟ وتمثل القيمة 0.5% عامل الدعم، والقيمة 60% عامل الثقة. وتكون قوة القاعدة 60%. تفيد القاعدة الثانية في معرفة هل الطالب x الذي سجّل على مقرر مهارات الحاسوب CS ومقرر قواعد البيانات DB سيحصل على التقدير "A". تملك هذه القاعدة عامل دعم بنسبة 1% وعامل ثقة بنسبة 75%. أي 75 بالمئة من الطلاب حققوا هذ التقدير لهذا تُعدُّ هذه القاعدة قاعدة قوية نوعاً ما. يستفيد المشرف الأكاديمي من هذه المعرفة في اتخاذ قرار بتوجيه الطلاب على هذه القاعدة وكذلك الأمر بالنسبة لبقية المقررات.

لتوضيح عمل الخوارزمية، سنأخذ خمسة سجلات من المشتريات التي يقوم بها الزبائن لستة أغراض كما في الجدول(1):

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

الجدول(1) سجلات المشتريات لستة أغراض يقوم بها خمسة زبائن

ونريد التحقق من القاعدة التي تقول: هل كل الزبائن الذين اشتروا الحليب Milk وحفائض الأطفال Diaper سيشترون البيرا Beer:

$$\{\text{Milk, Diaper}\} \Rightarrow \text{Beer}$$

للتحقق من ذلك، يجب حساب عامل الدعم S وذلك بتقسيم عدد الزبائن الذين اشتروا الأغراض الثلاثة Milk, Diaper, Beer مع بعضهم وعددهم 2 على عدد الزبائن الكلي |T| وهو 5 كالآتي.

$$s = \frac{\sigma(\text{Milk, Diaper, Beer})}{|\text{T}|} = \frac{2}{5} = 0.4$$

وحساب عامل الثقة C، وذلك بتقسيم عدد الأشخاص الذين اشتروا الأغراض الثلاثة Milk, Diaper, Beer مع بعضهم وعددهم 2 على عدد الأشخاص الذين اشتروا Milk, Diaper مع بعضهم مع أو بدون أغراض أخرى وهم 3.

$$c = \frac{\sigma(\text{Milk, Diaper, Beer})}{\sigma(\text{Milk, Diaper})} = \frac{2}{3} = 0.67$$

لفهم عمل هذه الخوارزمية، سنُعرف بعض المصطلحات الآتية:

- **K−Itemset** مجموعة الأغراض المشتراة مع بعضها وعددها K، وهي في

مثالنا هذا تمثل الأغراض {Milk, Bread, Diaper}.

- **Support count** عدد السلل الشرائية نرمز له (σ)، ويمثل في مثالنا هذا عدد السلل التي تم فيها شراء الأغراض مع بعضها σ({Milk, Bread, Diaper})=2.
- **Support** عامل الدعم وينتج من تقسيم عدد السلل الشرائية للأغراض التي تُشترى مع بعضها على عدد الزبائن الكلي s({Milk, Bread, Diaper})=2/5
- **Frequent Itemset** المجموعات التكرارية التي يكون عامل الدعم فيها أكبر أو يساوي عتبة الدعم الصغرى minsup التي يضعها الخبير.

يتم في هذه الخوارزمية مقارنة عامل الدعم وعامل الثقة مع عتبة الدعم الصغرى minsup وعتبة الثقة الصغرى minconf التي يضعها الخبير في المؤسسة لكل قواعد الارتباط المستنتجة، ويجب أن يحققا ما يلي:

1- أن يكون عامل الدعم أكبر أو يساوي عتبة الدعم الصغرى التي يضعها الخبير أي:

$$\text{Support} \geq \text{minsup threshold}$$

2- أن يكون عامل الثقة أكبر أو يساوي عتبة الثقة الصغرى التي يضعها الخبير أي:

$$\text{Confidence} \geq \text{minconf threshold}$$

يتم في خوارزمية قواعد الارتباط مراعاة الأمور الآتية:

1- إيجاد كل قواعد الارتباط الممكنة.

2- حساب عامل الدعم والثقة لكل قاعدة.

3- إهمال القواعد التي يكون معدل الدعم فيها أقل من عتبة minsup وعامل الثقة

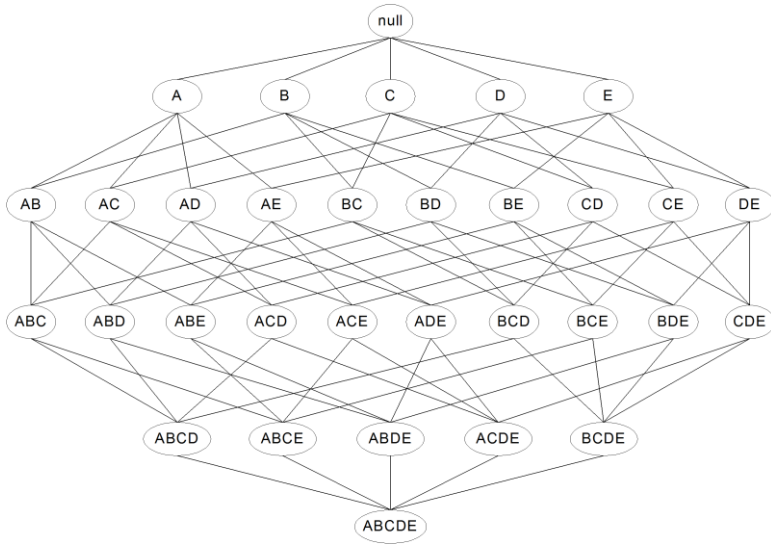
أقل من عتبة minconf.

من الجدول (1) تم استنتاج جميع قواعد الارتباط الممكنة من البيانات المتوفرة وفق الاتي:

- {Milk,Diaper} $\rightarrow$ {Beer} (s=0.4, c=0.67)
- {Milk,Beer} $\rightarrow$ {Diaper} (s=0.4, c=1.0)
- {Diaper,Beer} $\rightarrow$ {Milk} (s=0.4, c=0.67)
- {Beer} $\rightarrow$ {Milk,Diaper} (s=0.4, c=0.67)
- {Diaper} $\rightarrow$ {Milk,Beer} (s=0.4, c=0.5)
- {Milk} $\rightarrow$ {Diaper,Beer} (s=0.4, c=0.5)

يُحسب عامل الدعم والثقة في هذه القواعد وفق الآتي:

في القاعدة الأولى: يُحسب عامل الدعم بتقسيم عدد التكرارات التي يتم فيها شراء مجموعة الأغراض {Milk, Diaper, Beer} (أي عدد الزبائن الذين اشتروا هذه الأغراض مع بعضها) وهم 2 (السجلان الثالث والرابع في الجدول (1)) على عدد الزبائن الكلي وهو 5 بالتالي s=2/5=0.4. ويُحسب عامل الثقة بتقسيم عدد تكرارات المجموعة نفسها وهو 2 على مجموع الزبائن الذين اشتروا أغراض من ضمنها يجب أن تكون المجموعة {Milk, Diaper} وهم 3 (السجلات 3 و4 و5 في الجدول (1)) بالتالي c=2/3=0.67. وهكذا بالنسبة لبقية القواعد.



الشّكل(1) المجموعات المُولدة من خمسة أغراض.

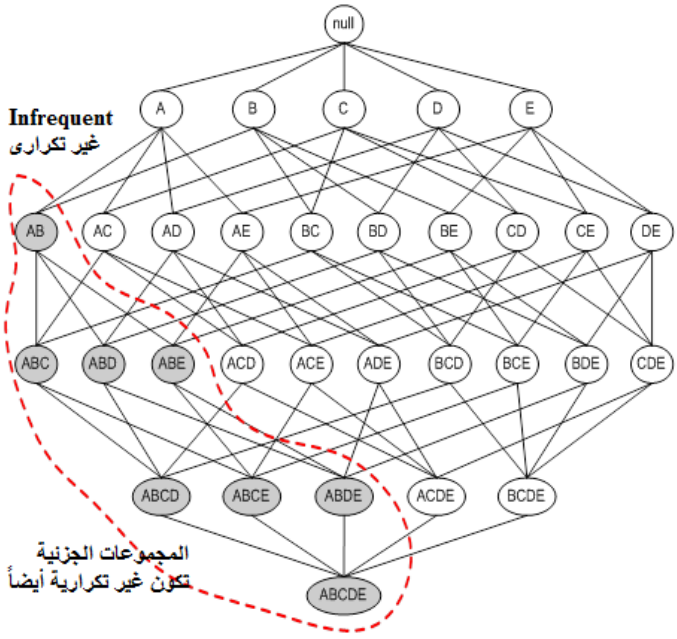
نلاحظ أن عامل الدعم في القواعد السابقة لمجموعة البيانات نفسها مع اختلاف القاعدة متساوٍ وهو 0.4، أما عامل الثقة فيختلف من قاعدة لأخرى وسنوضح السبب لاحقاً. [3]

يتم توليد المجموعات التكرارية لخمسة أغراض A, B, C, D, E على سبيل المثال، كما في الشّكل (1) ونلاحظ أن عدد المجموعات ينتج من $2^{(\text{عدد الأغراض})}$ أي $2^5=32$ مجموعة مرشّحة بما فيها عند عدم شراء أي غرض Null. وهذا العدد سيزداد كثيراً عند وجود عدد كبير من الأغراض.

الآن نريد تحديد المجموعات التكرارية المرشّحة Candidates وغير التكرارية من بين هذه المجموعات لتحديد أي المجموعات الأكثر شِراءً باستخدام خوارزمية Aprori التي تعمل وفق المبدأ الآتي:

- عند وجود مجموعة itemset تكرارية فإن جميع المجموعات الجزئية التابعة لها

يجب أن تكون تكرارية أيضاً. والعكس صحيح أي عند وجود مجموعة غير تكرارية لا داعي لتوليد المجموعات الجزئية التابعة لها لأنها لـن تكـون تكرارية أيضاً. إذا اعتبرنا من الشكّل(2) المجموعة AB غير تكرارية أي حسبنا عامل الدعم لها فوجدناه أقل من minsup الذي يضعه الخبير، ومن ثم فإن جميع المجموعات التابعة لها في مستوى الأبناء والأحفاد والأدنى منها تكون غير تكرارية أيضا لهذا تُحذف.



الشّكل(2) استثناء المجموعات الجزئية التي تكون مجموعتها الأساسية غير تكرارية

<u>مثال تطبيقي عن عمل خوارزمية Apriori العادية:</u>

بفرض أنه يوجد خمسة زبائن يريدون شراء مجموعة من الأغراض مؤلفة من ستة أشياء هي: الخبز والحليب وحفائض الأطفال والكولا والبيرا والبيض {Bread, Milk, Diaper, Coke, Beer, Eggs}، والمطلوب معرفة الثلاثيات (الأغراض الثلاثة) التي تُشترى مع بعضها؟. أعطى الخبير الحد الأدنى للدعم Minimum Support بمقدار

3، وهؤلاء الزبائن الخمسة يمكن أن يشتروا مجموعة من هذه الأغراض أو لا يشتروا أي شيء. الآن لنبحث عن المجموعات التكرارية باستخدام خوارزمية Apriori. يتم تطبيق الخوارزمية على الجدول(1)، ونلاحظ من خلاله أن الزبون الأول اشترى خبزاً وحليباً والثاني اشترى خبزاً وحفائض وبيرا وبيضاً وهكذا لبقية الزبائن. بمسح كامل لقاعدة البيانات يمكن معرفة عدد المرات التي أُشتري فيها كل غرض من الأغراض كما هو موضحٌ في الجدول (2): [4]

Item	Count
Bread	4
Coke	2
Milk	4
Beer	3
Diaper	4
Eggs	1

الجدول (2) عدد مشتريات كل غرض

نلاحظ من الجدول(2)، أن عدد مشتريات الكولا Coke بلغت 2 ومشتريات البيض Eggs بلغت 1، وهي أقل من minsup=3. إذن فإن كل المبيعات المرشحة الثنائية 2–itemsets Candidates اللاحقة لا يتم توليدها باستخدام الكولا والبيض (يشير العدد 2 في 2–itemsets إلى المجموعات الثنائية)، من ثم ستولّد هذه الثنائيات المرشحة من 4 أغراض فيكون عددها 6 ثنائيات مرشحة. يوضح الجدول(3) هذه الثنائيات. يحتوي العمود Count عدد تكرار مجموعات المشتريات، فمثلاً المجموعة الثنائية {Bread, Milk} تكرر شراؤها من قبل الزبون الأول والرابع والخامس 3 مرات وهكذا بالنسبة لبقية المجموعات كما في الجدول(3). ويبقى لدينا 4 ثنائيات مرشحة وهي: {{Milk,Diaper}, {Bread,Diaper}, {Bread,Milk}, {Beer,Diaper}}.

Itemset	Count
{Bread,Milk}	3
{Bread,Beer}	2
{Bread,Diaper}	3
{Milk,Beer}	2
{Milk,Diaper}	3
{Beer,Diaper}	3

الجدول (3) توليد المرشحات الثنائية

ستولّد ثلاثيات مرشحة 3–itemset من هذه الثنائيات الأربعة. لاحظ أنه لا يوجد سوى ثلاثية واحدة مرشحة يمكن تشكيلها وهي {Bread,Milk,Diaper} وقد تكررت مرتين في الجدول(4)، وما تبقى من الثلاثيات ليست تكرارية لأن عامل الدعم فيها أقل من عتبة الدعم الصغرى، كما موضح في الجدول(4):

Itemset	Count
{Bread,Milk,Diaper}	2

الجدول(4) توليد المرشحات الثلاثية

نلاحظ أن الخوارزمية Apriori قلّلت من عدد المجموعات المرشحة نوعاً ما، حيث بلغ في المثال السابق 13 مجموعة وهي ناتجة عن جمع {13=1+6+6}، إذ تمثل 6 الأولى مجموع المرشّحات في الجدول(2) وتمثل 6 الثانية مجموع المرشّحات في الجدول(3) ويمثل الواحد مجموع المرشّحات في الجدول(4). إذاً قللت هذه الخوارزمية عدد المرشحات كثيراً فبدلاً من أن تكون 2^6=64 أصبحت 13 مجموعة مرشحة.

في نهاية الخوارزمية يُحسب عامل الثقة للمجموعة الأخيرة حسب القاعدة {Bread,Milk}=>Diaper، فنجد أن عدد الأشخاص الذين اشتروا الأغراض {Bread,Milk} مع بعضهم 3 من ثم يكون عامل الثقة c=2/3=0.67 أي احتمال

شراء الحفائض عند شراء كل من الحليب والخبز بلغ 67% أما العكس: أي ما هو عامل الثقة للزبائن الذين اشتروا الحفائض أولاً ثم اشتروا الخبز والحليب Diaper=>{Bread,Milk}. بالحساب نستنتج أن عامل الثقة هو c=2/4=50% حيث تمثل القيمة 4 عدد الزبائن الذين اشتروا الحفائض وهو مختلف عن السابق. كما نلاحظ أن عامل الثقة لم يكن كبيراً بشكل كافٍ لاتخاذ قرار بوضع الثلاثية {Bread,Milk,Diaper} مع بعضها بعضاً على الرغم من أنه مقبول.

خطوات خوارزمية Apriori العادية:

تعمل خوارزمية Apriori على مبدأ توليد عدد K من المرشحات ثم ربطها مع نفسها لتشكيل L_{k-1} من التكرارات. والاعتماد على المبدأ الأساسي لهذه الخوارزمية الذي يقول: إن أية مجموعة مرشحة itemset غير تكرارية فإن كل المجموعات الجزئية الناتجة عنها تكون غير تكرارية.

الخطوات:

C_k: المجموعة المرشحة من المجموعات K.

L_k: المجموعة التكرارية من المجموعات K.

$L1$={الأغراض التكرارية}

for (k = 1; L_k !=∅; k++) **do**

{

$C_{k+1}=L_k$ المرشحات من المجموعات التكرارية

for each مناقلة t في قاعدة البيانات (D) do

زيادة عدد كل المرشحات في C_{k+1} المحتواة في t

L_{k+1}=min_support بأقل دعم C_{k+1} المرشحات في

}

Return $\cup_k L_k$

رغم أن هذه الخوارزمية خففت عدد المرشحات لكن بقي تعقيدها عالياً نسبياً بسبب وجود الأُس 2^d، لأنها تقوم بمسح كامل قاعدة البيانات في كل مرة يتم فيها توليد المرشحات مما يُشغل نظام الذاكرة وتتطلب زمن طويل لتنفيذها.[5]

4. الدراسات المرجعية:

- قام الباحث يوبو جيا yubo jia [6] بتحسين خوارزمية Apriori المحسنة اعتماداً على تقسيم البيانات وعدد مجموعات العناصر الديناميكية، حيث قام أولاً بتقسيم قاعدة بيانات المعاملات D إلى n جزء لا تتقاطع مع بعضها البعض، وإذا كان الحد الأدنى للدعم هو minsup لقاعدة البيانات D، فسيكون الحد الأدنى للدعم لكل جزء من قاعدة البيانات هو (minsup * num_of_transaction_of_partition). تعمل هذه الخوارزمية أولاً على مسح قاعدة البيانات والبحث عن جميع المجموعات التكرارية لكل قسم. يطلق عليه المجموعات المحلية التكرارية. سيتم استخدام بنية بيانات خاصة للاحتفاظ بهذه المجموعات التكرارية المحلية، ويتم استخدام TID الخاص بسجل المعاملات لتتبع هذه المجموعات التكرارية المحلية. يمكن العثور على جميع مجموعات عناصر k المحلية التكرارية بمسح واحد لقاعدة البيانات k=1,2,... ثم يتم جمع كل المجموعات المحلية التكرارية لإنشاء مجموعات العناصر المرشحة للمجموعات التكرارية في قاعدة البيانات بأكملها D. ثم مسح قاعدة البيانات مرة أخرى للحصول على minsup لجميع مجموعات العناصر المرشحة، ثم في النهاية يتم تحديد مجموعات العناصر التكرارية العامة، إلا أن هذه الخوارزمية تكرر عملية مسح قاعدة البيانات لمرات عديدة.
- أما الباحث روي تشانغ Rui Chang [7] فقدم خوارزمية تحسين جديدة تسمى Apriori–Improvement تستخدم هذه الخوارزمية بنية من النوع المختلط، يتم فيها

تخزين العلامات TID وعناصر مكملة للعناصر الأصلية إذا كانت العلامة صحيحة. ويوفر هذا الاستخدام مساحة التخزين، إلا أنه بقي تعقيد الخوارزمية عالٍ.

- استخدم الباحث شيلا أ. عباية Sheila A. Abaya [8] الحجم المحدد وتردد المقاس المحدد للسمة لتحسين أداء خوارزمية Apriori. يشير الحجم المحدد إلى عدد العناصر لكل معاملة ويشير تردد الحجم المحدد إلى عدد المعاملات التي تحتوي على عناصر حجم معين على الأقل. يتم في هذه الخوارزمية أولاً حساب الحجم المحدد لكل المعاملات ثم عدد تردد الحجم المحدد. وبعدها يتم تحديد جميع المعاملات التي لديها الحد الأدنى لتكرار الحجم المحدد. ثم يتم إنشاء مجموعات ذات حجم معين. المجموعات ذات التردد الأقل من الحد الأدنى من الدعم minsup يتم إزالتها. وتستمر هذه العملية حتى يتم العثور على مجموعة العناصر النهائية. إلا أن هذه الخوارزمية ستكون بطيئة إذا لم يتم تصميم وظيفة الدمج بشكل جيد.

5. خوارزمية Apriori المحسّنة:

1.5 خطوات تحسين خوارزمية Apriori:

من أجل تعزيز كفاءة إنتاج مجموعات العناصر التكرارية في خوارزمية Apriori، نقوم بحل مشكلتين في هذه الخوارزمية. الأولى: هي أنه يتطلب مسح قاعدة البيانات عدة مرات، والثانية: هي توليد مجموعات عناصر مرشحة كبيرة، مما يزيد من تعقيد الخوارزمية. للتغلب على هاتين المشكلتين نقوم أولاً بمسح قاعدة البيانات بحثًا عن مجموعة العناصر التكرارية الأولى، ثم توليد مجموعة رئيسة واحدة ونضع عندها عداد مجموعة العناصر يساوي الصفر Count=0 ونسمي هذه المجموعة بالمجموعة الرئيسية العامة.

عندما نقوم بفحص قاعدة البيانات بحثاً عن عدد مجموعة العناصر، نقوم أولاً بحذف العناصر غير الموجودة في قائمة مجموعة العناصر التكرارية الأولى من

المعاملة، ستعمل هذه الخطوة على تقليل التوليد الإضافي لمجموعات العناصر المرشحة. وبعد عملية الحذف نقوم بإنشاء مجموعة رئيسة محلية من العناصر المتبقية من المعاملة ومقارنتها مع المجموعة الرئيسة العامة، وعند وجود زيادة نضيف لعدد مجموعة العناصر واحد. ستعمل هذه الخطوة على تقليل المسح المتعدد لقاعدة البيانات بالتالي تزداد كفاءة الخوارزمية.

2.5 خطوات تحسين خوارزمية Apriori:

- الإدخالات (Input):

1. قاعدة البيانات D بالتنسيق (Tid ، مجموعة العناصر) ، حيث يكون Tid هو معرف العمل ومجموعة العناصر هي مجموعة عناصر العمل.
2. الحد الأدنى للدعم minsup.

- المخرجات (Output): Li مجموعة العناصر التكرارية في قادة البيانات D:

وفيما يلي خطوات الخوارزمية:

1) ضع L1 يساوي مجموعة العناصر التكرارية الأولى في قاعدة البيانات (D).
2) قم بتوليد المجموعة الرئيسة منL1 ((D) Frequency_one_itemset) وضع عداد مجموعة العناصر يساوي صفر Count=0 وسميها المجموعة الرئيسة العامة.
3) امسح قاعدة البيانات D حتى النهاية:

1. اقرأ مجموعة العناصر من المعاملة واحذف العناصر غير الموجودة في L1 ثم قم بإنشاء مجموعة رئيسة محلية من العناصر المتبقية في المعاملة.
2. قارن المجموعة الرئيسة المحلية مع المجموعة الرئيسة العامة واحدة تلو الأخرة، وإذا كانت مجموعة العناصر متطابقة فقم بإضافة واحد للعداد.

قم بالاحتفاظ بمجموعة العناصر المرشحة التي تم الحصول عليها.

4) امسح المجموعة الرئيسة العامة واختبر كل عدد مجموعات العناصر مع مجموعة العناصر المرشحة.

1. إذا كان عدد مجموعة العناصر من مجموعة العناصر المرشحة أقل من minsup، قم بحذف مجموعة العناصر من المجموعة الرئيسة العامة.

7) العناصر المتبقية من مجموعة الرئيسة العامة العامة ستكون مجموعة العناصر التكرارية المطلوبة.

6. التطبيق العملي لخوارزمية Apriori المحسّنة:

يتم في هذا التطبيق استخدام مجموعة عناصر العمل الموضحة بالجدول(5):

Tid	Itemset	Tid	Itemset
1	**I3,I4**	6	**I2**
2	**I1,I2,I3,I4**	7	**I1,I3**
3	**I1,I3**	8	**I1,I3**
4	**I2,I3**	9	**I1,I2**
5	**I1,I2,I3**	10	**I2**

الجدول(5) مجموعات عناصر العمل في قاعدة البيانات D

تتكون مجموعة عناصر العمل من Tid {1 إلى 10} ، وتتكون مجموعات عناصر قاعدة البيانات من مجموعة العناصر {I1، I2، I3، I4}، وبفرض أن عتبة الدعم الصغرى التي يضعها الخبير هي ثلاثة (minsup=3) وبتنفيذ الخطوات الآتية:

الخطوة 1: يتم توليد مجموعة تكرارية واحدة من قاعدة البيانات D. كما في الجدول(6):

Items	Frequency of item
I1	6
I2	6
I3	7

الجدول(6) مجموعة عناصر التكرارية

تم حذف العنصر I4 لأن عدد مرات تكراره 2 وهو أقل من minsup.

الخطوة 2: يتم توليد المجموعة الرئيسة العامة ووضع count=0، كما في الجدول(7)

Candidate 1 Itemset	I1	I2	I3
1-Count	0	0	0
Candidate 2 Itemset	**I1,I2**	**I1,I3**	**I2,I3**
2-Count	0	0	0
Candidate 3 Itemset	**I1,I2,I3**		
3-count	0		

الجدول(7) ايجاد المجموعة الرئيسة العامة

الخطوة 3: مسح قاعدة البيانات مناقلة مناقلة.

يتم توضيح هذه الخطوة بالنسبة للمعاملة Tid = 1، والتي تحتوي العناصر {I3، I4}. ويتم حذف العنصر الغير موجود في L1 (أي I4)، وبعد الحذف من المعاملة Tid = 1 يكون هناك عنصر واحد فقط.

إذا كانت المعاملة بعد الحذف تحتوي على أكثر من عنصر واحد، يتم توليد مجموعة رئيسة محلية واحدة من هذه العناصر ومقارنتها مع المجموعة الرئيسة العامة.

في نهاية المسح الكامل لقاعدة البيانات، يتم تخزين البيانات الموضحة في الجدول(8) بالمجموعة الرئيسة العامة.

Candidate 1 Itemset	**I1**	**I2**	**I3**
1-Count	6	6	7
Candidate 2 Itemset	**I1,I2**	**I1,I3**	**I2,I3**
2-Count	3	5	3
Candidate 3 Itemset	**I1,I2,I3**		
3-count	2		

الجدول(8) إحصائيات مجموعات العناصر التكرارية المرشحة

الخطوة 4: بناءً على كل مجموعة عناصر مرشحة تم الحصول عليها من الجدول(8) ووفقًا لعدد مجموعة العناصر من مجموعة العناصر المرشحة ومقارنته مع minsup، يتم حذف مجموعات العناصر المرشحة التي يكون عددها أقل من عتبة الدعم الصغرى.

كما هو مبين في الجدول(9):

Candidate 1 Itemset	**I1**	**I2**	**I3**
1-Count	6	6	7
Candidate 2 Itemset	**I1,I2**	**I1,I3**	**I2,I3**
2-Count	3	5	3
Candidate 3 Itemset	I1,I2,I3		
3-Count	2		

الجدول(9) حذف مجموعات العناصر المرشحة التي عددها أقل من درجة الدعم الصغرى

الخطوة 5: بعد خطوة الحذف يتم عرض مجموعات العناصر التكرارية النهائية كما في الجدول(10):

Candidate 1 Itemset	**I1**	**I2**	**I3**
1-Count	6	6	7
Candidate 2 Itemset	**I1,I2**	**I1,I3**	**I2,I3**
2-Count	3	5	3

الجدول(10) المجموعات التكرارية

من الجدول(10) وبتحليل بياناته نستنتج أن أكبر مجموعة عناصر مستخرجة من قاعدة بيانات العمل هو مجموعتي عناصر تكرارية فقط هي(Candidate 1 Itemset, Candidate 2 Itemset).

7. الخلاصة

تولد خوارزمية Apriori المحسّنة أقصى عدد من مجموعات العناصر التكرارية، وتحتاج فقط إلى مسحين لقاعدة البيانات لتوليد أقصى عدد من مجموعات العناصر التكرارية. يتم في المسح الأول إيجاد مجموعة العناصر التكرارية (L1) فقط. والهدف الرئيسي من هذه الخطوة هو إنشاء مجموعة رئيسة عامة، والتي ستقلل من توليد مجموعات عناصر مرشحة غير ملائمة. ويتم في المسح الثاني قراءة المعاملة واحدة تلو الأخرى وحذف عناصر المعاملة الغير موجودة في القائمة (L1). وبهذا يتم تحسين الخوارزمية من خلال تقليل مسح قاعدة البيانات لمرات كثيرة وبالتالي تحقق فعاّلية عالية ونتائج سريعة.

وبمقارنة خوارزمية Apriori العادية مع المحسنة، نستنتج من خلال البحث أن الخوارزمية المحسّنة تقلل من وقت مسح قاعدة البيانات، وبالتالي تحسين كفاءة وجودة استخراج البيانات، وتقليل استهلاك موارد النظام وتحسين أدائه بشكل كبير.

8. المراجع:

1. Zifeng.X, Chen.C,2009-ONTOLOGY-BASED WEB MINING Computer Applications and software, Maebashi, Japan, 380p.
2. Agrawal.R,1993-Fast Algorithms for Mining Association Rules in Large Databases, Proceedings of the ACM SIGMOD International Conference Management of Data, Washington, p216.
3. Agrawal.R; Imielinski.T,1993-Swami.A; Mining Association rules between Sets of Items in large Databases,p283
4. Sheng.C, Jia.Y; Yang.C,2007-The Research of Improved Apriori Algorithm for Mining Association Rules, Service System and Service Management, International Conference on, Vol.no.pp 1-4.
5. Wanjun.Y, Xiaochun.W, Erkang.W, BowenC,2008-The research of improved apriori algorithm for mining association rules, Communication Technology, 2008. ICCT 11th IEEE International Conference on, vol., no., pp.513-516,
6. Yubo.J, Guanghu.X, Hongdan.F, Qian.Z, Xu.L,2012-An Improved Apriori Alogirhm Based on Association Analysis, Third International conference on networking and distributed computing, Vol,no,pp.208-211.
7. Rui.C, Zhiyi.L,2017-"An improved apriori algorithm,"

Electronics and Optoelectronics (ICEOE), International Conference on, vol.1, no., pp.V1-476-V1-478.

8. Sheila.A, Abaya.A,2019-Association rule mining based on apriori algorithm in minimizing candidate generation, International journal of scientific and engineering research volumev3, issue 7.