Université des Sciences et de la Technologie Houari Boumediene
DÉPARTEMENT D'IA ET SD

Master Informatique visuelle

Apriori

Apriori

KASHI Thiziri
MESSAOUDENE Lydia

UNIVERSITÉ DES SCIENCES ET DE LA TECHNOLOGIE HOUARI BOUMEDIENE, APRIL 2023

Contents

1 Introduction à l'algorithme Apriori et ses applications

1.1 Introduction

L'algorithme Apriori a été proposé en 1994 par Agrawal et Srikant comme une méthode d'exploration de données pour extraire des règles d'association à partir de données transactionnelles. Depuis lors, il est devenu un outil important dans de nombreux domaines tels que la gestion de la relation client, la recherche de marché et la publicité en ligne.

1.2 Description de l'algorithme Apriori

L'algorithme Apriori est un algorithme utilisé pour calculer les règles d'association entre des objets, En d'autres termes , comment deux ou plusieurs objets sont liés les uns aux autres. c'est un apprentissage de règles d'association qui analyse les personnes qui ont acheté le produit A ont également acheté le produit B.

L'objectif principal de l'algorithme Apriori est de créer la règle d'association entre différents objets. La règle d'association décrit comment deux ou plusieurs objets sont liés les uns aux autres. L'algorithme Apriori est également appelé fouille de motifs fréquents.

Prenons un exemple concret : imaginez que vous allez faire vos courses a UNO et que vous achetez différents produits. L'algorithme Apriori peut être utilisé pour déterminer les règles d'association entre ces produits, par exemple, que les personnes qui ont acheté du lait ont également acheté du pain. Cela peut aider les clients à acheter leurs produits plus facilement et augmenter les performances de vente de UNO.

Dans ce rapport, nous allons nous intéresser plus en détail à l'algorithme Apriori et à ses applications dans différents domaines, tels que la gestion de la relation client, la recherche de marché et la publicité en ligne. Nous examinerons également les limites de l'algorithme Apriori et les améliorations qui peuvent être apportées, comme l'utilisation des l'algorithmes Close , FP-Growth, pour surmonter certaines de ces limites. Enfin, nous discuterons egalement l'utilisation de l'amelioration du support , de l'utulisation de la correlation ou meme de fussionner apriori avec l'algorithme de clustering

2 Fonctionnement de l'algorithme Apriori

Pour mieux comprendre le foconctionnement d'Apriori , il faut d'abord comprendre quelque mot cle :

2.1 Itemset

: en anglais " a set of items together is called an itemset " , Itemset est un ensemble de 2 ou plus item

2.2 Frequent Itemset

un ensemble d'item est dit frequent si ca support value et confidence est superieur a un certain seuil minimum

2.3 Association Rules

aussi appeler les regles d'association en francais , elle sont utiliser pour determiner la relation entre des attribut dans un dataset ,exemple : A=>B , cela veux dire que certaine valeur de A detrime la valeur B

2.4 Support et confidence

$$\text{Support (A)} = \frac{\text{Number of transaction in which A appears}}{\text{Total number of transactions}}$$

$$\text{Confidence (A}\longrightarrow\text{B)} = \frac{\text{Support(AUB)}}{\text{Support(A)}}$$

2.5 Etape :

Prenons un exemple simple supposant que dans un supermarché nous vendons les produits suivant :Riz,huile,lait ,poulet,pomme .
Nous allons annalyser 6 transactions de differents clients , nous considererons le 0 comme produit non-acheter et le 1 pour produit achete

Transaction ID	Riz	poulet	huile	lait	pomme
t1	1	1	1	0	0
t2	0	1	1	1	0
t3	0	0	0	1	1
t4	1	1	0	1	0
t5	1	1	1	0	1
t6	1	1	1	1	1

L'algorithme Apriori repose sur les bases suivantes :

- Si P(I) est inférieur au seuil de support minimum, alors I n'est pas considéré comme fréquent.
- Si P(I+A) est inférieur au seuil de support minimum, alors I+A n'est pas considéré comme fréquent, où A appartient également à l'ensemble d'éléments.
- Si un ensemble d'éléments a une valeur inférieure au support minimum, alors tous ses ensembles supérieurs seront également inférieurs au support minimum et pourront donc être ignorés. Cette propriété est appelée **propriété d'antimonotonie.**

Étape 1 : Génération des ensembles de fréquences :
Dans cette étape, l'algorithme scanne la base de données transactionnelle pour identifier les ensembles d'éléments fréquents. Un ensemble d'éléments est considéré comme fréquent s'il apparaît dans un nombre suffisant de transactions, selon un seuil de support défini à l'avance.
dans notre cas nous avons la table suivante qui indique les prduit les plus acheter par le client (support>50
Support threshold= 50=> 0.5*6= 3 => minsup=3

Produit	freauence
Riz	4
poulet	5
huile	4
lait	4

Etape 2:Creation d'un itemset apartir des item restant de l'etape precedente

Produit	freauence
RP	4
RO	3
RM	2
PO	4
PM	3
OM	2

Nous nous débarrassons des items en dessous du threshold il nous reste RP,RO,PO,PM
Creation d'un itemset apartir des item restant de l'etape precedente

Produit	freauence
RPO	4
POM	2

et donc RPO sont les items les plus frequent

2.6 Generate Association Rules:

dans cette etapes nous devons genere toute les regles possible pour cette exemple ca sera :
R,P=>O,R,O=>P, R=>P,O , O=>R,P ,P,O=>R, P=>R,P
Nous calculerons "confidence" avec la formule site au paravant et nous eliminerons les regles nons segnifiante grace au seuil de confidencialite .

3 Limitations de l'algorithme Apriori

Bien que l'algorithme Apriori soit une technique très populaire pour l'exploration des données, il présente certains inconvénients, tels que :

- **Coût de calcul :** La technique Apriori prend beaucoup de temps à calculer car elle doit générer et examiner de nombreux ensembles de données pour dériver les règles d'association.
- **Dimensions de la base de données :** La méthode Apriori peut rencontrer des problèmes lorsqu'elle est appliquée à des ensembles de données comportant un volume élevé de transactions, car elle doit examiner chaque transaction pour trouver des ensembles d'éléments récurrents.
- **Dépendance aux paramètres :** L'algorithme Apriori dépend fortement de facteurs tels que les seuils de soutien et de confiance. Le choix de ces paramètres peut avoir un impact significatif sur les résultats de l'algorithme
- **Incapacité à gérer des données complexes :** L'algorithme Apriori est conçu pour traiter des données transactionnelles simples, telles que les achats de produits dans un magasin. Il peut rencontrer des difficultés lorsqu'il traite des données plus complexes, telles que des données textuelles brutes ou des données multidimensionnelles.
- **Problèmes liés à la dimensionnalité :** L'algorithme Apriori peut souffrir d'un mauvais calcul de la dimensionnalité, ce qui signifie que le coût du calcul augmente de manière exponentielle avec le nombre de dimensions, rendant l'approche inefficace pour les grands ensembles de données.

Ces limites doivent être prises en compte lors de l'utilisation de l'algorithme d'Apriori, et d'autres approches peuvent être envisagées pour des ensembles de données plus importants ou plus compliqués et c'est ce que nous allons voir dans les prochains chapitre

4 Présentation de l'algorithme Close et ses avantages

La méthode close est une variante de l'algorithme Apriori qui identifie les groupes d'éléments fréquents et fixes dans une base de données. Les ensembles d'éléments fermés sont des groupes d'éléments qui ne peuvent pas être étendus tout en conservant leur support et leur fermeture. En d'autres termes, ils ne peuvent pas être améliorés sans changer leur signification.

4.1 groupes fermés d'éléments

Un groupe d'éléments est considéré comme fermé s'il ne peut être étendu sans modifier son support et sa fermeture.

4.2 Etape

L'algorithme Close suit les mêmes étapes de base que l'algorithme Apriori, mais il ajoute des étapes supplémentaires pour identifier les groupes fermés d'éléments. Les étapes supplémentaires sont les suivantes :

- Générer tous les groupes d'éléments fréquents, comme dans la méthode Apriori.
- Vérifier si chaque groupe d'éléments fréquents est fermé ou non.
- Supprimez tous les groupes d'éléments qui ne sont pas fermés
- Trouvez tous les sous-ensembles fermés de chaque ensemble fermé d'éléments trouvé.
- Supprimer les sous-ensembles fermés qui partagent le même support que leur ensemble parent.
- Renvoyer les groupes d'éléments fermés restants.

4.3 Comparaision de l'algorithme close a apriori

La méthode Close et l'algorithme Apriori sont tous deux des algorithmes de fouille de données qui sont utilisés pour identifier des modèles dans un ensemble de données. Néanmoins, il existe des différences significatives entre les deux algorithmes.
L'algorithme Apriori est souvent considéré comme plus simple et plus rapide que l'algorithme Close car il utilise une approche de recherche **large** qui implique l'exploration de toutes les combinaisons d'éléments potentielles dans un premier temps, suivie de l'exclusion des combinaisons peu fréquentes. Close, quant à lui, utilise une approche de recherche **approfondie** qui examine chaque combinaison d'éléments en profondeur et détermine si elle est fermée ou non.
Cependant, l'approche Close présente l'avantage de générer des ensembles fermés d'éléments, qui peuvent être plus utiles pour l'analyse des données que les ensembles ouverts d'éléments de l'algorithme Apriori. Les groupes d'éléments fermés sont des groupes d'éléments qui ne peuvent être étendus sans modifier leur support et leur fermeture, ce qui implique qu'ils représentent des modèles de données plus significatifs et precis

5 Amélioration du calcul du support dans l'algorithme Apriori

Le choix du calcul du support dans la méthode Apriori est principalement déterminé par la taille de la base de données et la mémoire disponible.
Si la base de données est relativement petite, il est possible de stocker toutes les transactions en mémoire et de parcourir la base de données plusieurs fois pour calculer le support de chaque itemet. Cette méthode est plutôt basique et ne nécessite pas de techniques avancées de gestion de la mémoire ou de parallélisation.
Néanmoins, pour les bases de données plus importantes, le stockage de toutes les transactions en mémoire peut s'avérer peu pratique. Dans ce cas, des structures de données telles que les tables de hachage peuvent être utilisées pour stocker les transactions de manière efficace et compacte. Cette méthode permet de réduire l'utilisation de la mémoire et d'augmenter le temps de calcul du support.

6 Utilisation de la corrélation pour améliorer l'efficacité de l'algorithme Apriori

L'utilisation de la corrélation peut être une méthode efficace pour améliorer l'efficacité de l'algorithme Apriori. En effet, la corrélation peut aider à identifier les itemsets qui ont tendance à apparaître ensemble, même s'ils ne répondent pas toujours à l'exigence de support minimal.
L'idée est de calculer une mesure de corrélation entre chaque paire d'itemsets et d'utiliser cette mesure pour identifier les itemsets ayant une forte corrélation. Ces ensembles peuvent ensuite être combinés pour former des ensembles plus importants sans qu'il soit nécessaire de vérifier leur support individuel.
Plusieurs mesures de corrélation peuvent être utilisées, notamment le coefficient de corrélation de Pearson, le coefficient de corrélation de Spearman et le coefficient de corrélation de Kendal-Tau. Ces mesures peuvent être calculées en utilisant la fréquence d'apparition de chaque paire d'itemsets dans la base de données.
L'utilisation de la corrélation peut contribuer à réduire le temps de calcul et le nombre d'ensembles d'éléments candidats générés par la méthode Apriori tout en conservant une grande précision. Cependant, cette méthode peut ne pas être applicable à toutes les applications car elle suppose que les itemsets sont fortement corrélés.

7 Utilisation de l'algorithme de clustering en conjonction avec l'algorithme Apriori

L'utilisation de l'algorithme de regroupement en conjonction avec l'algorithme Apriori peut être une méthode efficace pour améliorer l'efficacité et la pertinence des résultats obtenus.
L'idée est d'utiliser un algorithme de clustering pour regrouper les transactions similaires en grappes, puis d'appliquer la méthode Apriori à chaque grappe individuellement plutôt qu'à l'ensemble des transactions. Cela permet de réduire le nombre de transactions à considérer, et donc le temps nécessaire au calcul de la méthode Apriori.
En outre, en regroupant des transactions similaires, la méthode de regroupement peut aider à identifier des corrélations plus significatives entre les articles. Par exemple, les articles fréquemment achetés ensemble dans un certain groupe peuvent présenter une corrélation plus forte que s'ils étaient considérés dans l'ensemble des transactions.
Toutefois, l'utilisation d'un algorithme de regroupement peut poser des problèmes si les regroupements ne sont pas représentatifs de l'ensemble des transactions. En outre, la technique de regroupement et ses paramètres doivent être adaptés à chaque application en fonction de la nature des données.

8 temps d'execution

Nous avons tester sur 2 datasets :

- la dataset titanic de 891 lignes nous avons optenue un temps d'execution de 22s pour apriori
- la dataset wether qui contient 20 lignes nous avons obtenu un temps de 0.00001s pour l'algorithme apriori