

NIM : 2211522011

Tanggal : Rabu, 03 April 2024

Nama : Ghifari Rizki Ramadhan

Asisten : 1. Muhammad Irsyadul Fikri  
2. Athifa Rifda Andra  
3. Annisa Gita Subhi  
4. Syakina Triyana  
5. Husna Afiqah Yossyafra  
6. Vania Zerlina Utami  
7. Ghina Fitri Hidayah  
8. Sukma Anggarmadi  
9. Rafiqatul Ulya

Mata Kuliah : Praktikum Data Mining

Modul : 06

Kelas : A

## Resume dan Tugas “Teknik Asosiasi dengan Algoritma Apriori”

### A. Asosiasi

Association rule mining atau analisis asosiasi adalah teknik data mining untuk menemukan aturan asosiasi antara suatu kombinasi item. Contoh aturan asosiasi dari analisa pembelian di suatu pasar swalayan adalah dapat diketahuinya berapa besar kemungkinan seorang pelanggan membeli roti bersamaan dengan susu. Dengan pengetahuan tersebut pemilik pasar swalayan dapat mengatur penempatan barangnya atau merancang kampanye pemasaran dengan memakai kupon diskon untuk kombinasi barang tertentu. Analisis asosiasi menjadi terkenal karena aplikasinya untuk menganalisa isi keranjang belanja di pasar swalayan. Analisis asosiasi juga sering disebut dengan istilah market basket analysis. Analisis asosiasi dikenal juga sebagai salah satu teknik data mining yang

menjadi dasar dari berbagai teknik data mining lainnya. Khususnya salah satu tahap dari analisis asosiasi yang disebut analisis pola frekuensi tinggi (frequent pattern mining) menarik perhatian banyak peneliti untuk menghasilkan algoritma yang efisien. Dalam gambaran umum, Association Rule Mining memiliki dua tahap proses yaitu:

- Temukan semua frequent itemsets: anggota frequent itemsets setidaknya memiliki jumlah dalam database sama dengan minimum support yang telah ditentukan di awal.
- Temukan semua aturan asosiasi yang kuat dari frequent itemsets aturan asosiasi yang dihasilkan harus memenuhi minimum support dan minimum confidence.

Association Rule adalah teknik data mining untuk menemukan aturan asosiatif dengan kombinasi item. Asosiasi adalah aturan yang mendukung pengolahan data, mengidentifikasi korelasi (generalisasi asosiasi), dan mempertimbangkan urutan pengambilan materi sebagai suatu basis pembangkit aturan.

Masalah dalam pencarian seluruh association rule dapat dibagi menjadi dua, yaitu bagaimana menemukan seluruh item-item (itemset) yang memiliki support di atas minimum support yang telah ditentukan. Support untuk sebuah itemset adalah jumlah transaksi yang terdapat dalam itemset itu. Itemset dengan minimum support disebut large itemset (I-itemset). Masalah yang kedua adalah bagaimana menggunakan large itemset untuk menghasilkan aturan-aturan yang diperlukan. Aturan yang hendak dihasilkan adalah dalam bentuk  $a \rightarrow (I-a)$ , yaitu perbandingan antara support (I) dengan support (a) haruslah lebih besar dari minconf (minimum confidence). Misalkan  $[I] = \{I_1, I_2, \dots, I_m\}$  adalah himpunan literal yang disebut item. Himpunan item-item disebut dengan itemset. D adalah himpunan transaksi, di mana setiap transaksi T itemset sehingga  $TE$ . I. Sebuah association rule adalah sebuah implikasi berbentuk  $X \rightarrow Y$ , di mana  $X \subseteq I$ ,  $Y \subseteq I$ , dan  $X \cap Y = \emptyset$ .

## B. Apriori

Algoritma Apriori adalah algoritma yang digunakan untuk menghitung aturan asosiasi antar objek. Aturan asosiasi menjelaskan bagaimana dua atau lebih objek terkait satu sama lain. Dengan kata lain, algoritma apriori adalah

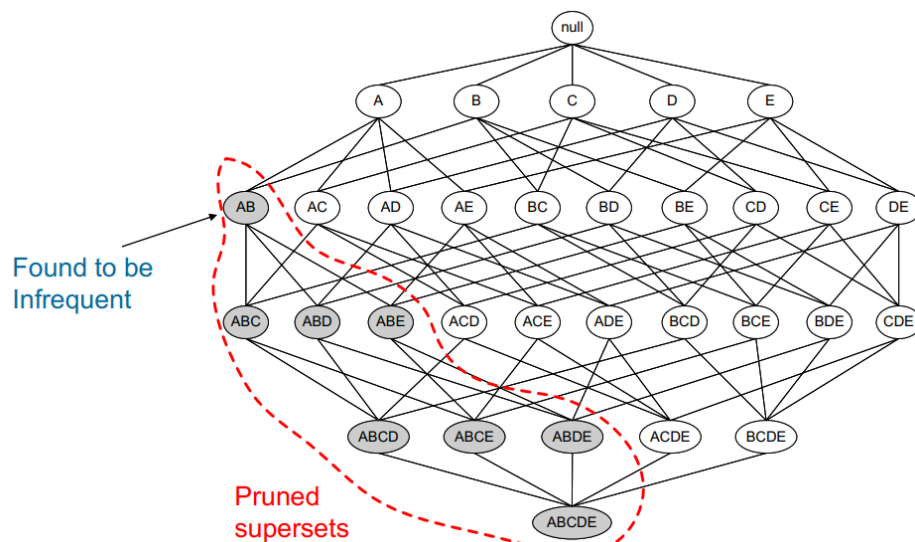
algoritma berbasis aturan asosiasi yang menganalisis apakah orang yang membeli produk A juga membeli produk B.

Algoritma ini dikemukakan oleh ilmuwan R. Agrawal dan Srikant pada tahun 1994. Algoritma ini kebanyakan digunakan untuk analisis keranjang belanja dan membantu menemukan produk yang dapat dibeli bersama. Apriori juga dapat digunakan di bidang perawatan kesehatan untuk menemukan reaksi obat untuk pasien.

Pada data mining, algoritma Apriori banyak digunakan untuk menemukan data yang paling sering muncul dalam sebuah database. Item data transaksi pada database membentuk itemset. Itemset yang paling sering muncul dipilih oleh algoritma Apriori agar dapat digunakan untuk menentukan aturan asosiasi yang menyoro tren umum dalam database. +

#### Cara kerja algoritma apriori

Untuk menentukan itemset yang paling sering muncul, algoritma Apriori menggunakan pendekatan "bottom-up". Pendekatan ini menggunakan Breadth-first search dan struktur data Hash tree untuk menghitung kandidat itemset secara efisien.



Pendekatan bottom up dimulai dari setiap item dalam daftar itemset. Kemudian, kandidat dibentuk dengan self-join (penggabungan). Setiap iterasi, kita memperluas panjang itemset satu item yang membentuk subhimpunan.

Setelah itu dilakukan proses pengujian terhadap subhimpunan. Apabila ditemukan itemset yang berisi subhimpunan yang jarang, maka dilakukan pruning (pemangkasan). Proses dilakukan secara berulang-ulang sampai tidak ada lagi itemset yang berhasil diturunkan dari data.

### Langkah-langkah algoritma apriori

Berikut ini adalah langkah-langkah dari algoritma Apriori :

- Hitung support dari itemset (dengan ukuran  $k = 1$ ) dalam database. Proses ini akan menghasilkan himpunan kandidat.
- Pangkas (pruning) kumpulan kandidat dengan cara menghilangkan item yang memiliki support lebih kecil dari ambang batas (threshold) yang diberikan.
- Gabungkan itemset yang paling sering muncul untuk membentuk himpunan berukuran  $k + 1$ , dan ulangi himpunan di atas sampai tidak ada lagi itemset yang dapat dibentuk.

### C. Tugas

#### 1. Data tidak memiliki kepala tabel

```
pip install apyori

Collecting apyori
  Downloading apyori-1.1.2.tar.gz (8.6 kB)
  Preparing metadata (setup.py) ... done
Building wheels for collected packages: apyori
  Building wheel for apyori (setup.py) ... done
  Created wheel for apyori: filename=apyori-1.1.2-py
  Stored in directory: /root/.cache/pip/wheels/c4/1a
Successfully built apyori
Installing collected packages: apyori
Successfully installed apyori-1.1.2

[17] import pandas as pd
     from apyori import apriori

[18] data = pd.read_csv("dataapriori_tugas.csv")

[19] data
```

## 2. Data diolah dengan range penuh

```
record = []
for i in range(0, len(data)):
    record.append([str(data.values[i, j]) for j in range(0, 20)])
```

## 3. Minimal lift = '5', min\_confidence = 0.2

```
[24] association_rules = apriori(record, min_support=0.0045, min_confidence=0.2, min_lift=5, min_length=2)
     association_results = list(association_rules)

[25] def association(association_results):
     lhs = [tuple(result[2][0][0]) for result in association_results]
     rhs = [tuple(result[2][0][1]) for result in association_results]
     support = [result[1] for result in association_results]
     confidence = [result[2][0][2] for result in association_results]
     lift = [result[2][0][3] for result in association_results]
     return list(zip(lhs, rhs, support, confidence, lift))
```

## 4. Tidak ada data duplikat

```
dr = data[data.duplicated()]
print(dr.shape)
```

```
(968, 20)
```

```
[21] data = data.drop_duplicates()
     data
```

```
dr = data[data.duplicated()]
print(dr.shape)
```

```
(0, 20)
```

## 5. Urutkan Output Akhir Berdasarkan Lift teratas

```
resultdata = pd.DataFrame(association_results, columns=['Product 1', 'Product 2', 'Support', 'Confidence', 'Lift'])
resultdata['Rule'] = resultdata['Product 1'] + ' -> ' + resultdata['Product 2']
a = resultdata.sort_values(['Lift'], ascending = False)
b = a.drop_duplicates(subset=['Product 1', 'Product 2'], keep='first')
b.head(5)
```

	Product 1	Product 2	Support	Confidence	Lift	Rule
3	soup	milk	0.005439	0.466667	6.710615	soup -> milk
0	pasta	shrimp	0.009324	0.558140	5.462552	pasta -> shrimp
2	chocolate	french fries	0.005051	0.203125	5.281250	chocolate -> french fries

## Daftar Pustaka

*Wikipedia,*

[https://lms-paralel.esaunggul.ac.id/pluginfile.php?file=%2F313692%2Fmod\\_resource%2Fcontent%2F2%2F9\\_7705\\_%20MIK620\\_052018\\_pdf.pdf](https://lms-paralel.esaunggul.ac.id/pluginfile.php?file=%2F313692%2Fmod_resource%2Fcontent%2F2%2F9_7705_%20MIK620_052018_pdf.pdf). Accessed 7 April 2024.

“Algoritma Apriori: Pengertian, Cara Kerja, Kelebihan, dan Kekurangannya.” *Trivusi*, 17 September 2022, <https://www.trivusi.web.id/2022/08/algoritma-apriori.html>. Accessed 7 April 2024.

