Webinar Series, 7 Nopember 2020
PROGRAM PASCASARJANA TERAPAN
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

Workshop & Tutorial
Data Mining with Python

Association Rule

Ali Ridho Barakbah

Knowledge Engineering Laboratory
Department of Information and Computer Engineering
Politeknik Elektronika Negeri Surabaya

Assocation rule?

- Mencari suatu kaidah keterhubungan dari data
- Diusulkan oleh Agrawal, Imielinski, and Swami (1993)

- Contoh: Dalam suatu supermarket kita ingin mengetahui seberapa jauh orang yang membeli celana juga membeli sabuk?

- Manfaat: Dapat digunakan untuk Market Basket Analysis (menganalisa kebiasaan customer dengan mencari asosiasi dan korelasi dari data transaksi)
  - Sebagai saran penempatan barang dalam supermarket
  - Sebagai saran produk apa yang dipakai dalam promosi

Support & Confidence

- Misalkan $I=\{I_1, I_2, \ldots, I_m\}$ merupakan suatu himpunan dari literal, yang disebut item-item.
- Misalkan $D=\{T_1, T_2, \ldots, T_n\}$ merupakan suatu himpunan dari n transaksi, dimana untuk setiap transaksi $T \in D$, $T \subseteq I$.
- Suatu himpunan item $X \subseteq I$ disebut itemset.
- Suatu transaksi T memuat suatu itemset X jika $X \subseteq T$.
- Setiap itemset X diasosiasikan dengan suatu himpunan transaksi $T_X = \{T \in D \mid T \supseteq X\}$ yang merupakan himpunan transaksi yang memuat itemset

- Support dari itemset X → supp(X)) :

$$|T_X| / |D|$$

- Confidence (keyakinan) dari kaidah X→Y, ditulis conf(X → Y) adalah
  - conf(X→Y) = supp(X∪Y) / supp(X)
  - Confidence bisa juga didefinisikan dalam terminologi peluang bersyarat

$$conf(X \rightarrow Y) = P(Y|X) = P(X \cap Y) / P(X)$$

Contoh

Transaksi	A	B	C	D
T1	1	0	1	14
T2	0	0	6	0
T3	1	0	2	4
T4	0	0	4	0
T5	0	0	3	1
T6	0	0	1	13
T7	0	0	8	0
T8	4	0	0	7
T9	0	1	1	10
T10	0	0	0	18

Jumlah transaksi |D| = 10

Kemunculan item A pada transaksi (|Ta|) sebanyak 3 kali yaitu pada T1, T3, T8.

Supp(A)=|Ta|/|D| = 3/10 = 0.3.

|Tcd| sebanyak 5 kali, yaitu pada T1, T3, T5, T6, T9.
Supp(CD)=|Tcd|/|D| = 5/10 = 0.5.

Frequent itemset adalah itemset yang mempunyai support >= minimum support yang diberikan oleh user.

Itemset	Sp
A	0.3
B	0.1
C	0.8
D	0.7
AB	0
AC	0.2
AD	0.3
BC	0.1
BD	0.1
CD	0.5
ABC	0
ABD	0
ACD	0.2
BCD	0.1
ABCD	0

Jika minsupport diberikan oleh user sebagai threshold adalah 0.2, maka frequent itemset adalah semua itemset yang support-nya >= 0.2, yakni

A, C, D, AC, AD, CD, ACD

Dari frequent itemset bisa dibangun kaidah asosiasi sbb:

A → C C → A A → D
D → A C → D D → C
A,C → DA,D → CC,D → A

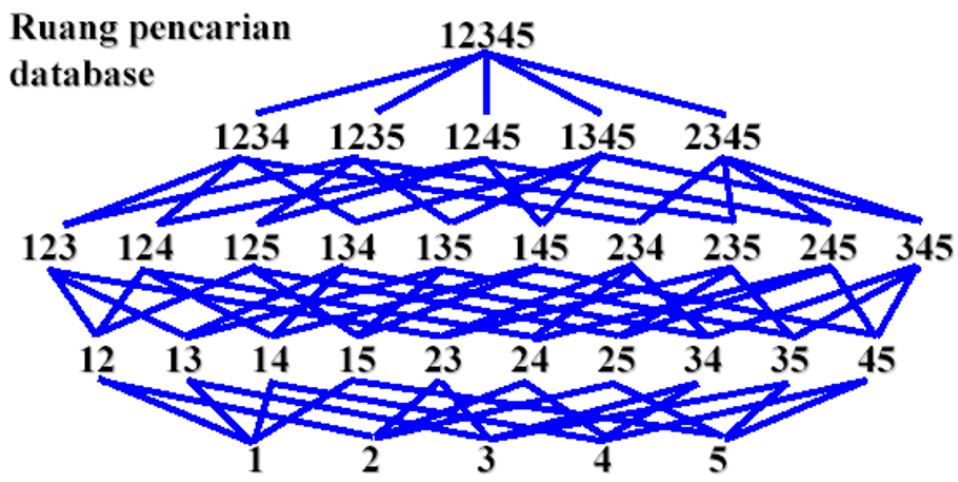
Conf(A→C) = supp(A,C) / supp(A)

Apriori

- Prinsip apriori :

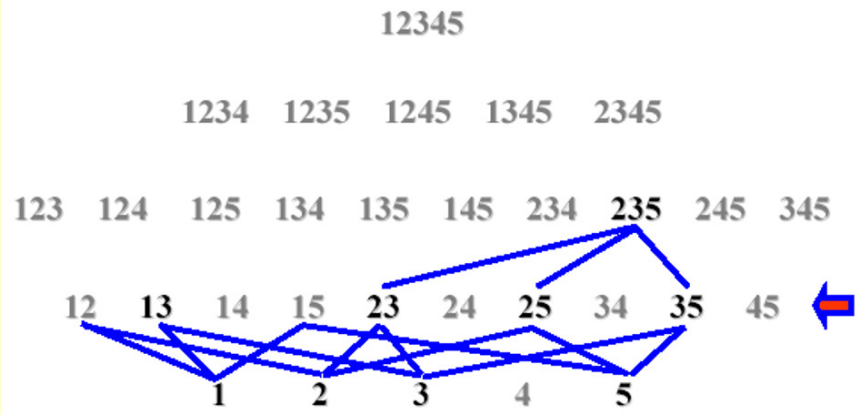
  Subset apapun dari suatu frequent itemset harus frequent
- L3={abc, abd, acd, ace, bcd}
- Penggabungan sendiri : L3*L *L3
  - abcd dari abc dan abd
  - acde dari acd dan ace
- Pemangkasan Pemangkasan:
  - acde dibuang sebab ade tidak dalam L3
- C4={abcd}

Ilustrasi Search space pada apriori



Ruang pencarian tanpa apriori

Ruang pencarian dengan apriori



Contoh

T1	{roti, selai, mentega}
T2	{roti, mentega}
T3	{roti, susu, mentega}
T4	{coklat, roti}
T5	{coklat, susu}

Itemset	Sp
{roti}	0.8
{selai}	0.2
{mentega}	0.6
{susu}	0.4
{coklat}	0.4

Itemset	Sp
{roti,mentega}	0.6
{roti,susu}	0.2
{roti,coklat}	0.2
{mentega,susu}	0.2
{mentega,coklat}	0
{susu,coklat}	0.2

Conf(roti→mentega) = Supp({roti,mentega})/Supp({roti})
= 0.6 / 0.8 = 0.75 → 75%
Conf(mentega→roti) = Supp({mentega,roti})/Supp({mentega})
= 0.6 / 0.6 = 1 → **100%**

Eksperimen dengan Data Pembelian

No_Kwitansi,Nama_Barang,Jumlah
1, cpu, 7
1, monitor, 20
1, mouse, 4
2, monitor, 9
2, meja, 4
2, cpu, 5
2, mic, 12
2, speaker, 12
3, mic, 5
3, speaker, 5
3, ram, 3
4, ram, 2
4, harddisk, 2
4, flashdisk, 8

5, speaker, 1
5, flashdisk, 5
5, cpu, 2
6, speaker, 3
6, mic, 5
6, monitor, 2
6, flashdisk, 3
7, cpu, 2
7, monitor, 5
7, meja, 2
8, monitor, 9
8, cpu, 6
8, ram, 4

Association Rule

```
import pandas as pd
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules

dataset = pd.read_csv('pembelian.csv')
transaksi = dataset.groupby(['No_Kwitansi','Nama_Barang'])['Jumlah'].sum()

transaksi = transaksi.unstack().reset_index().fillna(0).set_index('No_Kwitansi')
transaksi[transaksi>0]=1

print('Tabel Transaksi:\n', transaksi)

frequent_itemsets=apriori(transaksi, min_support=0.3, use_colnames=True)
rules=association_rules(frequent_itemsets, metric="confidence", min_threshold=0.7)

print('\nAssociation Rules:\n', rules[['antecedents', 'consequents', 'confidence']])
```

```
Tabel Transaksi:
 Nama_Barang    cpu   flashdisk   harddisk  ...   mouse   ram   speaker
No_Kwitansi                                ...
1              1.0         0.0        0.0  ...     1.0   0.0       0.0
2              1.0         0.0        0.0  ...     0.0   0.0       1.0
3              0.0         0.0        0.0  ...     0.0   1.0       1.0
4              0.0         1.0        1.0  ...     0.0   1.0       0.0
5              1.0         1.0        0.0  ...     0.0   0.0       1.0
6              0.0         1.0        0.0  ...     0.0   0.0       1.0
7              1.0         0.0        0.0  ...     0.0   0.0       0.0
8              1.0         0.0        0.0  ...     0.0   1.0       0.0

[8 rows x 9 columns]

Association Rules:
   antecedents consequents  confidence
0      ( cpu)  ( monitor)        0.80
1  ( monitor)      ( cpu)        0.80
2  ( speaker)      ( mic)        0.75
3      ( mic)  ( speaker)        1.00
```

Mlxtend (http://rasbt.github.io/mlxtend):

- conda **install** mlxtend
- conda **install** mlxtend --channel conda-forge
- pip install mlxtend
- pip install mlxtend --upgrade --no-deps