04 – Association Rules

Data Science and Management

Corso di Laurea Magistrale in Ingegneria Gestionale

Marco Mamei, Natalia Hadjidimitriou, Fabio D'Andreagiovanni, Matteo Martinelli

{marco.mamei, selini, fabio.dandreagiovanni, matteo.martinelli}@unimore.it

- Data Mining
- Association Rules
- A-priori

Data Mining

Transform data into **intelligible knowledge**

Data mining is sometimes also referred to as **Knowledge Discovery in Databases** (KDD)

Learning vs. Mining

Main difference

- Machine Learning has to do with **learning** a model from data and use it to classify, forecast, sometimes even generate new data
- Data Mining has to do with **searching** for frequent patterns, rules, co-occurrences in data

Of course, there are several **overlapping** methods (e.g., clustering, rule learning, statistical approaches)

Basket Analysis

A classic example — Supermarket items

Suppose to have a database of **basket** data that have been purchased at a supermarket by some users. It would be interesting to know whether there are some **regularities** in the form of patterns of rules among the products that have been purchased together.

Basket Analysis

A classic example — Supermarket items

For example, it would not be surprising to see that people that purchase **cereals** and **sugar** also purchase **milk** in 90% of the cases.

Other Applications…

- **Recommendation systems.** For example, a music streaming service might use association rule mining to recommend new artists or albums to a user based on their listening history.

- **Customer Segmentation.** For example, a company might use association rule mining to discover that customers who purchase certain types of products are more likely to be younger. Similarly, they could learn that customers who purchase certain combinations of products are more likely to be located in specific geographic regions.

- **Fraud Detection.** For example, a credit card company might use association rule mining to identify patterns of fraudulent transactions, such as multiple purchases from the same merchant within a short period of time.

- **Social network analysis.** For example, an analysis of Twitter data might reveal that users who tweet about a particular topic are also likely to tweet about other related topics, which could inform the identification of groups or communities within the network.

In general they are mechanism to identify relationships between features.

Basket Analysis

Formal problem definition

$I = \{I_1, \ldots, I_m\}$ a set of **binary attributes**

$T = \{T_1, \ldots, T_n\}$ a **database of transactions**

Each transaction T_j is a **binary vector** of m elements where $T_j[k] = 1$ if item I_k was purchased in transaction T_j and $T_j[k] = 0$ otherwise

Basket Analysis

Formal problem definition

$I = \{I_1, \ldots, I_m\}$ a set of **binary attributes**

$T = \{T_1, \ldots, T_n\}$ a **database of transactions**

Given a subset X of some items in I, we say that transaction T_j **satisfies** X if for all items I_k in X we have that $T_j[k] = 1$

Basket Analysis

Formal problem definition

$I = \{I_1, \ldots, I_m\}$ a set of **binary attributes**

$T = \{T_1, \ldots, T_n\}$ a **database of transactions**

By an **association rule** we mean an **implication** of the form $X \Rightarrow I_k$, where X is some subset of items in I and I_k **is not present** in X

Support and Frequency

We define the **support** of an itemset as the **fraction of transactions** that support (contain) that itemset

A **frequent** itemset is one with at least the **minimum support** (i.e., present at least N% in the database)

Every **subset** of a frequent set is a **frequent** set (suppose {Cereal,Milk} is frequent, than both {Cereal} and {Milk} are necessarily frequent)

Support and Frequency

We define the **support** of a rule R as the **fraction of transactions** that support (contain) **the union of the antecedent and the consequent of the rule**

$$\text{Support}(A \Rightarrow B) = |t \in T \, \text{such that} \, \{A \cup B\} \subseteq t|$$

Confidence

We define the **confidence** of a rule as the **ratio** between the **support of the rule** and the **support of the antecedent**

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

Association Rule Mining

Example

	Milk	Cereals	Biscuits	Ham	Tea
T1	1	1	0	0	0
T2	1	0	1	0	1
T3	1	1	1	0	0
T4	0	0	1	1	1
T5	1	1	1	0	1

Association Rule Mining

Example

Are {M,C}, {T,H} frequent itemsets?
{M,C} — 3 occurrences out of 5
{T,H} — 1 occurrence out of 5

{M,C} **is** a frequent itemset
{T,H} **is not** a frequent itemset

MIN-SUPPORT = 30%

	M	C	B	H	T
T1	**1**	**1**	0	0	0
T2	1	0	1	0	1
T3	**1**	**1**	1	0	0
T4	0	0	1	**1**	**1**
T5	**1**	**1**	1	0	1

Association Rule Mining

Example

Are {M,C,B}, {B,H,T} frequent itemsets?

{M,C,B} — 2 occurrences out of 5

{B,H,T} — 1 occurrence out of 5

{M,C,B} **is** a frequent itemset

{B,H,T} **is not** a frequent itemset

	M	C	B	H	T
T1	1	1	0	0	0
T2	1	0	1	0	1
T3	**1**	**1**	**1**	0	0
T4	0	0	**1**	**1**	**1**
T5	**1**	**1**	**1**	0	1

Association Rule Mining

Example

Consider the following associative rules:

- {M,C} => B
- {C,B} => M

Compute confidence and support

	M	C	B	H	T
T1	1	1	0	0	0
T2	1	0	1	0	1
T3	1	1	1	0	0
T4	0	0	1	1	1
T5	1	1	1	0	1

Association Rule Mining

Example

- {M,C} => B

Support = 0.4
Confidence = 0.4/0.6= 66%

	M	C	B	H	T
T1	1	1	0	0	0
T2	1	0	1	0	1
T3	1	1	1	0	0
T4	0	0	1	1	1
T5	1	1	1	0	1

Association Rule Mining

Example

- {C,B} => M

Support = 0.4
Confidence = 0.4/0.4 = 100%

	M	C	B	H	T
T1	1	1	0	0	0
T2	1	0	1	0	1
T3	1	1	1	0	0
T4	0	0	1	1	1
T5	1	1	1	0	1

Lift

We define the **lift** of a rule A => B as the **ratio** between the support of the rule and the **product** of the supports of antecedent and consequent

$$\mathrm{Lift}(A \Rightarrow B) = \frac{\mathrm{Support}(A \Rightarrow B)}{\mathrm{Support}(A) \times \mathrm{Support}(B)}$$

$$\mathrm{Lift}(A \Rightarrow B) = \frac{\mathrm{Confidence}(A \Rightarrow B)}{\mathrm{Support}(B)}$$

Lift

- Lift > 1 implies that A and B appear together more than expected (**positive dependence**)
- Lift < 1 implies that A and B appear together less than expected (**negative dependence**)
- Lift = 1 implies that A and B appear together as often as expected (**independence**)

Caveat! Lift for {A=>B} is **the same** as that for {B=>A}

Association Rule Mining

Example

Consider the following associative rules:

- {M,C} => B
- {C,B} => M

Compute the lift of such rules

	M	C	B	H	T
T1	1	1	0	0	0
T2	1	0	1	0	1
T3	1	1	1	0	0
T4	0	0	1	1	1
T5	1	1	1	0	1

Association Rule Mining

Example

- {M,C} => B

Support = 0.4

Confidence = 0.4/0.6= 66%

Support {M,C} = 0.6

Support {B} = 0.8

Lift = 0.4/(0.6*0.8) = 0.833

	M	C	B	H	T
T1	1	1	0	0	0
T2	1	0	1	0	1
T3	1	1	1	0	0
T4	0	0	1	1	1
T5	1	1	1	0	1

Association Rule Mining

Example

- {C,B} => M

Support = 0.4

Confidence = 0.4/0.4 = 100%

Support {C,B} = 0.4

Support {M} = 0.8

Lift = 1.25

	M	C	B	H	T
T1	1	1	0	0	0
T2	1	0	1	0	1
T3	1	1	1	0	0
T4	0	0	1	1	1
T5	1	1	1	0	1

Interest

Interest of a rule R (very similar to lift)

- **Difference** between its **confidence** and the **support of the consequent** of R

$$\text{Interest}(A \Rightarrow B) = \text{Confidence}(A \Rightarrow B) - \text{Support}(B)$$

Interest

If A has **no influence** on B, then the number of transactions containing both A and B is **exactly equal** to the number of transactions containing only B

Such a rule has **no interest**

Interest

If a rule A => B has a **large positive** interest it means that the fraction of A-buyers that **also** buy B is **much larger** than the percentage of all customers buying only B

If a rule A => B has a **large negative** interest it means that people who buy B are **unlikely** to buy also A

Apriori Algorithm

Apriori algorithm [Agrawal and Srikant, 1994]

Main idea

- Identify **frequent individual** items
- Given **all the frequent** itemsets of length K-1, **generate** candidate itemsets of length K
- Prune search containing **infrequent** itemsets
- From itemsets, generate the rules

Apriori Algorithm

Remember: every **subset** of a frequent set is necessarily **itself** a frequent set

Consequently, if an itemset is **not** frequent, then all the **supersets** that contain it are also **not** frequent

Clearly, **not all combinations** of frequent itemsets are frequent as well…

Apriori Algorithm

Frequent Itemset Lattice

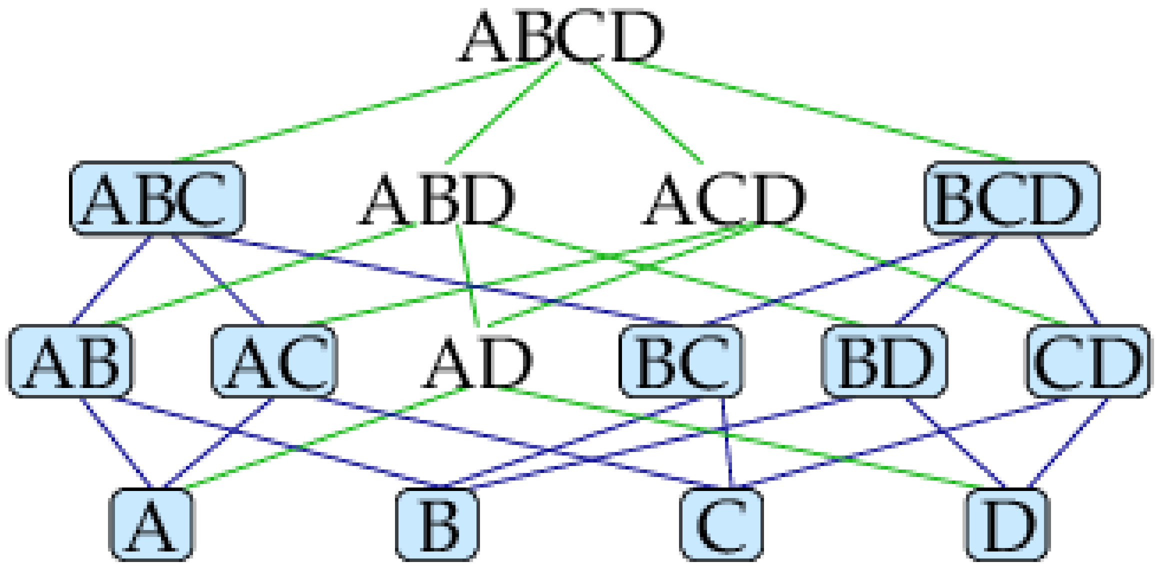

Figure from Li et al. 2003

Apriori Algorithm

```
k = 1
L1 = {frequent items}
while Lk not empty
  Ck+1 = candidates generated from Lk
  for each transaction t in database
    for each candidate c in Ck+1
      If t contains c
          count[c]++
  Lk+1 = candidates in Ck+1 with min_support
  k = k + 1
return union of Lk
```

FOR EXAMPLE
TRIPLETS FROM PAIRS, ETC…
(INCLUDING PRUNING)

IT IS A SET OF ITEMSETS

Apriori Algorithm

Example

- {M,C}, {M,B}, {B,C}, {M,T}, {B,T}
  are frequent itemsets
- {M,B,C}, {M,B,T}
  are candidate frequent itemsets because
  all their subsets are frequent itemsets
- {B,C,T}
  is not a frequent itemset because {C,T} is not either

	M	C	B	H	T
T1	1	1	0	0	0
T2	1	0	1	0	1
T3	1	1	1	0	0
T4	0	0	1	1	1
T5	1	1	1	0	1

Apriori Algorithm

From **frequent itemsets** to **rules**

- There are different possible variations
- One of the easiest is to generate rules A => B from frequent itemsets, **splitting items** between A and B

$$I = \{A, B, C, D\}$$

$$\{A, B\} \Rightarrow \{C, D\}$$
$$\{A, B, D\} \Rightarrow \{C\}$$
$$\{A, B\} \Rightarrow \{C\}$$
$$\dots$$

Extensions

- Sequential or temporal patterns
- Periodical patterns
- Sporadical patterns
- Geospatial patterns
- Multimedial data
- Approximate frequent itemsets

- Process mining

Example with Python

```python
import pandas as pd
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import apriori, association_rules

dataset = [['Milk', 'Cereals'],
           ['Milk', 'Biscuits', 'Tea'],
           ['Milk', 'Cereals', 'Biscuits'],
           ['Biscuits', 'Tea', 'Ham'],
           ['Milk', 'Cereals', 'Biscuits', 'Tea']]

encoder = TransactionEncoder()
transactions = encoder.fit(dataset).transform(dataset)
df = pd.DataFrame(transactions, columns=encoder.columns_)
print(df)
```

Example with Python

```
import pandas as pd
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import apriori, association_rules
import ast

df = pd.read_csv("Groceries.csv", sep=",", index_col='index', converters={'items':
ast.literal_eval})

dataset = df["items"]
encoder = TransactionEncoder()
transactions = encoder.fit(dataset).transform(dataset)
df = pd.DataFrame(transactions, columns=encoder.columns_)

frequent_itemsets = apriori(df, min_support=0.001, use_colnames=True)
rules = association_rules(frequent_itemsets, metric="confidence", min_threshold=0.01)
sorted_rules = rules.sort_values('lift', ascending=False)
print(sorted_rules[["antecedents", "consequents", "lift"]])
```