Machine Learning
Associative rules

1

Association rules

Find interesting **REGULAR PATTERNS** and **RECURRENCES** within a large set of transactions

Some application domains:

- **market basket analysis**
  find recurrent rules relating the purchase of a group of products to the purchase of other products
- **credit card transactions analysis**
  analyze the purchase via credit cards to direct future promotions (product and services are virtually infinite)
- **web mining**
  understand the patterns of navigation paths
- **fraud detection**
  identify potential fraudulent behavior from a set of transactions consisting of incident reports and applications for compensation

2

Association rules: some definitions

Consider a set of n objects (items):

$$\mathcal{O} = \{o_1, o_2, \ldots, o_n\}$$

A **K-ITEMSET** is a generic subset $L \subseteq \mathcal{O}$ containing k objects

A **TRANSACTION** is a generic itemset recorded in a database in a single activity

The data set is composed by a list of m transactions T_i, each associated to a unique identifier t_i

Transaction T contains itemset L if $L \subseteq T$

Data set of transactions

List of transactions

identifier t_i	transaction T_i
001	$\{a, c\}$
002	$\{a, b, d\}$
003	$\{b, d\}$
004	$\{b, d\}$
005	$\{a, b, c\}$
006	$\{b, c\}$
007	$\{a, c\}$
008	$\{a, b, e\}$
009	$\{a, b, c, e\}$
010	$\{a, e\}$

$$\mathcal{O} = \{a, b, c, d, e\} = \{\text{bread, milk, cereals, coffee, tea}\}$$

Data set of transactions

The list of m transactions may be expressed by means of a bi-dimensional matrix:

- columns correspond to objects in set $\mathcal{O}$
- rows correspond to transactions
- the generic elements of the matrix is given by $x_{ij} = \begin{cases} 1 & \text{if } o_j \text{ belongs t} T_i \\ 0 & \text{otherwise} \end{cases}$

Matrix of transactions

identifier t_i	a	b	c	d	e
001	1	0	1	0	0
002	1	1	0	1	0
003	0	1	0	1	0
004	0	1	0	1	0
005	1	1	1	0	0
006	0	1	1	0	0
007	1	0	1	0	0
008	1	1	0	0	1
009	1	1	1	0	1
010	1	0	0	0	1

5

Association rules

The EMPIRICAL FREQUENCY of itemset L is defined as the number of transactions in the data set containing L

When dealing with a large sample, the ratio $f(L)/m$ approximates the probability of occurrence of the itemset L

$\mathcal{O} = \{a, b, c, d, e\} = \{\text{bread, milk, cereals, coffee, tea}\}$

identifier t_i	a	b	c	d	e
001	1	0	1	0	0
002	1	1	0	1	0
003	0	1	0	1	0
004	0	1	0	1	0
005	1	1	1	0	0
006	0	1	1	0	0
007	1	0	1	0	0
008	1	1	0	0	1
009	1	1	1	0	1
010	1	0	0	0	1

$L = \{a, c\} \quad \Rightarrow \quad f(L) = 4 \quad \Rightarrow \quad \Pr(L) \approx 4/10 = 0.4$

6

What is a rule?

Let Y and Z be two propositions which may be true or false

A **RULE** is an implication in the form Y $\Rightarrow$ Z : “ if Y is true, then Z is also true ”

A rule is called **PROBABILISTIC** if the validity of Z is associated with a certain probability p : “ if Y is true, then Z is also true with probability p ”

Consider two disjoint itemsets, $L \subset \mathcal{O}$ and $H \subset \mathcal{O}$, and the transaction T

An **ASSOCIATIVE RULE** is a probabilistic implication denoted as L $\Rightarrow$ H with the following meaning:

“ if L is contained in T , then H is also contained in T with probability p ”

$\downarrow$ body $\qquad$ $\downarrow$ head

7

Confidence and support

CONFIDENCE $\quad p = \mathrm{conf}\{L \Rightarrow H\}$

- proportion of transactions containing H among those including L (reliability of the rule)
- it approximates the conditional probability that H belongs to T given that L belongs to T

SUPPORT $\quad s = \mathrm{supp}\{L \Rightarrow H\}$

- proportion of transactions containing both H and L (frequency of the pair L-H)
- it approximates the probability that L and H are both contained in a future transaction

$\mathcal{O} = \{a, b, c, d, e\} = \{$bread, milk, cereals, coffee, tea$\}$

$L = \{a, c\}$

$H = \{b\}$

identifier t_i	a	b	c	d	e
001	1	0	1	0	0
002	1	1	0	1	0
003	0	1	0	1	0
004	0	1	0	1	0
005	1	1	1	0	0
006	0	1	1	0	0
007	1	0	1	0	0
008	1	1	0	0	1
009	1	1	1	0	1
010	1	0	0	0	1

$p = \mathrm{conf}\{L \Rightarrow H\} =$ $\qquad$ $s = \mathrm{supp}\{L \Rightarrow H\} =$

8

Strong association rules

Objective $\Rightarrow$ find the **STRONG** association rules:

$$s \geq s_{min} \qquad p \geq p_{min}$$

what about extracting all the rules?...

The number of possible rules increases exponentially as the number of objects increases …

$\Downarrow$

… a two-phase procedure is applied:

- generation of **FREQUENT ITEMSETS**
- generation of strong rules

From a set of 20 objects it is possible to generate about 3.5 billions of association rules!

Strong association rules

Strong rules are always meaningful?

A retailer whishes to analyze a set of transactions to identify associations between sales of color printers and sales of digital cameras:

- 1000 transactions
- 600 include digital cameras
- 750 include color printers
- 400 include both products

probability of purchasing a printer conditioned to the purchase of a camera

$\uparrow$

The rule { digital camera } $\Rightarrow$ { color printer } has support 0.4 and confidence 0.66

What is the probability of purchasing a printer?...0.75, which is greater than 0.66!

$\downarrow$

printers and cameras show a negative correlation…
…the purchase of a camera reduces the probability of buying a printer!

Lift index

A third measure of significance is the LIFT :

$$l = \text{lift}\{L \Rightarrow H\} = \frac{\text{conf}\{L \Rightarrow H\}}{\Pr(H)} = \frac{f(L \cup H)}{f(L)f(H)/m}.$$

- the lift is greater than 1: body and head are ***positively associated***
  the rule is effective in predicting the presence of the head in a given transaction
- the lift is lower than 1: body and head are ***negatively associated***
  the rule is less effective than the estimate obtained by the frequency of the head

For the former example:

$f(L \cup H)$ = 400
$f(L)$ = 600
$f(H)$ = 750

$\text{lift}\{L \Rightarrow H\}$ = 400/(600*750/1000) ≈ 0.88

Association rules

The support of the rule L ⇒ H only depends on the set $L \cup H$ given by the union of the itemsets L and H

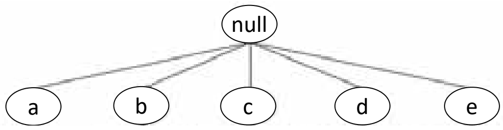
⇓

If the itemset $L \cup H$ is not frequent, we can exclude from the analysis all the rules obtained by using all the proper subsets $L \cup H$ ...

$$\begin{array}{ll} \{a,b\} \Rightarrow \{c\} & \{a,c\} \Rightarrow \{b\} \\ \{b,c\} \Rightarrow \{a\} & \{a\} \Rightarrow \{b,c\} \\ \{b\} \Rightarrow \{a,c\} & \{c\} \Rightarrow \{a,b\} \end{array}$$

... the real problem, therefore, is how to determine the frequent itemsets !

Itemset grid



Association rules

A data set of *m* transactions defined over a set of *n* objects may contain up to 2^n-1 frequent itemsets (excluding the empty set) ...exhaustive enumeration is impracticable...

The APRIORI ALGORITHM is an effective method for extracting strong rules.
It relies on the following property (APRIORI PRINCIPLE):

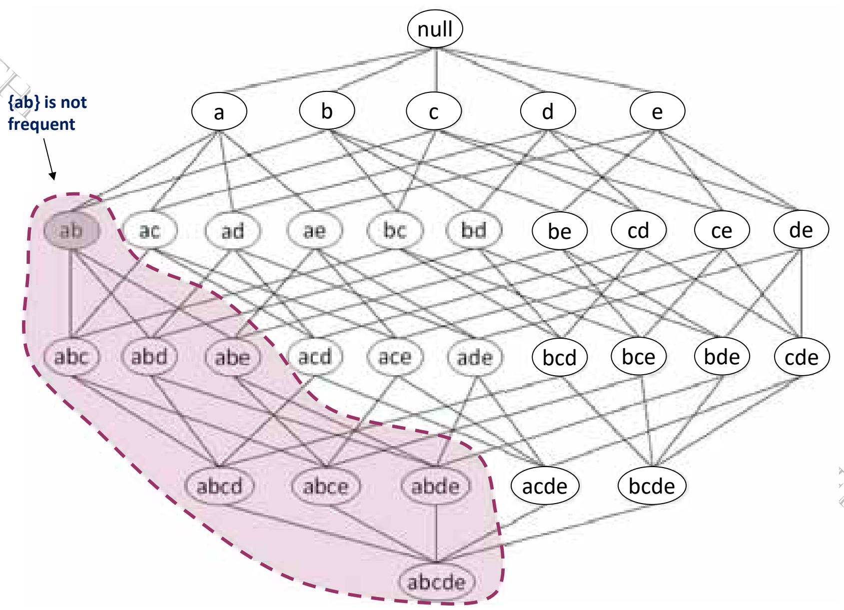
"if an itemset is frequent, then all its subsets are also frequent"

$\Downarrow$

if an itemset is not frequent, then each of the itemset containing it must turn out to be not frequent!

Once a non-frequent itemset is identified in the course of the algorithm, all the other itemsets (with greater cardinality) containing it are implicitly eliminated and excluded from the analysis

Itemset grid



15

Apriori algorithm: phase 1

The Apriori algorithm generates the frequent itemsets iteratively, starting from the frequent 1 – itemset to the frequent k – itemsets

- **Phase 1 - initialization**
  Compute the relative frequency of each object.
  Discard the objects with a frequency smaller than the given minimum threshold.

$\mathcal{O} = \{a, b, c, d, e\}$ = {bread, milk, cereals, coffee, tea}

itemset	relative frequency	status
$\{a\}$	$7/10 = 0.7$	frequent
$\{b\}$	$7/10 = 0.7$	frequent
$\{c\}$	$5/10 = 0.5$	frequent
$\{d\}$	$3/10 = 0.3$	frequent
$\{e\}$	$3/10 = 0.3$	frequent

16

Apriori algorithm: phase 1

- **Phase 1 - iteration 1**
  The candidate 2 – itemsets are obtained from the 1 – itemset found during the former iteration.
  Non-frequent itemsets are discarded.

$\mathcal{O} = \{a, b, c, d, e\}$ = {bread, milk, cereals, coffee, tea}

itemset	relative frequency	status
$\{a, b\}$	$4/10 = 0.4$	frequent
$\{a, c\}$	$4/10 = 0.4$	frequent
$\{a, d\}$	$1/10 = 0.1$	not frequent
$\{a, e\}$	$3/10 = 0.3$	frequent
$\{b, c\}$	$3/10 = 0.3$	frequent
$\{b, d\}$	$3/10 = 0.3$	frequent
$\{b, e\}$	$2/10 = 0.2$	frequent
$\{c, d\}$	$0/10 = 0.0$	not frequent
$\{c, e\}$	$1/10 = 0.1$	not frequent
$\{d, e\}$	$0/10 = 0.0$	not frequent

17

Apriori algorithm: phase 1

- **Phase 1 - iteration 2**
  The candidate 3 – itemsets are obtained from the 2 – itemsets found during the former iteration.
  Non-frequent itemsets are discarded.

$\mathcal{O} = \{a, b, c, d, e\}$ = {bread, milk, cereals, coffee, tea}

itemset	relative frequency	status
$\{a, b, c\}$	$2/10 = 0.2$	frequent
$\{a, b, e\}$	$2/10 = 0.2$	frequent

The first phase of the algorithm terminates since no candidate 4 – itemsets can be generated!

18

Apriori algorithm: phase 2

- **Phase 2 - step 1**
  Split each frequent itemset B into two disjoint non-empty subsets L and H = B – L , according to each possible combination
  Generate all the corresponding candidate rules ⟶ **given a frequent k-itemesets it is possible to generate 2^k-2 candidate rules which may by strong rules (excluding those with empty body or head)**

- **Phase 2 - step 2**
  For each candidate rule L ⇒ H compute the confidence
  ↓
  very fast step – the frequencies of (L∪H) and L are known at the end of the first phase

- **Phase 2 - step 3**
  Include into the list of strong rules all the rules provided with a confidence greater than the prefixed minimum threshold

Apriori algorithm: phase 2

Strong rules generation

itemset	rule	confidence	status
$\{a, b\}$	$\{a \Rightarrow b\}$	$p = 4/7 = 0.57$	strong
$\{a, b\}$	$\{b \Rightarrow a\}$	$p = 4/7 = 0.57$	strong
$\{a, c\}$	$\{a \Rightarrow c\}$	$p = 4/7 = 0.57$	strong
$\{a, c\}$	$\{c \Rightarrow a\}$	$p = 4/5 = 0.80$	strong
$\{a, e\}$	$\{a \Rightarrow e\}$	$p = 3/7 = 0.43$	not strong
$\{a, e\}$	$\{e \Rightarrow a\}$	$p = 3/3 = 1.00$	strong
$\{b, c\}$	$\{b \Rightarrow c\}$	$p = 3/7 = 0.43$	not strong
$\{b, c\}$	$\{c \Rightarrow b\}$	$p = 3/5 = 0.60$	strong
$\{b, d\}$	$\{b \Rightarrow d\}$	$p = 3/7 = 0.43$	not strong
$\{b, d\}$	$\{d \Rightarrow b\}$	$p = 3/3 = 1.00$	strong
$\{b, e\}$	$\{b \Rightarrow e\}$	$p = 2/7 = 0.29$	not strong
$\{b, e\}$	$\{e \Rightarrow b\}$	$p = 2/3 = 0.67$	strong
$\{a, b, c\}$	$\{a, b \Rightarrow c\}$	$p = 2/4 = 0.50$	not strong
$\{a, b, c\}$	$\{c \Rightarrow a, b\}$	$p = 2/5 = 0.40$	not strong
$\{a, b, c\}$	$\{a, c \Rightarrow b\}$	$p = 2/4 = 0.50$	not strong
$\{a, b, c\}$	$\{b \Rightarrow a, c\}$	$p = 2/7 = 0.29$	not strong
$\{a, b, c\}$	$\{b, c \Rightarrow a\}$	$p = 2/3 = 0.67$	strong
$\{a, b, c\}$	$\{a \Rightarrow b, c\}$	$p = 2/7 = 0.29$	not strong
$\{a, b, e\}$	$\{a, b \Rightarrow e\}$	$p = 2/4 = 0.50$	not strong
$\{a, b, e\}$	$\{e \Rightarrow a, b\}$	$p = 2/3 = 0.67$	strong
$\{a, b, e\}$	$\{a, e \Rightarrow b\}$	$p = 2/3 = 0.67$	strong
$\{a, b, e\}$	$\{b \Rightarrow a, e\}$	$p = 2/7 = 0.29$	not strong
$\{a, b, e\}$	$\{b, e \Rightarrow a\}$	$p = 2/2 = 1.00$	strong
$\{a, b, e\}$	$\{a \Rightarrow b, e\}$	$p = 2/7 = 0.29$	not strong

Association rules

The computational effort required grows exponentially as the number *n* of objects increases

To improve the efficiency:

- resort to advanced data structures (dictionaries, binary trees)
- split the data set into disjoint subsets of transactions and apply the algorithm to each subset (local frequent itemsets)
- use the algorithm on a given sample of transactions
- resort to hierarchies of objects
- discard less interesting objects in the preprocessing phase

Association rules

The aforementioned rules are *binary*, *asymmetric* and *one-dimensional* rules

binary ↓ presence vs. absence; *asymmetric* ↓ presence is more relevant

More general association rules:

- rules for binary symmetric variables ({male,female}, {yes,no}, …)
  → two asymmetric binary variables are introduced for each symmetric variable
- rules for categorical variables (education, profession,…)
  → a set of asymmetric binary variables is introduced for each categorical variable
- rules for continuous variables (age, income,…)
  → discretization is first applied to obtain a categorical variable
- multi-dimensional rules
- sequential rules (when transactions are recorded according to a specific temporal sequence)