Statistical Data Mining

Mining Association Rules

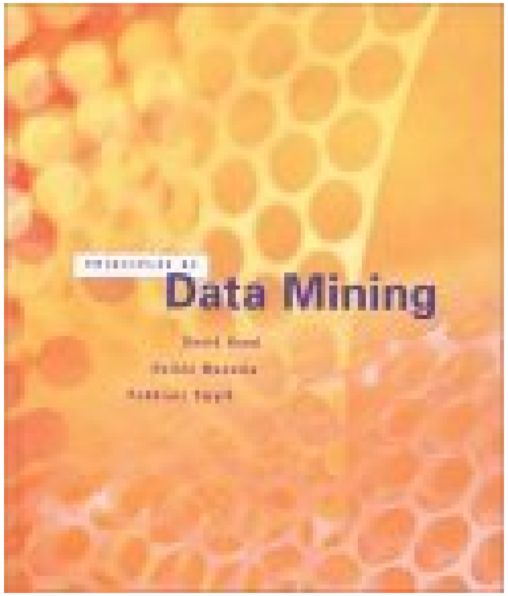
Professor Dr. Gholamreza Nakhaeizadeh

content

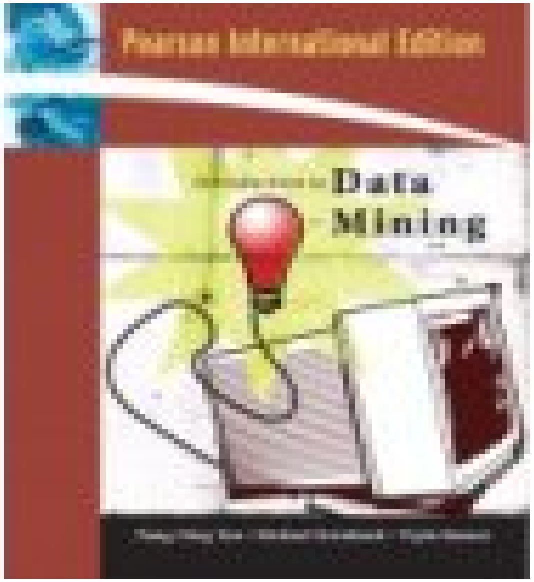
Literature used

- Mining frequent patterns
- Association Rules
- Support and Confidence of an AR-Rule
- AR-Discovery
- Rule Pruning before computing support and confidence
- Frequent itemset generation
- Reduce candidate itemsets
- Apriori-Algorithm

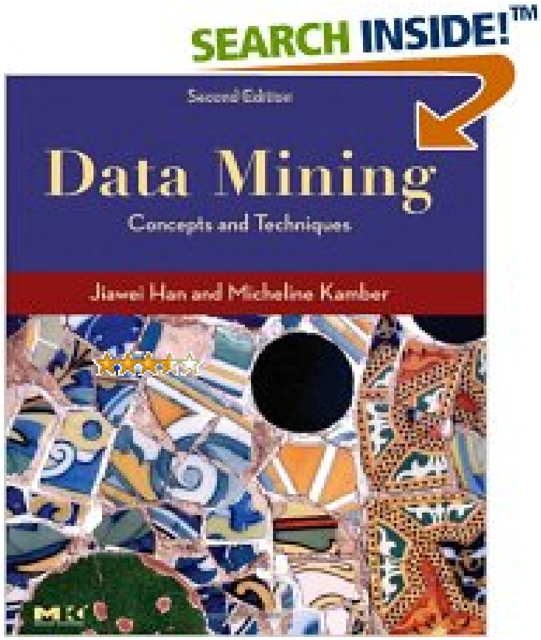
Literatur used (1)



Principles of Data Mining
David J. Hand, Heikki Mannila,
Padhraic Smyth



Pang-Ning Tan,
Michael Steinbach,
Vipin Kumar



Jiawei Han and
Micheline Kamber

Literature Used (2)

http://cse.stanford.edu/class/sophomore-college/projects-00/neural-networks/

http://www.cs.cmu.edu/~awm/tutorials

http://www.crisp-dm.org/CRISPwP-0800.pdf

http://en.wikipedia.org/wiki/Feedforward_neural_network

http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html#Feedback%20networks

http://www.dmreview.com/

http://www.planet-source-code.com/vb/scripts/ShowCode.asp?lngWId=5&txtCodeId=378

http://download-uk.oracle.com/docs/html/B13915_02/i_olap_chapter.htm#BABCBDFA

http://download-uk.oracle.com/docs/html/B13915_02/i_rel_chapter.htm#BABGFCFG

http://training.inet.com/OLAP/home.htm

http://www.doc.gold.ac.uk/~mas01ds/cis338/index.html

http://wwwmaths.anu.edu.au/~steve/pdcn.pdf

www.kdnuggets.com

The Data Warehouse Toolkit by Ralph Kimball (John Wiley and Sons, 1996)

Building the Data Warehouse by William Inmon (John Wiley and Sons, 1996)

Mining Association Rules

Mining Frequent Pattern

Frequent patterns:

{ Milk, eggs, butter } often observed → **Frequent itemset**

{ buying computer , buying printer (after 2 months), buying scanner (after 6 months) }

often observed over the time → **Frequent Sequential pattern**

Frequent itemsets and frequent sequential patterns play a very import role in Mining Association

Famous application: Market Basket Transaction

Example

TID	Items
1	bread, milk
2	bread, meat, orange juice, eggs
3	milk, meat, orange juice, cola
4	bread, milk, meat, orange juice
5	bread, milk, meat, cola

{ Milk } → { meat }

{ Meat } → { Orange juice }

Association rules

The rules show that apparently there is a strong relationship between buying of milk and meat as well meat and orange juice

Mining Association Rules

Association Rules (AR)

Problems in AR-Mining:
- AR-mining from large datasets is pretty time consuming
- mined Associations could be spurious because may happen by chance

Binary representation of market basket data

TID	bread	milk	meat	Orange juice	eggs	cola
1	1	1	0	0	0	0
2	1	0	1	1	1	0
3	0	1	1	1	0	1
4	1	1	1	1	0	0
5	1	1	1	0	0	1
Total	**4**	**4**	**4**	**3**	**1**	**2**

→ ignore quantity,
Price, expiration
date, supplier, ingredient etc.

Notations:

$I = \{ i_1, i_2, \ldots\ i_m \}$ set of all items

$T = \{t_1, t_2, \ldots\ t_N \}$ set of all transactions

t_i contains a subset of items of I

$\{ i_1, i_2,..i_k \}$: k-itemset

Example: { milk, meat, eggs } : 3-ietemset

X: Itemset

$\rho(X)$ = number of transactions contin X

Example: in the table

ρ {bread, milk } = 3 ρ {eggs, cola } = 0

Mining Association Rules

Association Rules (AR) | **Support and Confidence of an AR-Rule**

Definition:
$X \rightarrow Y$ (X is associated to Y) is called an AR
X and Y are disjoint itemsets : $X \cap Y = \emptyset$

Definition (support and confidence of an AR-Rule)

$$\text{Support}, \ s(X \rightarrow Y) = \frac{\rho(X \& Y)}{N}$$

Percentage of the transactions contain both X and Y in the whole transactions

Probability of (X & Y) appear together

$$\text{Confidence}, \ c(X \rightarrow Y) = \frac{\rho(X \& Y)}{\rho(x)}$$

Percentage of the transactions containing both X and Y in the transactions contain X

Conditional probability of Y by given X

Support

0 → 100%

Whole transactions

Confidence

0 → 100%

transactions contain X

Mining Association Rules

Association Rules (AR)

Support and Confidence of an AR-Rule

Example

TID	bread	milk	meat	Orange juice	eggs	cola
1	1	1	0	0	0	0
2	1	0	1	1	1	0
3	0	1	1	1	0	1
4	1	1	1	1	0	0
5	1	1	1	0	0	1
Total	**4**	**4**	**4**	**3**	**1**	**2**

Rule:
{ milk, meat } → {orange Juice }

X → Y

X = { milk, meat } Y= {orange Juice}

ρ (X& Y) = ρ { milk, meat, orange juice } = 2

ρ (X) = ρ { milk, meat } = 3

$$\text{Support , } s(X \rightarrow Y) = \frac{\rho(X \& Y)}{N} = 2/5 = 40\%$$

$$\text{confidence , } c(X \rightarrow Y) = \frac{\rho(X \& Y)}{\rho(X)} = 2/3 = 67\%$$

Rule:
{meat, orange juice } → {eggs}
S= 20%, c= 33%

Rule:
{bread} → {milk}
S = 60% C = 75%

Rule:
{eggs} → {cola}
S = 0% c= 0%

Mining Association Rules

Association Rules (AR) **AR-Discovery**

Definition:

Given: a set of transactions T **Find: Association Rules having :**

Support ≥ sup_min and Confidence ≥ conf_min

***sup_min*: given support threshold *conf_min* : given confidence threshold**

Methods of AR-Mining

Brute-force approach: calculate support and confidence for every possible rules
Problem: many many rules
A dataset with 10 items would generate 57000 rules;
a department store could have more than 10.000 items

Mining Association Rules

Association Rules (AR)

AR-Discovery

Rule Pruning before computing support and confidence

Example: Consider the itemset

{ orange juice, meat, milk }

the following AR-Rules involve the same Itemset:

{ orange juice, meat } → { milk }
{ orange juice, milk } → { meat }
{ meat, milk } → {orange juice}
{orange juice → { meat, milk }
{ milk } → { orange juice, meat }
{ meat } →{ orange juice, milk }

TID	bread	milk	meat	orange juice	eggs	cola
1	1	1	0	0	0	0
2	1	0	1	1	1	0
3	0	1	1	1	0	1
4	1	1	1	1	0	0
5	1	1	1	0	0	1
Total	**4**	**4**	**4**	**3**	**1**	**2**

→ Have the same Support 40%

It means : if we define a *sup_min* of e. g. 50% , after calculating the support of the first rule (40%) we see that we can prune all the others rule before we calculate their support and confidence

Mining Association Rules

Association Rules (AR) **AR-Discovery**

Viewing the AR-Mining as a two steps Process:
(adopted by many AR-Mining algorithms)

1. **Frequent Itemset Generation (FIG)**
2. **Rule Generation**

The aim of FIG is to find all itemsets with support ≥ *sup_min*
Such itemsets called *frequent itemsets* (sometimes large itemsets)

The aim of Rule Generation is to extract from frequent itemsets the rules with Confidence ≥ *conf_min*; such rules are called strong rules

In the past years a lot of attempts put to find efficient methods for generating the frequent itemsets

Mining Association Rules

Association Rules (AR) | AR-Discovery | Frequent itemset generation

Candidate Itemset

Generally for an itemset with n items, potentially $2^n - 1$ candidate itemsets can be generated

Example

Consider itemset { a, b, c, d, e }

n=5 number of candidat itemsets = 31

a b c d e

ab ac ad ae bc bd be cd ce de

abc abd abe acd ace ade bcd bce bde cde

abcd abce abde acde bcde

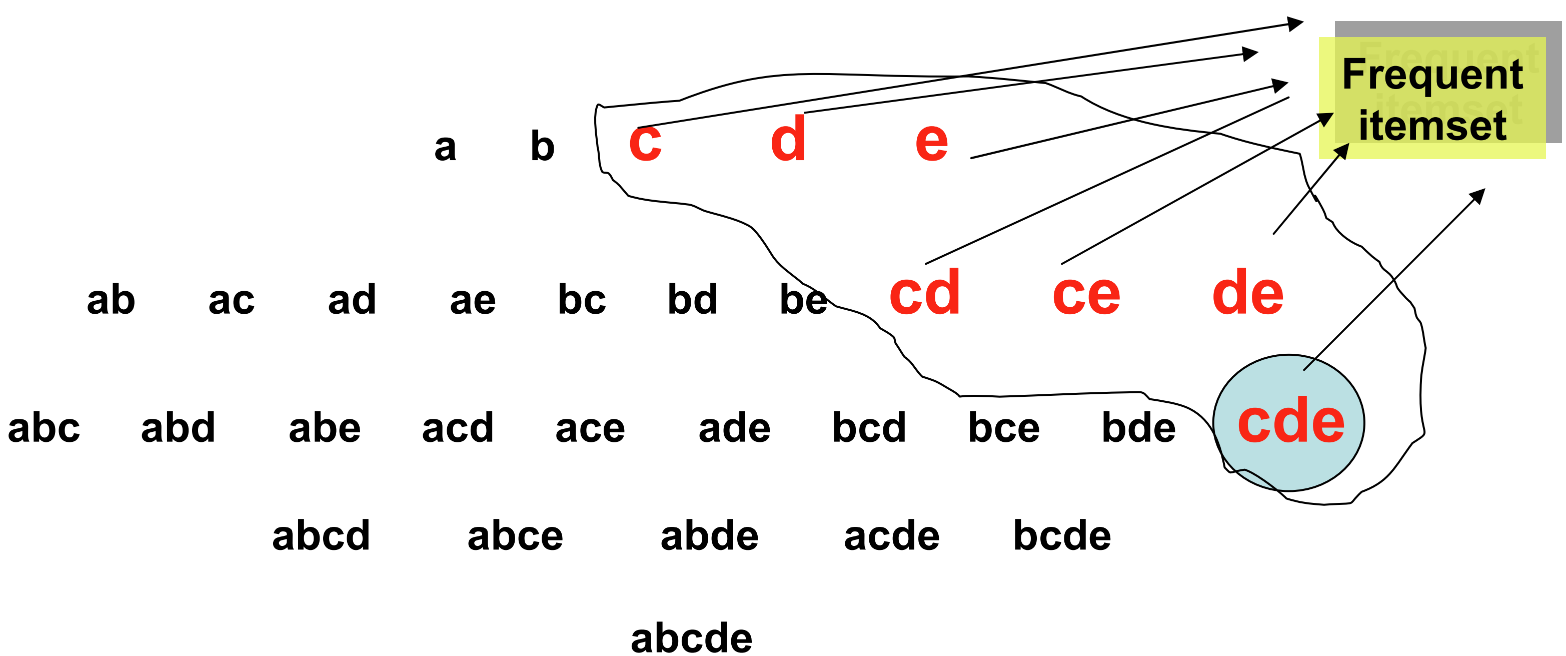
abcde

Mining Association Rules

Association Rules (AR) | AR-Discovery | Reduce candidate itemsets

Apriori – Principal (1)

- All of the subsets of a frequent itemset must be frequent itemsets too



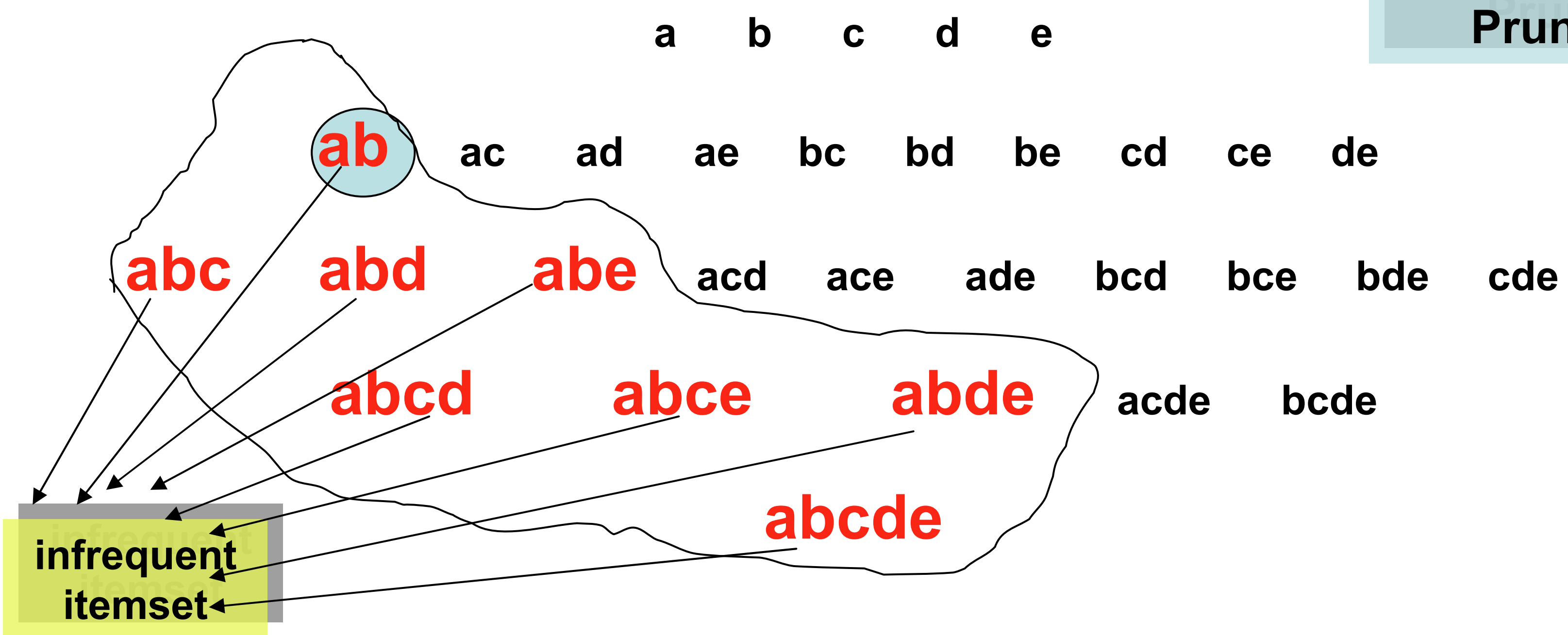
Mining Association Rules

Association Rules (AR) | AR-Discovery | Reduce candidate itemsets

Apriori – Principal (2)

- All of the supersets of an infrequent itemset must be infrequent itemsets too

Support-based Pruning



Mining Association Rules

Association Rules (AR) | **AR-Discovery** | **Apriori-Algorithm (AA)**

Frequent itemset generation in AA

Item	Count
Orange juice	3
bread	4
cola	2
meat	4
milk	4
eggs	1

Itemset	Count
{orange juice, bread}	2
{orange juice, meat}	3
{orange juice, milk}	2
{bread, meat}	3
{bread, milk}	3
{meat, milk}	3

Itemset	Count
{bread, meat, milk}	2

Example

TID	Items
1	bread, milk
2	bread, meat, orange juice, eggs
3	milk, meat, orange juice, cola
4	bread, milk, meat, orange juice
5	bread, milk, meat, cola

Given: sup_min = 60% ~ min support count = 3

Candidate itemsets (up to size 3)

Brute-force strategy: $\binom{6}{1} + \binom{6}{2} + \binom{6}{3} = 6 + 15 + 20 = 41$

Apriori principal: $\binom{6}{1} + \binom{4}{2} + 0 = 6 + 6 + 0 = 12$

Mining Association Rules

Association Rules (AR) | AR-Discovery | Apriori-Algorithm (AA)

Rule generation in AA

$$\text{Conf}(X \rightarrow Y) = \frac{\rho(X \,\&\, Y)}{\rho(x)} = \frac{N * \text{Support}(X \,\&\, Y)}{N * \text{Support}(X)} = \frac{\text{Support}(X \,\&\, Y)}{\text{Support}(X)}$$

For each frequent itemset f, generate all non-empty subsets of f
For every non-empty subset s of f
Generate rule s → (f – s) if support (f) / support (s) ≥ *conf_min*

Notes:
1- Rule generation in AA is less computing time consuming as frequent itemsets generation, because the needed supports are already calculated

Mining Association Rules

Association Rules (AR) | AR-Discovery | Apriori-Algorithm (AA)

Rule generation in AA

Example: Given: conf_min = 80%

Item	Count
orange juice	3
bread	4
meat	4
milk	4

Itemset	Count
{orange juice, meat}	3
{bread, milk}	3
{bread, meat}	3
{meat, milk}	3

Itemset	Count
{meat, orange juice}	3

We consider the frequent itemset

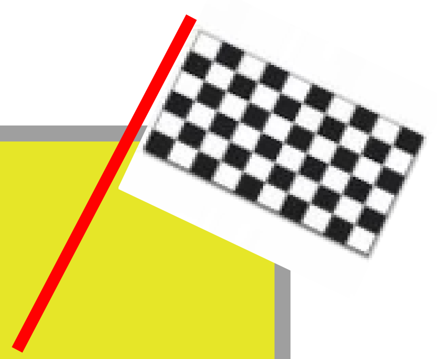
Conf of { meat } → { orange juice} = 3/4 = 75%

Conf of { orange juice } → { meat } =3/3 = 100%

Generated AR from the { meat, orange juice }

{ orange juice} → { meat }

Short Reveiew



Mining frequent patterns

- Association Rules
- Support and Confidence of an AR-Rule
- AR-Discovery
- Rule Pruning before computing support and confidence
- Frequent itemset generation
- Reduce candidate itemsets
- Apriori-Algorithm

- Clementine Demo
- Basklinks_association.str