Chapter 2
Getting to your Data:

Basic Statistical Structure of Data:

3 types/ or read

→ measure of central tendency — measure: location of the middle or center of a data distribution
- → mean
- → median
- → mode
- → midrange

→ dispersion of data
- → range
- → quartile
- → inter quartile range (IQR)
- → five number summery
- → boxplots
- variance
- Standard deviation

→ graphic displays of basic Statistical description
- quantile plot
- quantile quantile plot
- histogram
- Scatter plot

Mean/arithmetic mean

Let $x_1, x_2 \ldots x_n$ be a set of observations.

the mean of this set of values,

$$\bar{x} = \frac{\sum_{i=1}^{N} x_i}{N}$$

$$= \frac{x_1 + x_2 + \cdots + x_N}{N}$$

→ corresponds to the built in aggregate function, average (avg())

Weighted arithmetic mean / weated average:

→ each value x_i associated with a weight w_i for $i = 1, 2 \ldots N$.

→ weight → reflects the
- → significance
- → importance
- → occurrence
- frequency

attached to their respective

$$\bar{x} = \frac{\sum_{i=1}^{N} w_i x_i}{\sum_{i=1}^{N} w_i}$$

$$\therefore \frac{w_1 x_1 + w_2 x_2 + \cdots + w_N x_N}{w_1 + w_2 + \cdots + w_N}$$

Problem of mean:

→ Sensitivity to extreme/ outr values

outlier

Small number of extreme values can corrupt the mean

Sol^n : trimmed mean

↳ mean obtained after chopping off values at the high & low extreme.

→ remove top & bottom 2%

Median:

→ expensive to compute
→ large number of observation

→ better for skewed data

middle value in a set of ordered data values.

~~→ N is odd~~

⇒ N is odd → middle value of the ordered set

⇒ N is even → two middlemost values & any values in between
→ conventionally average of the two middlemost value

→ .

For grouped data

→ data are grouped in intervals in a.t. their
→ data values
→ frequency

$$\text{median} = L_1 + \left(\frac{N/2 - (\Sigma f_{req})_1}{f_{req\,median}} \right) \text{width}.$$

where,

L_1 = Lower boundary

$(\Sigma f_{req})_1$ = sum of the frequencies of intervals that are lower than median

f_{median} = frequency of the median.

width = weidth of the median interval.

Mode:

Mode for a set of dat is

value that occurs most frequently in the set.

Unimodal: Data set with one mode

bimodal: " " two modes

trimodal: " " three "

multimodal: " " with two or more modes

Relation among mean, mode, median: moderately skewed → for unimodal numeric data

mean - mode ≃ 3 (mean - median)

Midrange:

Average of the largest and smallest values in the set.

Skewed data: / asymmetric data

2 types

→ positively skewed

→ negatively "

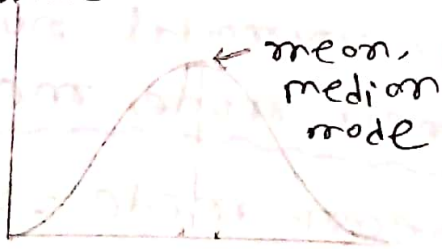
→ mode occurs at a value, smaller than the median (mode < median < mean)

→ mode occurs at a " greater " the median.

(mean < median < mode)

Symmetric data:

mean = median = mode



Measuring the Dispersion of Data:

- Range
- Quartile
- Quartiles
- percentiles
- interquartile range

Range:

Difference between the largest and smallest values.

$\therefore$ Range = max() - min()

Quartile:

Points taken at regular intervals of data distribution.

dividing it into essentially equal size consecutive sets.

★ k-th q-quantile for a given distribution is the value x

↳ such that
- → at most k/q data values $< x$
- → " " $(q-k)/q$ " " $> x$

→ k is an integer $\boxed{0 < k < q}$

→ $q-1$ q-quantiles

Quartiles:

3 in 4- quantiles each of the four distributed part of data is known as quartile:

Percentiles:

100- quantiles are commonly referred to as percentiles.

→ divide the data distribution into 100 equal-sized consecutive data.

Interquartile range (IQR)

Distance between the first and third quartiles is called IQR and defined as,

$$IQR = Q_3 - Q_1$$

→ 1st quartile (Q_1) → cut off the lowest 25% of the data

→ 2nd " (Q_2) → it gives the center of the data distribution

→ 3rd " (Q_3) → cut of the lowest 75% of data

Rule of thumb

→ used for identifying suspected outliers

values falling at least $1.5 \times IQR$

above the 3rd quartile

below the first "

Five number Summary:

Consists of → median (Q_2)
→ the quartiles (Q_1 & Q_3)
→ Smallest and largest individual observation

→ written in the order of
Minimum, Q_1, Median, Q_3, Maximum.

Boxplot → proper way of visualizing a distribution.

↳ can be computed in $O(n \log n)$ time

→ ends of the box are at quartiles
↓
box length is the IQR

→ median is marked by
↓
a line within the box

→ 2 lines outside the box extend to th

(whiskers)

→ smallest / minimu

→ & largest / maxim um

→ ~~dot~~ → extended to the extreme low and high observations only if

values are less than $1.5 \times IQR$

→ outliers are indicated by dot (··)

Variance:

Variance of N observations.

$x_1, x_2 \ldots x_N$ for numeric attribute X is,

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2$$

$$= \left(\frac{1}{N}\sum_{i=1}^{N}x_i^2\right) - \bar{x}^2$$

Standard deviation:

Square root of the variance (σ^2)

→ denoted by σ

~~Probes~~

Basic properties of Standard deviation (σ)

→ measures spread about the mean.

→ considered only when mean is chosen as the measure of

→ $\sigma = 0$. → there is no spread
→ all observations have the same value.

→ otherwise $\sigma > 0$

Chebyshev's inequality:

**

Shown that (at least $(1 - \frac{1}{k^2}) \times 100\%$ of the observations are no more than k standard deviations from the mean.

Chapter-6

Mining Frequent Patterns, Associations and Correlations: Basic Concepts and Methods

→ Frequent itemset mining leads to

discovery of association & correlations

among items in large transactional / relational datasets

→ discovery of interesting correlation relationships

can help in

- → catalog design
- → cross-marketing
- → customer shopping behavior analysis

Example of frequent pattern mining:

market basket analysis

→ help retailers to develop

marketing strategies

→ by gaining insight into which items are frequently purchased together by customer

Item set:

A set of items referred to as an itemset

→ denoted by $I = \{I_1, I_2 \dots I_m\}$

k-itemset:

An itemset that contain k-items

$$X = \{x_1, x_2 \dots x_k\}$$

- Occurrence frequency of an itemset / frequency / support count / count:

  Number of transactions that contain the itemset.

  → also called absolute support

- relative support: / support $(A \Rightarrow B)$

  $$\text{Support}(A \Rightarrow B) = P(A \cup B)$$

  probability that a transaction contains $A \cup B$

- Confidence $(A \Rightarrow B)$:

  $$= \frac{\text{support}(A \cup B)}{\text{support}(B)}$$

  $$\text{confidence}(A \Rightarrow B) = P(B|A)$$

  conditional-probability that a transaction having A also contains B

Frequent itemset:

A itemset J is frequent if J's support is no less than minimum support threshold.

i.e support(J) >= minsup(J)

Strong:

Rule that satisfy both

→ a minimum support threshold

→ and a " confidence "

Association rule:

It is an implication of the form $A \Rightarrow B$, where $A \subset I$, $B \subset I$, $A \neq \phi$, $B \neq \phi$ and $A \cap B = \phi$

Process of association rule mining:

two step process:

→ find all frequent itemsets

→ generate strong association rules from the frequent itemset.

→ rule must satisfy
- minimum support
- " confidence

Closed itemset:

An itemset X is closed in a data set D if there exists no proper super itemset Y such that sup_count(Y) = sup_count(X)

Closed frequent itemset:

An itemset X is a closed frequent itemset in set D if

→ X is both closed and frequent in D

maximal frequent itemset (or max-itemset):

- → X is frequent
- → no super-itemset Y
  - such that
    - X ⊂ Y
- → Y is frequent

proper superitemset:

Y is a proper super-itemset of X, if X is a proper sub-itemset of Y. i.e

X ⊂ Y

- → every item of X is contained in Y.
- → at least one item of Y that is not in X

Algorithm to find frequent itemset:

1994 → Apriori → R. Agrawal, R. Srikant

→ FP-growth → Han, Pei & Yin

→ Vertical data format approach

→ for mining frequent itemset for Boolean association rules

Apriori property / Down ward closure property

All non empty subsets of a frequent itemset must also be frequent.

Apriori pruning principle:

If there is any itemset which is infrequent,

its superset should not be generate tested

Steps of Apriori algorithm:

2 steps processes

→ The join step

→ The prune step

The join step:

→ find set of candidate k-itemset → denoted by C_k

→ C_k is generated by joining L_{k-1} with itself

The prune step:

→ scan C_k & compare with min_sup

→ determine L_k → frequent k itemse

→ ~~to rc~~

→ Apriori property is used

↳ to reduce the size of C_k.

#Subset testing:

→ Any (k-1) -itemset that is not frequent, can't be subset of a frequent k-itemset.

→ Any (k-1) subset of C_k is not in L_{k-1}, then candidate can't be frequent.

↳ so can be removed from C_k

Frequent itemset → Association Rules:

→ for each frequent itemset l,
generate all nonempty subset of l

→ for every nonempty subset s of l,
output the rule "$s \Rightarrow (l - s)$"

if $\frac{\text{support\_count}(l)}{\text{support\_count}(s)} \geq \text{min\_conf}$

Example-6.3,6.4

Given that,

TID	List of item_IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

min-sup = 2

min-conf = 70%

Step-1

① Scan D for count of each candidates

C_1

Itemset	Sup_count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

Step 2

Compare candidate support with minimum support count

L_1

Itemset	Sup-count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

Step-3

Generate C_2 from L_1 and scan D for count of each. $C_2 = L_1 \bowtie L_1$

C_2

Itemset	Sup-count
{I1, I2}	4
{I1, I3}	4
{I1, I4}	1
{I1, I5}	2
{I2, I3}	4
{I2, I4}	1
{I2, I5}	2
{I3, I4}	0

Itemset	Sup-count
I3, I5	1
I4, I5	0

Step-4

Compare candidate count with minimum count.

L_2

Itemset	Sup. count
{I1, I2}	4
{I1, I3}	4
{I1, I5}	2
{I2, I3}	4
{I2, I4}	2
{I2, I5}	2

Step-5

Generate C_3 from L_2 and scan D for each count

C_3

Itemset	Sup-count
{I1, I2, I3}	2
{I1, I2, I5}	2
~~{I1, I3, I4}~~	
{I1, I3, I5}	–
~~{I1, I2, I4}~~	–
{I2, I3, I4}	–
{I2, I3, I5}	–

Itemset	SUP-count
{I2, I4, I5}	-

But According to apriori property

Last 4 sets are not be a member of frequent itemset.

Step-6

Generate L_3 from C_3

L_3

itemset	SUP-COUNT
{I1, I2, I3}	2
{I1, I2, I5}	2

And

Step 7:

Now $C_4 = L_3 \bowtie L_3$

$= \{I1, I2, I3, I5\}$

But it's subset {I1, I3, I5} is not frequent

So it is not frequent.

∴ Frequent itemset.

{{I1, I2, I3}, {I1, I2, I5}}

Ans

Generating association rule:

For frequent itemset {I1, I2, I3}

subsets are {I1}, {I2}, {I3},
{I1, I2}, {I1, I3}, {I2, I3}

∴ {I1, I2} → {I3}, Conf = $\frac{\text{sup-count}\{I1, I2, I3\}}{\text{sup-count}\{I1, I2\}}$

$= \frac{2}{4}$

$= 50\%$

{I1, I3} → {I2}, $\frac{\text{sup-count}\{I1, I2, I3\}}{\text{sup-count}\{I1, I3\}}$

$= \frac{2}{4} = 50\%$

$\{I2, I3\} \rightarrow \{I1\}$, $\frac{\text{sup\_count}\{I1, I2, I3\}}{\text{sup\_count}\{I2, I3\}}$

$= \frac{2}{4}$

$= 50\%$

$\{I1\} \rightarrow \{I2, I3\}$, $\frac{\text{sup\_count}\{I1, I2, I3\}}{\text{sup count}\{I1\}}$

$= \frac{2}{6}$

$= 33.3\%$

$\{I2\} \rightarrow \{I1, I3\}$:, $\frac{\text{sup\_count}\{I1, I2, I3\}}{\text{sup\_count}\{I2\}}$

$= \frac{2}{7}$

$= 28.57\%$

$\{I3\} \rightarrow \{I1, I2\}$, $\frac{\text{Sup\_count}\{I1, I2, I3\}}{\text{sup\_count}\{I3\}}$

$= \frac{2}{6}$

$= 33.33\%$

As minimum confidence is 70%

∴ there is no association rule

For frequent itemset {I1, I2, I5}.

Subsets are, {I1}, {I2}, {I5},

{I1, I2} {I1, I5} {I2, I5}

$\{I1, I2\} \rightarrow \{I5\}$: $\frac{\text{Sup. count } \{I1, I2, I5\}}{\text{Sup\_count } \{I1, I2\}}$

$= \frac{2}{4} = 50\%$

$\{I1, I5\} \rightarrow \{I2\}$ Confidence, $= \frac{2}{2}$

$= 100\%$

$\{I2, I5\} \rightarrow \{I1\}$ Confidence, $= \frac{2}{2}$

$= 100\%$

$\{I1\} \rightarrow \{I2, I5\}$ confidence, $= \frac{2}{6}$

$= 33.3\%$

$\{I2\} \rightarrow \{I1, I5\}$ confidence, $= \frac{2}{7}$

$= 28.57\%$

$\{I5\} \rightarrow \{I1, I2\}$ confidence, $= \frac{2}{2}$

$= 100\%$

As minimum confidence threshold is 7%

∴ association rules are,

{I1, I5} → {I2}
{I2, I5} → {I1}
{I5} → {I1, I2}

AND

2017-1(a)

Given that, D, |D| = 4

Transaction ID	itemset
T_1	{K, A, D, B}
T_2	{D, A, CE, B}
T_3	{C, A, B, E}
T_4	{B, A, D}

min_sup = 60%
min_conf = 80%

∴ min_sup = count = $4 \times \frac{60}{100}$ =

Step-1,

C_1

Itemset	Sup_count	%Sup_count
{A}	4	100%
{B}	4	100%
{C}	2	50%
{D}	3	75%
{E}	2	50%
{K}	1	25%

Step-2

L_1, comparing with minimum support count

itemset	Sup_count
{A}	4
{B}	4
{D}	3

Step-3

$L_1 \rightarrow C_2$

$C_2 = L_1 \bowtie L_1$

itemset	SUP-count	%SUP-count
{A,B}	4	100%
{A,D}	3	75%
{B,D}	3	75%

Step-4

L_2, comparing ϕ with minimum support

itemset	SUP-count
{A,B}	4
{A,D}	3
{B,D}	3

Step-5

$C_3 = L_2 \bowtie L_2$

itemset	SUP-COUNT	% SUP-COUNT
{A, B, D}	3	75%

∴ Step-6

L_3, comparing with minimum support

itemset	SUP-COUNT
{A, B, D}	3

$\therefore C_4 = L_3 \bowtie L_3$

$= \{A, B, D\} = L_3$

$\therefore$ Frequent itemset is $\{A, B, D\}$

generating association rule

frequent itemset is {A, B, D}

∴ subsets are

{A}, {B}, {D}, {A,B}, {A,D}
{B,D}

∴ {A,B} → {D} . confidence = $\frac{3}{4}$ = 75%

{A,D} → {B} , " = $\frac{3}{3}$ = 100%

{B,D} → {A} , " = $\frac{3}{3}$ = 100%

{A} → {B,D} , " = $\frac{3}{4}$ = 75%

{B} → {A,D} , " = $\frac{3}{4}$ = 75%

{D} → {A,B} , " = $\frac{3}{3}$ = 100%

. As minimum confidence threshold is 80%

∴ Association rules are,

{A,D} → {B}
{B,D} → {A}
{D} → {A,B}

And

Antimonotonicity property:

If a set cannot pass a test, all of its supersets will fail the same test as well.

Finding interesting pattern:

→ support and confidence measure are insufficient

↳ at filtering out uninteresting association rule

Q) - Sol^n: correlation measure can be used.

Correlation rule

It is measured not only by its support & confidence but also by the correlation between itemset A and B.

2 types of correlation analysis

→ lift

→ χ^2

lift :

Lift between occurrence of A and B is,

$$lift(A,B) = \frac{P(A \cup B)}{P(A)\,P(B)}$$

$$= \frac{conf(A \Rightarrow B)}{sup(B)}$$

→ lift (A, B) < 1

occurrence of A is negatively correlated with " of B

→ lift (A, B) > 1 " of A is positively " with " of B

→ lift (A, B) = 1 , ★ A and B are independent
★ there is no correlation

χ^2 testing:

$$\chi^2 = \sum_{i=1}^{c} \sum_{j=1}^{r} \frac{(O_{ij} - e_{ij})^2}{e_{ij}}$$

where,

e_{ij} = expected frequency of (A_i, B_j)

O_{ij} = observed " of (A_i, B_j)

$$c_{ij} = \frac{\text{count}(A = a_i) \times \text{count}(B = b_j)}{n}$$

↳ number of data tuples

Problem of Apriori

two nontrivial costs:

→ It may still need to generate a huge number of candidate sets

Ex 10^4 frequent 1-itemset → 10^7 candidate-2 item

→ need to repeatedly scan the whole database

→ Check a large set of candidates

by pattern mining

Soln:

→ Frequent pattern growth (FP-growth)

FP growth:

→ it adopts a divide-and-conquer strategy.

Steps:

→ Scan DB once, find frequent-1 itemset

→ sort frequent itemset in frequency decreasing order

→ scan DB again, construct FP-tree

→ construct conditional pattern base
→ " conditional FP-tree
→ " frequent pattern

Example 6.5

Given that,

TID	List of item_IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

min_sup = 2

Step-1

item	sup_count
I1	6
I2	7
I3	6

item	sup_count
I1	2
I5	2

As min sup = 2, so preceding table contain all frequent-1 itemset.

Step-2

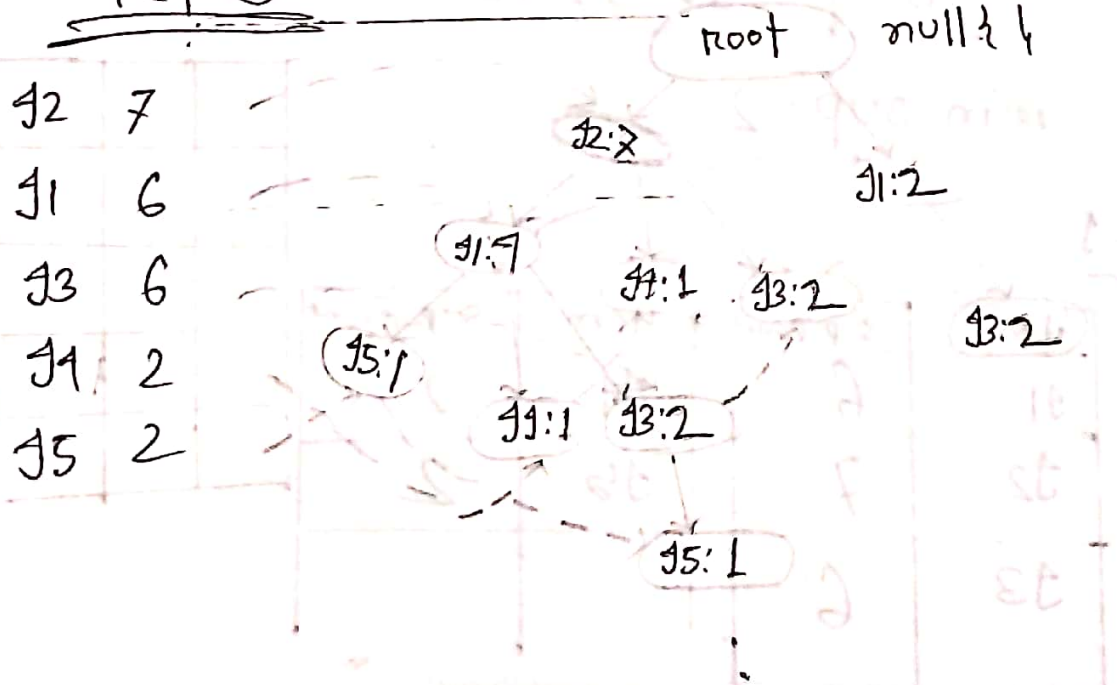
Sorted frequent-1 itemset in decreasing order

item	Sup. count
I2	7
I1	6
I3	6
I4	2
I5	2

pattern :

I2, I1, I3, I4, I5

Step-3



Step-1

item	conditional pattern Base	Conditional FP-tree	Frequent patterns Generated
I5	{{I2, I1 : 1}, {I2, I1, I3 : 1}}	{I2 : 2, I1 : 2}	{I2, I5 : 2}, {I1, I5 : 2}, {I2, I1, I5 : 2}
I4	{{I2, I1 : 1}, {I2 : 1}}	{I2 : 2}	{I2, I4 : 2}
I3	{{I2 I1 : 2}, {I2 : 2}, {I1 : 2}}	{I2 : 4, I1 : 2}, {I1 : 2}	{I2, I3 : 4}, {I1, I3 : 4}, {I2, I1, I3 : 2}
I1	{I2 : 4}	{I2 : 4}	{I2, I1 : 4}

2017 - 3(a)

Given that,

TID	Items_bought	
100	I6, I1, I3	I1, I3
200	I1, I2, I4, I5, I3	I1, I2, I5, I3
300	I3, I2, I5	I2, I5, I3
400	I8, I7	
500	I1, I3, I2, I4, I5	I1, I2, I5, I3
600	I1, I3, I6	I1, I3
700	I1, I2, I5, I7	I1, I2, I5
800	I2, I8, I5, I1	I1, I2, I5
900	I4, I6	
1000	I1, I2, I5	I1, I2, I5

min_support = 40%

Step-1 :

min_support = 40%

min_sup_threshold = $\frac{40}{100} \times 10$

$= 4$

item	sup_count
I1	7
I2	6
I3	5
I4	3
I5	6
I6	3
I7	2
I8	2

Step-2

As min_supp_threshold = 4

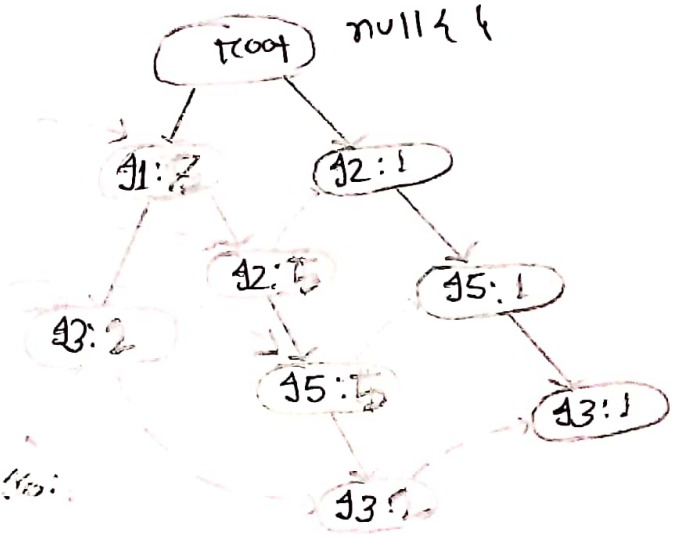
∴ frequent-1 itemset is given below:

item	sup_count
I1	7
I2	6
I5	6
I3	5

pattern: I1, I2, I5, I3

Step-3 :

item	sup_count	node_link
I1	7	
I2	6	
I5	6	
I3	5	



Step-1 CPB: Conditional Pattern Base
CFPT: " FP Tree

Item	CPB	CFPT	Frequent pattern
I3	{{I1:2}, {I1, I2, I5:2}, {I2, I5:1}}	{I1:4}	{I1, I3:4}
I5	{{I1, I2:5}, {I2:1}}	~~{I1:6}~~ {I1:5, I2:5}	{I1, I5:5} {I2, I5:5} {I1, I2, I5:5}
I2	{I1:5}	{I1:5}	~~{I1, I5:5}~~ {I1, I2:5}

2017- 1(b)

Given that.

TID	Items Bought
1	{Milk, Beer, Diapers}
2	{Bread, Butter, Milk}
3	{Milk, Diapers, Cookies}
4	{Bread, Butter, Cookies}
5	{Beer, Cookies, Diapers}
6	{Milk, Diapers, Bread, Butter}
7	{Bread, Butter Diapers}
8	{Beer, Diapers}
9	{Milk, Diapers, Bread, Butter}
10	{Beer, Cookies}

minsup > 0

Let, Milk = Mi
Beer = Be
Diapers = Di
Butter = Bu
Cookies = Co
Bread = Br

(a)

i. Length of transaction ID 4 & 6 and 9 is 4.

∴ So maximum size of frequent itemsets is = 4 Ans

ii. Maximum number of size 3 itemsets.

is = $^6C_3$ =

= 20 Ans

iii. Let C_1

item	Sup. count
Milk	5
Beer	4
Butter	5
Bread	5
Diaper	7
Cookies	4

As minsup > 0

∴ $C_1 = L_1$

(b)
Now make pair

a, b	~~conf(a→b)~~ sup(a)	~~conf(b→a)~~ sup(b)
Milk, Beer	4	5
Milk, Butter ✓	5	5
Milk, Bread ✓	5	5
Milk, Diapers	7	5
Milk, Cookies	4	5
Beer, Butter	5	4
Beer, Bread	5	4
Beer, Diapers	7	4
Beer Cookies ✓	4	4
Butter, Bread ✓	5	5
Butter, Diapers	7	5
Butter Cookies	4	5
Bread, Diaper	7	5
Bread Cookies	4	5
Diapers, Cookies	4	7

∴ (Milk, Butter) ~~are~~ has some confidence in ~~a→b~~ Milk → Butter & and Butter → Milk Ans

NB: Those pairs have same support count they have same confidence in ~~bot~~ a→b and b→a rules.

Finding support of size-2 itemsets

itemset	sup-count
Milk, Bread	3
Milk, Butter	3
~~Bitk~~ Milk, Beer	1
Milk, Cookies	1
Milk, Diapers	4
Bread, Butter	5 ✓
Bread, Beer	0
Bread, Cookies	1
Bread, Diaper	3
Butter, Beer	0
Butter, Cookies	1
Butter, Diaper	3
Beer, Cookies	2
Beer, Diaper	3
Cookies, Diapers	2

∴ Size-2 itemset with largest itemset is (Bread, Butter) Ans

Exercise 6.6

Given that,

TID	items-bought
T100	{M, O, N, K, E, Y}
T200	{D, O, N, K, E, Y}
T300	{M, A, K, E}
T400	{M, U, C, K, Y}
T500	{C, O, K, I, E}

min-sup = 60%

min-conf = 80%

a) Step-1

item	sup-count
A	1
C	2
D	1
E	4
I	1
K	5
M	3
N	2
O	3
U	1
Y	3

Step-2

As min-sup = 60% = $\frac{60}{100} * 5 = 3$

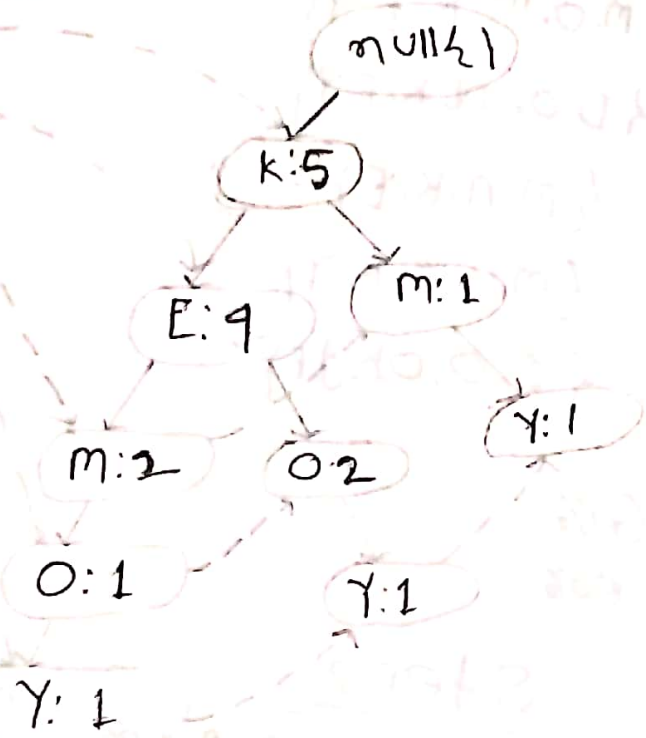
∴ Frequent-1 items is in decreasing order

item	sup-count
K	5
E	4
M	3
O	3
Y	3

∴ pattern: {K, E, M, O, Y}

Step-3:

item	Sup.count	node link
K	5	
E	4	
M	3	
O	3	
Y	3	



Step-4

Item	CPB	CFPT	Frequent pattern
Y	{{K, E, M, O:1}, {K, E, O:1}, {K, M:1}}	{K:3}	{K, Y:3}
O	{{K, E, M:1}, {K, E:2}}	{K:3, E:3}	{K, O:3} {E, O:3} {K, E, O:3}
M	{{K, E:2}, {K:1}}	{K:3}	{K, M:3}
E	{K:4}	{K:4}	{K, E:4}

∴ complete set of frequent item-sets

{{K:5}, {E:4}, {M:3}, {O:3}, {Y:3},
{K,Y:3}, {K,O:3}, {E,O:3}, {K,M:3}, {K,E:4},
{K,E,O:3}}

AND

Apriori Vs FP growth

Apriori	FP-growth
1) An array based algorithm	1) a tree based algorithm
2) It uses join and prune technique	2) It constructs conditional frequent pattern tree and conditional pattern base from database
3) It is a breadth-first-search	3) It used depth-first search
4) It utilizes to level wise approach	4) It utilizes a pattern-growth approach
5) requires large memory space	5) requires small memory space

Apriori	FP-growth
★ 6) It scans database multiple time	6) It scans database only twice

Advantages of FP-growth over Apriori

→ FP-growth is more efficient as

- it able to mine the confidential pattern base in
- substantially reduce the size of the dataset to be searched

Data format:

2 types

- horizontal data format → {TID: itemset}
- vertical data format → {item: TID_set}

→ in vertical data format

Eclat (Equivalence Class Transformation) is used.

Eclat:

→ transform the horizontally formatted data into the vertical format.

→ Frequent k itemset can be used to construct the candidate (k+1) itemsets → based on the Apriori property

→ Support count of an itemset.

length of the TID-Set.

Advantages:

→ I take the advantages of Apriori property

→ no need to scan the database

to find the support of (k+1) itemset

Disadvantages:

→ TID_sets can be quite long

→ taking substantial

→ memory space

→ computation time

for intersecting long Set.

Example-6.6

Given that

TID	List of item IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

min-sup = 2

item-1 vertical data format

item	TID-list
I1	{T100, T400, T500, T700, T800, T900}
I2	{T100, T200, T300, T400, T600, T800, T900}
I3	{T300, T500, T600, T800, T900}
I4	{T200, T400}
I5	{T100, T800}

As min_sup = 2
∴ All are frequent-1 itemset

2- itemset vertical data format.

itemset	TID set
{I1, I2}	{T100, T400, T800, T900}
{I1, I3}	{T500, T800, T900}
{I1, I4}	{T400}
{I1, I5}	{T100, T800}
{I2, I3}	{T300, T600, T800, T900}
{I2, I4}	{T200, T400}
{I2, I5}	{T100, T800}
{I3, I4}	∅
{I3, I5}	{T800}
{I4, I5}	∅

As min sup = 2
∴ ∅ {I1, I4}, {I3, I4}, {I3, I5}, {I4, I5}
are not frequent.

3-itemset vertical dataset.

Applying "Apriori property" on 2-itemset data:

{I}, itemset	TID_Set
{I1, I2, I3}	{T800, T900}
{I1, I2, I5}	{T100, T800}

Ans

Chapter-3
Data Preprocessing

Data Quality:

Data have quality if they satisfy the requirement of intended use.

Many factors comprising data quality.

- accuracy → The data are correct or not accurate
- completeness
- consistency
- timeliness
- believability
- interpretability

Accuracy: The data are correct or not accurate or not.

Reasons for inaccurate data:

→ data collection instruments used
  ↳ may be faulty

→ human or computer errors
  ↳ occuring at data entry

→ errors in data transmission

Disguised missing data:

→ User may not wish to submit personal data

→ Purposely submit incorrect data
  ↳ for mandatory fields

Completeness:

→ Attributeness of interest may not always be available.

→ they were not considered important at the time of data entry

→ Relevant data may not be recorded.

→ due to equipment malfunction

→ Inconsistent data may be deleted.

Consistency:

- Timeliness:

  ~~Data are~~

  Are the data timely updated or not.

- Believability:

  How trustable the data are.

- Interpretability:

  How easily the data can be understood.

Major tasks in data preprocessing:

- Data cleaning
- Data integration
- Data reduction
  - dimensionality reduction
  - numerosity "
- Data transformation and discretization

#Data cleaning:

Clean the data by

- filling in missing value
- smoothing noisy value
- identifying or removing outliers
- resolving inconsistencies

Data integration:

It involves integrating multiple → databases
→ data cubes
→ files

Ex

[customer_id, cust_id]

→ Redundant data may slow down/confuse the knowledge discovery process

→ Data cleaning and data integration are performed as a preprocessing step

Data reduction:

→ Obtain a reduced representation of data set

→ yet produce the same analytical result

2 Strategies

→ dimensionality reduction

→ numerosity "

→ data compression.

Dimensionality reduction:

→ obtain a compressed/reduced representation of original data.

Ex: → PCA

→ Wevlet transformation

→ attribute subset selection

→ attribute construction

↳ small set of more useful attribute is derived

↳ from original set

Numerosity reduction:

→ data are ~~represen~~ replaced by alternative, smaller representation

↳ by using

→ parametric model (regression, log-linear models)

→ non-parametric "

Histogram,
cluster,
sampling,
data aggregation)

#Discretization and concept hierarchy:

- → Normalization
- → data discretization
- → concept hierarchy generation

→ forms of data transformation.

Reason of data preprocess

→ it can improve data quality

helping to improve → accuracy
→ efficiency

→ important step in knowledge discovery process.

→ quality decisions must be based on quality data.

Missing value:

Many tuples have no recorded value

→ for several attribute

→ Data is not always available

Handling missing data

6 methods

→ Ignore the tuple

→ used when class label is missing

→ not very effective

→ Fill the missing value manually

→ time consuming

→ not feasible for large data

→ Use a global constant

→ Replace all missing attribute values → by the same constant

→ Unknown

→ ∝

→ Use a measure of central tendency for the attribute

→ for normal/symmetric data distribution → mean

→ " skewed " " → median

→ Use the attribute mean or median for all samples belonging to the same class

→ Use the most probable value to fill in missing value

→ determined with

- regression
- inference-based tools
  - Bayesian formalism
  - decision tree
-

Reason of missing value:

→ equipment malfunction

→ inconsistant thus deleted

→ data not entered
  - due to misunderstanding

→ certain data may not be considered important
  - at the time of entry

Noise:

A Random error or variance in a measured variable

Reason of noisy / inaccurate data:

- → faulty data collection instrument
- → Data entry problem
- → Data transmission ,,
- → technology limitation
- → inconsistency in naming convention

Handling Noisy Data:

1) Binning method
2) Regression
3) Outlier analysis
4) Semi-automated method
   → combined computer & human inspe

Binning method:

→ sort the data and partition into (equal frequency) bins

→ ~~then we can~~
→ smooth each bin by
   → smoothing by mean
   → " by median

→ Smoothing by bin boundary

→ maximum and minimum value of a bin is identified

→ each bin value is replaced by closest boundary value

★ Binning is also used as a discretization technique

Smoothing by mean:

a	b	c

$x = \frac{a+b+c}{3}$

x	x	x

Smoothing by median

a	b	c

$x = b$

b	b	b

Smoothing by bin boundary

a	b	c

$x = \min(|a - \text{bin}_i|), (|c - \text{bin}_i|))$

if $x = |a - \text{bin}_i|$ then replaced by a

if $x = |c - bin_i|$ then replaced by c

Regression:

→ conforms data values form a function

Ex: Linear regression

$$y = \theta_0 + x\theta_1$$

finding the best line to fit two attributes so that one attribute can be used to predict another attribute

Outlier analysis:

→ detected by clustering

→ Similar values are organized into groups / clusters

→ value that fall outside the cluster → outlier

Example:

Sorted data for price

4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34

bin size = 4

∴ number of bin = $\frac{12}{4} = 3$

Partition into (equi-depth) bins:

→ Bin 1: 4, 8, 9, 15

→ Bin 2: 21, 21, 24, 25

→ Bin 3: 26, 28, 29, 34

Smoothing by bin mean:

→ Bin-1: 9, 9, 9, 9

→ Bin-2: 23, 23, 23, 23

→ Bin 3: 29, 29, 29, 29

Smoothing by bin boundaries:

→ Bin-1 : 4, 4, 4, 15

→ Bin-2 : 21, 21, 25, 25

→ Bin 3 : 26, 26, 26, ~~29~~ 34

Data Integration

It is the merging of data from multiple data stoe stores.

Ad. of data integration:

→ help to reduce and avoid redundancies inconsistencies

→ help to improve the accuracy

→ help to " " speed of

↳ subsequent mining process

Schema integration:

→ integrate metadata from different sources

e.g. A.cust_id ≡ B.cust_#

- Considerable issue during data integration

2 issues
→ schema integration
→ object matching

- Entity identification problem:

Equivalent real world entities
from multiple data sources
→ can be different.

Ex:
- Will Smith = W. Smith
- customer_id = cust_number
  → how the data analyst / computer be sure

Sol^n:
→ attention must be paid to structure of the data

→

Detecting & resolving value conflicts

→ For the same entry, attribute values from different sources are different

- Possible reason:
  - → different representation
  - → " scaled

E.g. metric vs. British units
different scale.

Redundant attribute:

A attribute may be redundant if it can be derived from another attribute / set of attributes.

Handeling redundancy in data integration

→ by using correlation analysis

- Chi-square correlation → Nominal data
- Pearsons product moment coefficient / correlation coefficient → Numeric data

Chi-Square correlation test: / Pearson χ^2 Stati

It can be computed as

$$\chi^2 = \sum_{i=1}^{c} \sum_{j=1}^{r} \frac{(o_{ij} - e_{ij})^2}{e_{ij}}$$

where, o_{ij} = observed frequency

e_{ij} = expected ,,

$$e_{ij} = \frac{count(A=a_i) \times count(B=b_j)}{n}$$

of the joint event (A_i, B_j)

n → number of tuple

count(A=a_i) → ,, of tuple having value a_i for A

count(B=b_j) → ,, of ,, having value b_j for B

NB:

Hypothesis:

→ A and B are independent, there is no correlation between them

→ test is based on a significant level with

degree of freedom : $(r-1)(c-1)$

Correlation coefficient

Computed as

$$r_{A,B} = \frac{\sum_{i=1}^{n}(a_i - \bar{A})(b_i - \bar{B})}{n\,\sigma_A\,\sigma_B}$$

$$= \frac{\sum_{i=1}^{n}(a_i b_i) - n\bar{A}\bar{B}}{n\,\sigma_A\,\sigma_B}$$

Where,

n = number of tuples

a_i = values of A in tuple i

b_i = " of B in " i

σ_A = Standard ~~deprecative~~

σ_A = Standard deviation of A

σ_B = " " of B

$\bar{A}$ = mean values of A

$\bar{B}$ = " " of B

Evaluation

$$-1 \leq r_{A,B} \leq 1$$

→ $r_{A,B} > 0$. A and B are positively correlated

→ $r_{A,B} < 0$, A and B " negatively "

→ $r_{A,B} = 0$, A and B " independent

Covariance:

→ Two numeric attributes A and B

→ Set of n observations $\{(a_1, b_1), \ldots (a_n, b_n)\}$

$$Cov(A, B) = E((A - \bar{A})(B - \bar{B}))$$

$$= \frac{\sum_{i=1}^{n} (a_i - \bar{A})(b_i - \bar{B})}{n}$$

$$\boxed{Cov(A, B) = E(A \cdot B) - \bar{A}\bar{B}}$$

where,

$E(A) = \bar{A}$

$E(B) = \bar{B}$

Relation between Correlation and covariance

$$r_{A,B} = \frac{Cov(A, B)}{\sigma_A \sigma_B}$$

Variance:

Special case of covariance

where two attributes are identical

Example - 3.1

Given that,

	male	female	Total
fiction	250(90)	200(360)	450
non-fiction	50(210)	1000(840)	1050
Total	300	1200	1500

χ^2 = Number in the parenthesis are the expected frequencies

$$\chi^2 = \frac{(250-90)^2}{90} + \frac{(50-210)^2}{210} + \frac{(200-360)^2}{360} + \frac{(1000-840)^2}{840}$$

$$= \frac{160^2}{90} + \frac{(-160)^2}{210} + \frac{(-160)^2}{360} + \frac{(160)^2}{840}$$

$$= 160^2 \left(\frac{1}{90} + \frac{1}{210} + \frac{1}{360} + \frac{1}{840} \right)$$

$$= 160^2 \times \frac{5}{252}$$

$$= 507.936$$

$$\simeq 507.94$$

Ans

$\therefore$ degree of freedom $= (2-1)(2-1)$

$= 1 \times 1 = 1$ Ans

Example 3.2

Given that,

Time point	AllElectronics	HighTech
t1	6	20
t2	5	10
t3	4	14
t4	3	5
t5	2	5

(a) & (b)

Let, A = AllElectronics

B = HighTech

$$\therefore E(A) = \frac{6+5+4+3+2}{5} = 4 = \bar{A}$$

$$E(B) = \frac{20+10+14+5+5}{5} = 10.8 = \bar{B}$$

$$\therefore \text{Cov}(A,B) = \frac{\Sigma (A-\bar{A})(B-\bar{B})}{n}$$

$$\text{Cov}(A,B) = \frac{\{(6\times20)+(5\times10)+(4\times14)+(3\times5)+(2\times5)\}}{5} - \bar{A}\cdot\bar{B}$$

$$= \frac{251}{5} - 4\times10.8$$

$$= \frac{251}{5} - 43.2$$

$$= 50.2 - 43.2$$

$$= 7$$

As $\text{Cov}(A,B) > 0$ ~~So A and B i.e~~

So, positive covariance

So, Stock price for both companies rise together.

Example

Given that,

$(A, B) = (2,5); (3,8); (5,10); (9,11); (6,14)$

$Cov(A, B) = ?$

$$E(A) = \frac{2+3+5+9+6}{5} = \bar{A}$$

$$= 9$$

$$E(B) = \frac{5+8+10+11+14}{5} = \bar{B}$$

$$= \frac{48}{5} = 9.6$$

$$E(A \cdot B) = \frac{(2\times5) + (3\times8) + (5\times10) + (9\times11) + (6\times14)}{5}$$

$$= \frac{10+24+50+99+84}{5}$$

$$= 42.4$$

$\therefore Cov(A, B) = 42.4 - (9.6 \times 9)$

$= 42.4 - 38.4 = 4$

$\therefore$ Positive covariance

Ans

Wavelet transform:

A linear signal processing technique that when applied to a data vector X, transform it to a numerically different vector X'

→ of wavelet coefficients

→ wavelet coefficients larger than some user-specified threshold can be retained

→ All other coefficients are set to 0.

→ halves the data at each iteration

Hierarchical pyramid algorithm:

→ length, L of the input a data vector must be an integer
→ power of 2

→ Each transformation involves applying two functions
→ some data smoothing → sum / weighted average
→ a weighted difference

→ Functions are applied to pair of data points in X.
→ pair of measurements (x_{2i}, x_{2i+1})
→ results of two data sets of length L/2

→ Functions are applied to the data sets until the resulting data sets obtained are of length 2

→ Selected values from the data sets obtained in the previous iterations

Attribute subset selection

Reduces the data set size by removing irrelevant or redundant attribute.

Goal:

→ to find a minimum set of attribute so that resulting probability distribution of data is as close as possible to original distribution.

NB: For n attribute there are 2^n possible subsets

Greedy methods for attribute subset selection:

- methods
  - → Stepwise forward selection
  - → " backward elimination
  - → Combination of forward selection & backward elimination
  - → Decision tree induction

Stepwise forward selection:

- ** → Start with an empty set
  - → best of the original attributes is determined
    - → added to the reduced set.

Chapter-10
Cluster Analysis: Basic Concepts and Methods

Clustering:

It is the process of grouping a set of data objects
→ into multiple groups.
or
~~So that~~
Each group is known as cluster
→ Objects within the a cluster
→ high similarity
→ very dissimilar to
→ objects in other cluster.

☆ real life application of clustering

Application of clustering:

☆

☆ → taxonomy of living things

→ biology

→ security

→ business intelligence

→ web search

→ image pattern recognition

→ used to organize a large number of customers into groups

→ organize the search results into groups

→ present the result in a concise & easily accessible way

→ to discover clusters / subclasses in handwritten character recognition sys.

Clustering as a:

Application as a data mining function:

→ as a stand alone tool

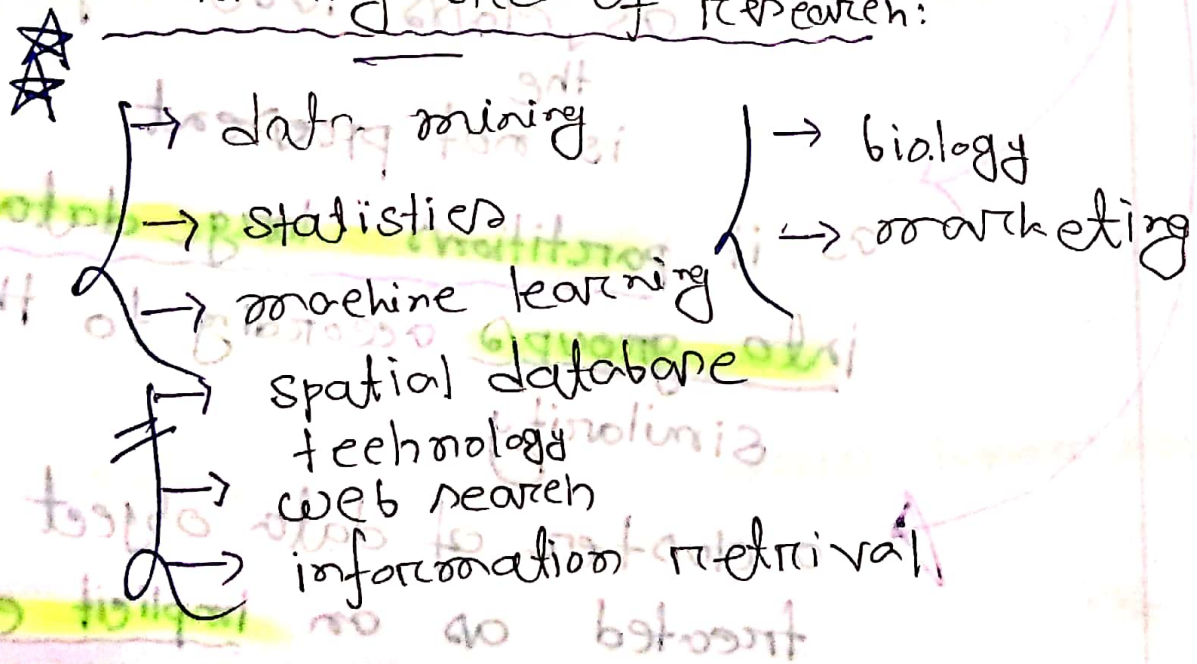
→ as a preprocessing step

→ outlier detection

→ to gain insight into the distribution of data

→ to observe the characteristic of each cluster.

Contributing area of research:



clustering ফলে সৃষ্ট বিভিন্ন cluster, cluster shape, clustering result নিয়ে আলোচনা

- Clustering is also called
  - → automatic classification
  - → data segmentation
  - → outlier detection
  - → unsupervised learning / learning by observation

as class label information is not present

as it partitions large data sets into groups according to their similarity

a cluster of data object can be treated as an implicit class.

★ input data এর বিভিন্ন দিক নিয়ে আলোচনা

Requirements of clustering:

→ **Scalability:**
Clustering all the data instead of sample

→ Ability to deal with different types of attribute:

→ typically designed to cluster numeric data

→ applications may require clustering other data types

- → binary
- → nominal
- → ordinal
- → mixture of these data type

→ **constraint-based clustering**

→ user may give inputs on constraints

→ use domain knowledge is used

→ interpretability and usability:

⇒ User wants cluster results to be → Inter-pretable,
→ comprehensible
→ usable

→ Discovery of clusters with arbitary space

→ Ability to deal with noisy data

→ capability of clustering high-dimensionality data

→ Incremental clustering & insensitivity to input order

Requirements for → high domain quality of clustering

domain knowledge to determine → input parameter

Considerations for clustering Analysis

** Orthogonal aspects of comparing Clustering methods:

→ The partitioning criteria

→ All the objects are partitioned

→ partition data object hierarchically

→ separation of clusters:

→ data object may belong to only one cluster — Exclusive

→ " " " to more than one cluster — non-exclusive

→ Similarity measure:

→ Distance based method
  → Euclidean space
  → vector space
  → road network
→ connectivity
  → cluster of arbitary space

→ take advantage of optimization

→ Clustering space

→ search for clusters within the entire given data space

→ subspace clustering
  → useful for high-dimensional

Basic clustering methods:

→ partitioning methods

→ Find mutually exclusive clusters of spherical shape

→ Distance based

→ use mean or medoid etc to represent cluster centers

→ Effective for small-to medium size data sets

Ex: K-means, K-medoids, CLARA, CLARANS

→ Hierarchical methods

→ clustering is a hierarchical decomposition.

2 types → agglomerative (bottom-up)
→ divisive (top-down)

→ distance based or density and continuity based

→ may incorporate other techniques like microclustering

Problem:

→ if one step is done, it can never be undone

→ can't correct erroneous decisions

Density based method

Continue to grow cluster as long as density in neighbourhood exceeds a threshold.

→ can find arbitary speed cluster

→ used to detect outlier

→ Clusters density:

* Each point must have a minimum number of points within its "neighborhood" :)

Ex:
- DBSAEN
- OPTICS
- DenClue

→ Grid-based methods:

quantize the object space into a finite number of cells

→ that forms a grid structure

Advantages:

→ Fast processing time

Ex
- STING,
- waveCluster
- CLIQUE

K-Means: A Centroid-Based Technique:

→ uses the centroid of a cluster C_i to represent that cluster.

→ centroied is defined by

- → mean
- → medoid

} of the objects assigned to the cluster

→ difference between an object $P \in C_i$ and c_i

→ measured by $dist(P, c_i)$

→ quality of cluster C_i

→ measured by sum of squared error

$$E = \sum_{i=1}^{k} \sum_{p \in c_i} dist(p, c_i)^2$$

c_i → centroid of cluster c_i

p → point in space

☆☆

K-means procedure:

Step-1:
Randomly selects k of the objects in D

Step-2:
Assign each object to the cluster to which the object is the most similar

Step-3:
update the cluster means

Step-4:
repeatedly perform step 2 and step 3 until the assignment is stable / no change

Complexity of k-mean:

→ time complexity is $O(nkt)$

where,

n = total number of objects

k = number of clusters

t = " of iterations

★ Its $k \ll n$ and $t \ll n$

so method is → relatively scalable
→ efficient

Problem:

★ → it is not guranteed to converge to the global optimum

→ terminates at a local optimum

Soln:

run the k-means algorithm multiple times
↳ with different initial cluster center

→ can't be applied to nominal data
Soln: use k-modes

How can we make the k-means more scalable?

Soln

→ use a good-sized ~~sample~~ set of samples in clustering

→ employ a filtering approach:
use a spatial hierarchical data index
↳ to save costs when computing means

⇒ Hierarchical clustering

Example 1 & 2

Given:

Data = {1, 2, 3, 8, 9, 10, 25}

K = 2 then

$C_1 = \{1, 2, 3\}$

$C_2 = \{8, 9, 10, 25\}$

$\therefore$ mean $(C_1) = 2$

mean $(C_2) = 13$

$\therefore E = (1-2)^2 + (2-2)^2 + (8-2)^2 + (8-13)^2 + (9-13)^2 + (10-13)^2 + (25-13)^2$

$= 1 + 0 + 1 + 25 + 16 + 9 + 64 + 144$

$= 196$

But comparing with this

$C_1' = \{1, 2, 3, 8\}$

$C_2' = \{9, 10, 25\}$

$\text{mean}(C_1') = 3.5$

$\text{mean}(C_2') = 14.67$

$\therefore E = (1-3.5)^2 + (2-3.5)^2 + (3-3.5)^2 + (8-3.5)^2$
$+ (9-14.67)^2 + (10-14.67)^2 + (25-14.67)^2$

$= 6.25 + 2.25 + 0.25 + 20.25 +$
$32.1489 + 21.8089 + 106.7089$

$= 189.67$

So k-mean is had sensitivity to outliers.

Strength of k-mean:

→ Efficient: $O(t k n)$.

Comparing → PAM
→ CLARA

Weakness of k-mean:

→ sensitive to → noisy data &
→ outliers

→ Need to specify K

→ good for numerical data.

K-modes: for categorical data

K-mediods: applied to a wide range of data

→ terminate at local optimum

→ not suitable to discover clusters → with non-convex shapes

K-Medoids:

Instead of taking mean value of the object in a cluster, as a reference point, mediods can be used.

mediods:

→ most centrally located object in a cluster.

Ex:

→ PAM (Partitioning Around mediods)

~~Which one~~

K-mean Vs k-medoid:

→ k-medoid is more robust than k-mean in the presence of
  → noise
  → outliers

→ complexity of k-mean $O(kmt)$
  " of k-medoid $O(k(n-k)^2)$
  if $k=1$ $O(k(n)^2)$ otherwise it is a NP hard problem

How can we modify the k-mean algorithm to diminish sensitivity to outliers?

Soln → pick actual objects to represent the cluster.

→ partitioning method is then performed based on ~~an~~ absolute ~~error~~ criterion.

the principle of minimizing the sum of the dissimilarities between

→ each object p and its corresponding representative object.

Absolute-error criterion:

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} dist(p, o_i)$$

where,

o_i = representive object of C_i

P = data point.

K = number of cluster

Q: How can we scale up the K-medoids method?

Sol

Answer:

→ use a sample based method

Ex: CLARA (Clustering LARge Application)

CLARA:

→ uses a random sample of data set.

→ PAM is applied them to compute

→ best medoids from sample

→ sample should closely represent the original data.

→ CLARA builds clustering

↓

from multiple random sample

Complexity:

→ $O(Ks^2 + K(n-k))$

Where,

s = size of the sample

k = number of cluster

n = number of objects

Weakness:

→ efficiency depends on sample size

→ sample may be biased

PAM

Step-1:

Select k representative objects arbitarily

Step-2:

For each pair of non-selected object h and selected object i calculate total swapping cost TC$_{ih}$

Step-3

For each pair of i and h

→ if TC$_{ih}$ < 0, i is replaced by h

→ assign each non-selected object to the most similar representative object.

Step-4:

Repeat steps 2-3 until there is no change.

- Efficiency improvement on PAM
  - → CLARA / PAM on samples
  - → CLARANS / Randomized re-sampling

Hierarchical clustering

→ It is a method works to group data object into a hierarchy or "tree" of clusters.

→ useful for data summarization

→ " for visualization

→ does not require number of clusters K as input

→ needs a termination condition

Types of hierarchical method

can be categorized into

→ algorithmic method

→ probabilistic

→ Bayesian

algorithmic method

* Agglomerative
* divisive
* multiphase

→ considered data object

↳ as deterministic

→ compute clusters according to the

deterministic distance

#Probabilistic method

→ probabilistic model to capture clusters.

→ measure quality of clusters by the features of model.

#Bayesian method

→ compute a distribution of possible clustering

→ it returns a group of
- clustering structure
- their probabilities
- Conditional on given data

Agglomerative Vs Divisive method

Agglomerative	Divisive
Strategy wise → it employs a bottom-up strategy	→ it employs a top-down strategy
→ starts by letting each object form its own cluster	→ starts by placing all objects in one cluster. (hierarchy's root)
→ iteratively merges cluster into larger and larger cluster	→ divides the root cluster into smaller subclusters.
→ until all the objects in a single cluster	→ recursively partitions those cluster into smaller ones. → until each cluster at the lowest level

Clustering approaches:

2 types

→ single-linkage

→ dendrogram

*Single-linkage:

→ each cluster is represented by all the objects in the cluster

→ similarity between two clusters is measured → by the similarity of the closest pair of data points belonging two different cluster.

→ merging process repeats until all the objects are eventually merged

→ to form one cluster

* dendogram:

→ it shows how objects are grouped together or partitioned step-by-step.

→ it use similarity scale & Leveling

Distance Measure in Algorithmic

Method:

Four widely used method

Minimum distance:

$$dist_{min}(C_i, C_j) = \min_{P \in C_i, P' \in C_j} \{|P - P'|\}$$

Maximum distance

$$dist_{max}(C_i, C_j) = \max_{P \in C_i, P' \in C_j} \{|P - P'|\}$$

Mean distance

$$dist_{mean}(C_i, C_j) = |m_i - m_j|$$

Average distance

$$dist_{avg}(C_i, C_j) = \frac{1}{n_i n_j} \sum_{P \in C_i, P' \in C_j} |P - P'|$$

Where,

$|p-p'|$ = distance between two points

m_i = mean for cluster C_i

n_i = number of object in C_i

nearest-neighbor clustering:

When an algorithm uses the minimum distance $d_{min}(C_i, C_j)$ to measure the distance between clustering

single-linkage algorithm:

Minimum distance between any two objects of two clusters exceeds a user defined threshold

Complete

farthest neighbor clustering:

Wher algorithem uses maximum distance $d_{max}(C_i, C_j)$ to measure the distance between clusters

Complete linkage algorithm:

Maximum distance $d_{max}(C_i, C_j)$ exceeds a user defined threshold.

Minimum spanning tree algorithm:

It is agglomerative hierarchical clustering algorithm that uses minimum distance measure

Spanning tree:

tree that covered all vertices

Minimal spanning tree:

One with the least sum of edge weight

MST

Agglomerative and divisive measure is sensitive to outliers or noisy data

↑ mean or average distance

can overcome the outliers sensitivity problem

→ average distance con handle

→ colegonie data

→ numeric "

~~BRJ~~

BJRCH

→ designed for clustering a large amount of numeric data

↓

by integrating hierarchical clustering & other clustering method

(hierarchical clustering + other ,,)

→ overcome two difficulties

1) scalability

2) inability to undo what was done in the previous step

→ it uses the notations of clustering feature
→ clustering feature (CF) tree

clustering feature (CF) tree

→ clustering feature is a 3D vector summarizing information
→ about clusters of objects.

$CF = (n, LS, SS)$

where,
LS = linear sum of the n points

$\sum_{i=1}^{n} x_i$

SS = Square sum of data points

$$\sum_{j=1}^{n} x_i^2$$

→ A CF is essentially a summary of the statistics

→ cluster's centroid, x_0

→ " radius, R

→ " diameter, D

$$x_0 = \frac{\sum_{i=1}^{n} x_i}{n} = \frac{LS}{n}$$

$$R = \sqrt{\frac{\sum_{i=1}^{n} (x_i - x_0)^2}{n}} = \sqrt{\frac{\sum_{i=1}^{n} x_i^2 - 2\sum_{i=1}^{n} x_i x_0 + \sum_{i=1}^{n} x_0^2}{n}}$$

$$= \sqrt{\frac{nSS - 2LS^2 + nLS}{n^2}}$$

$$D = \sqrt{\frac{\sum_{i=1}^{n}\sum_{j=1}^{n}(x_i - x_j)^2}{n(n-1)}}$$

$$= \sqrt{\frac{2nSS - 2LS^2}{n(n-1)}}$$

→ R is average distance from member object to the centroid

→ D is the average pairwise distance within cluster.

★ Clustering features are additive.

If C_1 and C_2 are two distinct clusters $CF_1 = (n_1, LS_1, SS_1)$

$CF_2 = (n_2, LS_2, SS_2)$

then, $CF_1 + CF_2 = (n_1+n_2, LS_1+LS_2, SS_1+SS_2)$

⊞ @ Example 10.5:

Given that,

3-points of C_1 is

(2,5), (3,2) and (4,3)

$CF_2 = (3, (35, 36), (417, 419))$

$\therefore CF_1 = (m, n_1, LS_1, SS_1)$

$= (3, (9,10), (29,36))$

$\therefore CF_3 = CF_1 + CF_2$

$= (6, (44,46), (446,478))$

And

CF-tree

- height-balanced tree
- stores the clustering features
  - for a hierarchical clustering
- non-leaf nodes store
  - sum of the CFs of their children

has two parameters:

- branching factor, B
- threshold, T

~~branching~~

branching factor, B:

maximum number of children per non-leaf node

threshold, T:

maximum diameter of subclusters → stored at the leaf nodes.

Multiphase clustering technique:

→ single scan of the data set yields a basic good clustering

→ additional scans can optionally be used to further improve the quality

Phases of CF-tree

** two phases

Phase 1:

→ Scan the database to build an initial in-memory CF tree

→ viewed as multilevel compression of data

Phase 2:

→ applies a (selected) cluster algorithm

→ to cluster the leaf nodes

Chameleon

→ uses dynamic modeling

to determine the similarity between pairs of clusters.

Similarity is assessed based on

1) how well connected objects are within a cluster
2) the proximity of cluster

★ Two clusters are merged if their interconnectivity is high and they are close together

Similarity measure between pair of cluster C_i and C_j

→ relative interconnectivity.

$RI(C_i, C_j)$

→ relative closeness

$RC(C_i, C_j)$

relative interconnectivity $RI(C_i, C_j)$:

- Defined as the absolute interconnectivity between C_i and C_j normalized with respect to

internal interconnectivity of two clusters.

$$RG(e_i, e_j) = \frac{|EC_{e_i, e_j}|}{\frac{1}{2}(|EC_{e_i}| + |EC_{e_j}|)}$$

where,

EC_{e_i, e_j} = Edge cut for a cluster containing both e_i & e_j

$|EC_{e_i}|$ = Minimum sum of the cut edge that

→ that partitions G into two equal parts

Relative closeness:

Defined as ~~abso~~ absolute closeness between C_i and C_j

normalized w.r.t.

→ internal closeness of the two clusters.

$$RC(C_i, C_j) = \frac{\bar{S}_{EC\{C_i,C_j\}}}{\frac{|C_i|}{|C_i|+|C_j|}\bar{S}_{EC_i} + \frac{|C_j|}{|C_i|+|C_j|}S_{EC_j}}$$

where,

- $\bar{S}_{EC\{C_i,C_j\}}$ = average weight of the edge

Chameleon's procedure

→ uses a k-nearest neighbor graph to construct a sparse graph

→ uses a graph partitioning algorithm to partition the k-nearest neighbor graph

↓

large number of small subclusters

→ use agglomerative hierarchical clustering that iteratively merges subclusters

→ based on their similarity

Density-Based Methods:

If we model clusters as a dense regions in the data space, separated by sparse regions. to find cluster of arbitary shape. called density based model

Types of density-based method

* DBSCAN
* OPTICS
* DBNCLUE

DBSCAN's features

- Discovers clusters of arbitrary shape
- Handle noise
- One scan
- Need density parameters as termination condition

Q: How can we find dense regions in density-based clustering?

Answer:

- density of an object o can be measured by
  - number of objects close to o.

→ It finds core objects

→ objects that have dense neighborhoods

- ~~Re~~ → maximum radius of
- ϵ-neighborhood: neighbourhood

The ϵ-neighborhood of an object o is the space within a radius ϵ centred at o

- MinPts:

Minimum number of points in an ϵ-neighbourhood of that points.

→ specifies the density threshold of dense regions.

Candidate set

$$N_{Eps}(P_i) : \{q \in D \mid dist(P,q) \leq \epsilon\}$$

Core object:

An object is core object if the ϵ-neighbourhood of the object contains at least MinPts objects.

Directly density-reachable:

For a core object q and an object P we say that P is directly density-reachable from q if P is within the ϵ-neighbourhood of q.

Density reachable:

A point p is density-reachable from q if there is a chain of objects $P_1 \ldots P_n$, such that $P_1 = q$, $P_n = p$, and P_{i+1} is directly-density reachable from P_i

Density-connected:

A point p is density-connected to a point q w.r.t ϵ and MinPts if there is a point o such that both p and q are density-reachable from o.

OR.

Two points $P_1, P_2 \in D$ are density-connected w.r.t ϵ and MinPts if there is an object $q \in D$ such that both P_1, P_2 are density reachable from q.

Question: How does DBSCAN find clusters?

Answer:

→ All the objects in a given data set D are marked as "unvisited"

→ randomly selects an unvisited object p.

→ marks p as visited

→ check whether p is core object or not

→ If not p marked as → noise point

→ Otherwise → a new cluster C is created for p.

→ All the objects in the ε neighbourhood of p added to N.

→ For an object (p) in N
if it visited

→ if p' is a core object then
all objects within ε-neighbourhood of p'
→ added to N

→ Add p' to C

→ iteratively adds to C
until N is empty

→ To find the next cluster
→ randomly selects an unvisited
unvisited objects from
remaining ones

→ continuous the process
until all of the points
have been processed.

Computational complexity of DBSCAN:

→ if a spatial index is used
complexity is $O(n \log n)$

→ otherwise complexity is $O(n^2)$

n → number of datasets of database objects

Evaluation of Clustering:

The major task of clustering evaluation include the following

→ Assessing clustering tendency

For a given data set, we assess whether a nonrandom structure exists in the data

→ Determining the number of clusters in a data set

It is desirable to estimate the number of cluster before a clustering algorithm is used to drive detailed cluster

→ necessity clustering quality?

→ how good the resulting cluster are

→ measure how well the clusters fit the data set

→ measure how well the cluster match the ground truth

Just

Question

⭐⭐ How can we assess the clustering tendency of a dataset?

Answer:

* By using 'Hopkins Statistics'

Calculation

1. Sample n points $p_1 \dots p_n$ uniformly from D

For each point p_i we find the nearest neighbour of p_i in D

$\therefore x_i = \min_{v \in D} \{dist(p_i, v)\}$

2. Sample m points $t_1 \ldots t_m$ uniformly.

For each t_i we find the nearest neighbour of t_i in D: $\{t_i\}$

$$y_i = \min_{v \in D, v \neq t_i} \{dist(t_i, v)\}$$

3. Hopkins Statistic

$$H = \frac{\sum_{i=1}^{m} y_i}{\sum_{i=1}^{m} x_i + \sum_{i=1}^{m} y_i}$$

If $H = 0.5$ (almost) D were uniformly distributed.

If $H = 0$ (almost) D - highly skewed

If $H > 0.5$ D were non-uniformly distributed

homogeneous hypothesis

D is uniformly distributed and thus contains no meaningful cluster

nonhomogeneous hypothesis:

D is nonuniformly distributg and thus contains significant cluster.

Determining the number of cluste

Empirical method

* number of clusters in a data set is $\sqrt{n/2}$

* number of data points in each cluster $\sqrt{2n}$ points

data set of n points

(ওয়ে)

Elbow method:

→ based on observation, that increasing the number of cluster can help to reduce the sum of within cluster variance

of each cluster

→ having more clusters allows one to capture finer groups of data objects
  → that are more similar to each other.

Procedure:

→ compute the clustering algorithm (like k-means) for different values of k (k > 0)

→ calculate the sum of within-cluster variance (var(k))

→ plot the curve of var w.r.t k

→ the first turning point of the curve

Suggests the "right" number.

- Cross-validation
  - → divide the given data set D into m parts
  - → use $m-1$ parts to build a clustering model
  - → use the remaining part to test the quality of the clustering
  - → for any $k>0$ repeat the process m times

→ average of the quality measure is taken as the overall quality measure

→ Compare the overall quality measure w.r.t different values of k.

→ find the number of clusters that best fit the data

Measuring clustering quality

Question:

How good is the clustering generated by method and how can we compare the clusterings generated by different method?

Answer:

Few methods to choose from for measuring the quality of a clustering.

two types of method:

→ Extrinsic methods

→ Intrinsic "

Extrinsic

1) The ground truth is available

2) Also known as Supervised method

3) Compare the clustering against the ground truth
→ using certain clustering quality measure

Intrinsic

1) The ground truth is unavailable

2) Also known as unsupervised method

3) Evaluate the goodness of a cluster

Extrinsic methods

It measures clustering quality $[Q(C, C_g)]$ for

→ a clustering C

→ given the ground truth C_g

Q is good if it satisfies the following 4 essential criteria

→ Cluster homogeneity

→ Cluster completeness

→ Rag bag

→ Small cluster preservation

clustering homogeneity:

The more pure the clusters in a clustering are, the better the clustering

Cluster completeness:

If any two objects belong to the same category, they should be assigned to same cluster.

→ according to ground truth

Raw bag:

Putting a heterogeneous object into a pure cluster should be penalized more than putting it into a raw bag.

Small cluster preservation

Splitting a small category into pieces is more harmful than splitting a large category into pieces

Deterministic Method:

Evaluate a clustering by examining

→ how well the clusters are separated

→ how compact the clusters are

Silhouette Coefficient:

let,

D = a set data set of n objects

$C_1, \ldots, C_k$ = k number of cluster of D

~~a(o) =~~

o = object $o \in D$

a(o) = average distance between o and all other objects in the same cluster.

$$a(o) = \frac{\sum_{o' \in C_i, o \neq o'} d(o, o')}{|C_i| - 1}$$

$b(o)$ = smallest average distance from o to all clusters

$$b(o) = \min_{C_j : 1 \leq j \leq k, j \neq i} \left\{ \frac{\sum_{o' \in C_j} d(o, o')}{|C_j|} \right\}$$

Silhouette coefficient of o

$$s(o) = \frac{b(o) - a(o)}{\max\{a(o), b(o)\}}$$

→ Silhouette coefficient is between -1 and 1

→ a(o) reflects the compactness of cluster

→ b(o) captures the degree to which o is separated from other cluster.

* positive: good
negative: bad (exception case)

Chapter-12
Outlier Detection

Outlier detection:

It is the process of finding data objects with behaviors that are very different from expection.

* also known as anomaly detection.

Outlier:

An outlier is a data objects that deviates significantly from the rest of the objects.

Outliers Vs noise

→ Noise is anything that is not the true signal

→ It may have values close to the true signal

→ An outlier is something that is much different than the other values

* noise is a random error or variance in a measured variable

→ Noise should be removed before outlier detection

Types of outliers

three categories

* global outliers
* contextual (conditional) outliers
* collective outliers

global outliers:

A data object is a global outlier if it deviates significantly from the rest of the data set.

* also called point anomalies
* Simplest type of outliers

determination:

Critical issue is to find an appropriate measurement of deviation ~~with~~ w.r.t the application in question

Contextual Outliers:

A data object is a contextual outlier if it deviates significantly w.r.t a specific context of the object.

* also called conditional outliers as they are conditional on the selected context.

Types of attributes of data object

→ Contextual attributes

→ Behavioral

Contextual attribute

define the object's context

Behavioral attribute

define the object's behavior characteristics

* It is a generalization of local outliers

Local outliers:

An object in a data set

is a local outlier if its density significantly deviates from local area in which it occurs

Collective outliers:

~~Given a~~

From a data set a subset of data objects from a colledive outlier, if the objects as a whole deviate significantly from the entire data set.

* it requires background information to model the relationship among objects to find group of outliers.

* Consider not only behavior of individual objects, but also that of groups of objects.

NB: → A data set may have multiple types of outlier

→ One object may belong to more than one type of outlier

Application of outlier detection:

- Credit card fraud detection
- Telecom
- Customer segmentation
- Medical analysis

Outlier detection method

Based on user-labeled

example of outliers can be obtained as.

* Supervised
* Unsupervised
* Semisupervised

Based on assumptions about normal data

* statistical methods
* proximity-based meth
* clustering-based meth

Supervised methods

→ outlier detection can be modeled as a classification problem

→ sample is used for training & testing

* model normal objects & report those not matching the model as outliers

* model the outliers and test object not matching the model is outlier.

Challenge: 2 challenges

→ Imbalanced classes. ~~ie~~

→ outliers are rare.

→ method of handling outlier.
imbalanced classes is oversampling

→ catch as many outliers as possible
& sensitivity and recall of a outlier is far more important than accuracy.

Unsupervised methods:

Assumption:

→ Assume that normal objects are somewhat clustered into multiple groups.

→ Normal objects follow a pattern far more frequently than outliers.

→ An outlier is expected to occur far away in feature space from any those groups of normal objects.

Weakness

Cannot detect collective outliers

~~→ In some case~~

~~In some case normal object~~

* normal objects are diversely distributed
* many such objects don't follow strong pattern

Ex:

In some intrusion detection and Computer virus detection problem, normal activities are diverse

Soln:

→ unsupervised may have high false positive rate

→ Supervised method is more effective for this

Process:

→ find clusters first

→ data objects not belonging to any cluster

→ are detected as outlier

Problem:

→ Hard to distinguish noise from outliers

→ It is often costly to find clusters first and then find outliers.

Semi supervised method

→ small set of the normal and/or outlier objects are labeled

→ most of the data are unlabeled

→ regarded as application of semi supervised learning methods

* If some labeled normal objects are available

→ used the labeled examples and unlabeled objects to train a model for scoring data

→ those not fitting the model of normal objects

→ detected as outlier

* If only some labeled outliers are available.

A small number of labeled outliers may not cover the possible outliers well.

→ to improve the quality of outlier detection, we can get help from normal objects

→ learned from unsupervised method

Chapter 7
Advanced Pattern Mining

Roadmap or pattern mining

three pattern mining aspects

- kind of patterns mined
- mining methodologies
- applications

Kind of patterns or Rule

3 types

* Basic patterns
* multilevel & multidimensional patterns
* Extended Patterns

Base pattern

Simple frequent pattern

Closed or max pattern, rare and negative pattern.

Frequent pattern

Pattern that satisfies minimum support and threshold

Closed pattern

P is closed pattern if there is no super-pattern P' with the same support as P

max pattern

P is max pattern if there is no frequent super-pattern of P

Multilevel and multidimensional Pattern:

- Based on the abstraction levels involved in a pattern
  - → multilevel association rule
  - → single-level " "
- Based on the number of dimension involved in the rule or pattern
  - → single-dimensional association rule
  - → multidimensional " "
- Based on the type of value handled in the rule or pattern
  - → Boolean association rule
  - → Quantitive " "

Example → 2 or more items

multilevel abstraction rule

buys(X, "computer") ⇒ buys(X, "printer")

buys(X, "laptop computer") ⇒

buys(X, "color laser printer")

computer → higher level of abstraction.

multilevel:

→ rule/pattern referenced two or more dimensions

- age
- income
- buys

age(X, "20...25") ^ income(X, "5K...5K") ⇒ buys(X, "IPad")

Extended pattern

Based on the constraints or criteria used to mine selective patterns:

- constraint
- approximate
- uncertain
- compressed
- mean weaken

satisfying a set of user defined constraint

- top-k
- redundancy-aware top-k

k most frequent itemsets for a user-specified value k

top-k patterns with redundant pattern excluded

Method for mining multilevel multidimensional Space:

→ multilevel association rule
→ multidimensional " "
→ quantative " "
→ rare patterns and negative pattern

→ involves numeric attributes that have an implicit ordering among values.

show negative correlations between items.

multilevel association rule:

Concept hierarchy:

It defines a set of mapping from a set of low-level concepts to a higher level more general concept set

* last level is the most specific abstraction level of hierarchy.

Association rules generated from mining data at multiple abstraction levels are called multiple-level or multilevel association rules.

NB: can be mined efficiently using concept hierarchies

→ with a support cont- from

Approach to mine multilevel association rule:

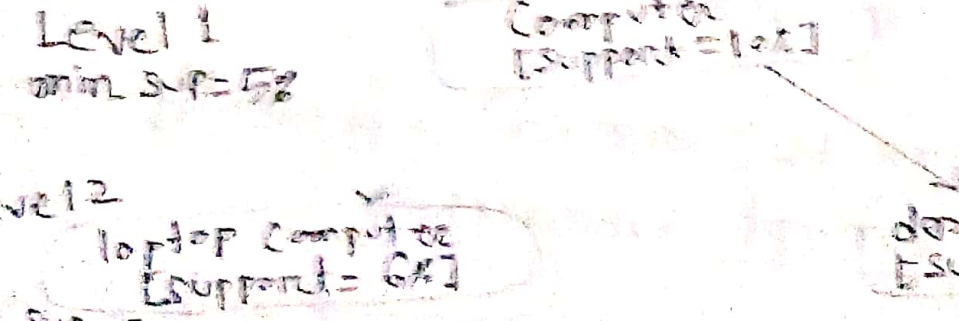
By using
- uniform minimum support
- reduced support
- group-based minimum support

~~Uniform level~~

~~Uniform level~~

Uniform minimum support:

The same minimum support threshold is used when mining at each abstraction level.



Adv:

→ search procedure is simplified

→ requires only one minimum threshold

drawback:

* if min_sup threshold is too high

→ miss some meaningful associations occurring at low abstraction level

* min_sup threshold is too low:

→ generate many uninteresting association occurring at high abstraction level.

Reduced support:

Each abstraction level has its own minimum support threshold

- The deeper the abstraction level → the smaller the corresponding threshold.

Level 1
min_sup = 5%

Computer [support = 10%]

Level 2
min_sup = 3%

laptop computer [support = 6%]

desktop computer [support = 4%]

- Uses on group-based minimum support:
  - User specific group based minimum support.

Example:

High price cameras may have lower threshold.

- Side effect of multilevel association rules:
  - generation of many redundant rules across multiple abstraction levels
    - → due to

Ancestor rule:

A rule R1 is an ancestor of rule R2, if R1 can be obtained by replacing the items in R2 by their ancestor in a concept hierarchy.

Example:

buys(X, "laptop computer") ⇒ buys(X, "HP printer")
[support = 8%, confidence = 70%]

buys(X, "Dell laptop computer") ⇒ buys(X, "HP printer")
[support = 2%, confidence = 72%]

- Multidimensional association rule:

Rules that involve two or more dimensions or predicates

- Interdimensional association rule:

Multidimensional association rules with no repeated predicates.

Ex:

age(x, "20...29") ^ occupation(x, "stu dent") => buys(x, "laptop")

- Hybrid dimensional association rule:

Multidimensional association rule with repeated predicates.

Ex:

age(x, "20...29") ^ buys(x, "lapto

=> buys(x, "HP print

Mining Quantitive Association Rules:

two approaches

~~dynamic~~

* Static discretization:

Quantitative attributes are discretized using → predefined concept hierarchies

* dynamic discretization:

Quantitative attributes are discretized or clustered into "bins" → based on the data distribution

Simple discretization:

→ discretize quantitative attributes into multiple intervals

→ treat them as nominal data in association mining

Problems of simple discretization:

generate enormous number of rules

Solution:

three methods to overcome this difficulty

1) a data cube method

2) a clustering-based method

3) a statistical analysis method

↓

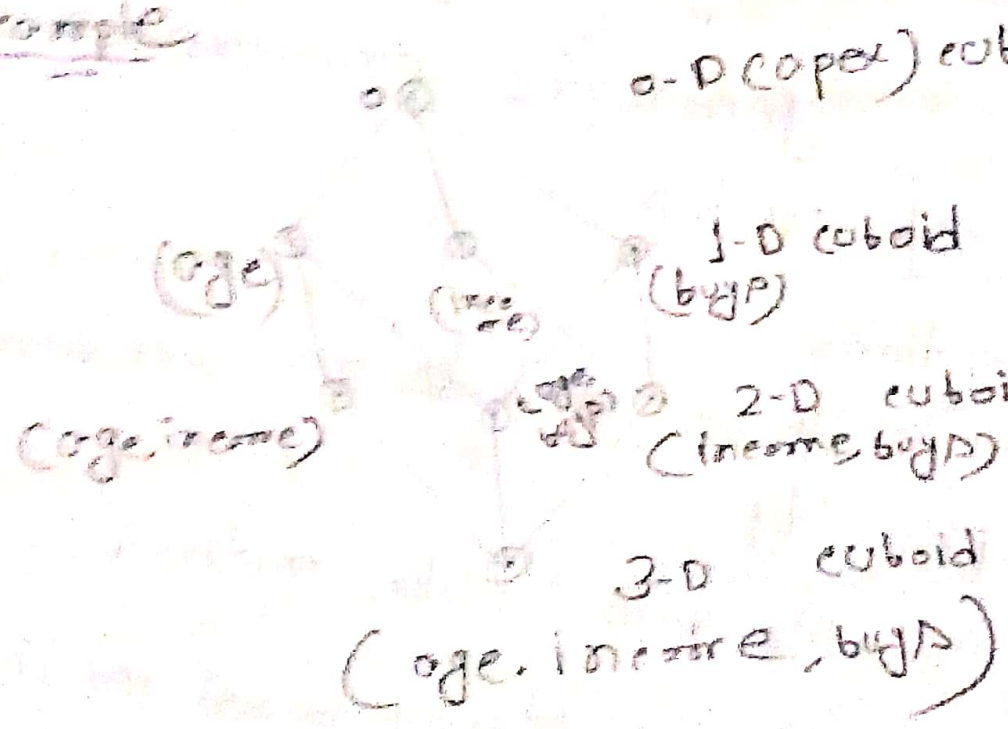
uneven exponential behaviors

Data Cube

Quantative → nominal
→ by discretization

Transformed multi dimensional data may be used to construct a data cube

Example



* cells of an n-dimensional cuboid can be used to store the support counts of the corresponding n-predicate Def.

Mining Clustering Based Quantitative Association

* Interesting frequent patterns are association rules are in general found of relatively dense clusters of quantitative attributes.

two approaches
* top-down approach
* bottom-up "

topdown approach:

Standard clustering algorithm can be applied to find clusters in this dimensions that satisfy the minimum support threshold.

↳ K-means
↳ density based clustering

bottom-up approach

→ first clustering in high-dimensional space to form clusters with ~~support~~ min_sup

→ progressively merging the clusters in space containing fewer dimensional combinations

Quantitative association based on Statistics:

It is possible to ~~disclose~~ exponential behaviour.

(sex = female) ⇒ (mean_wage = $7.90/hr)

overall_mean_wage = $9.24/hr

involves applying ~~statistical tests~~ to confirm the validity of our rules.

→ z-test

Infrequent or rare pattern:

Patterns with a frequency support that is below (far below) a user specified minimum support threshold.

Example

buying Rolex watches

Negative pattern:

If itemsets X and Y are both frequent but rarely occur together i.e. $sup(X \cup Y) < sup(X) \times sup(Y)$ then itemsets X and Y are negatively correlated and the pattern $X \cup Y$ is a negatively

correlated pattern.

Strongly correlated pattern / Def:-

If $sup(X \cup Y) < \min(sup(X), sup(Y))$
then X & Y are strongly negatively correlated and the pattern $X \cup Y$ is a strongly negatively correlated.

Problem

full inverted problem

~~multi transaction~~

transaction that does not contain any of the items being examined.

Example 7.4

Given that.

A store sold two needle A and B to patients out of 200 transactions only one transaction contained A and B.

$S(A \cup B) = \frac{1}{200} = 0.005$

$S(A) = \frac{100}{200}$

$S(B) = \frac{100}{200}$

$\therefore S(A) \times S(B) = \frac{100}{200} \times \frac{100}{200}$

$= 0.5 \times 0.5$

$= 0.25$

$S(A \cup B) < S(A) \times S(B)$

Suppose we have 10^6 null transactions

$\therefore S(A \cup B) = \frac{1}{10^6} = 10^{-6}$ $S(A) \times S(B)$

$\therefore S(A \cup B) < S(A) \times S(B) = \frac{10^1}{10^2} = \frac{1}{10^8} = \frac{100}{10^8} \times \frac{100}{10^2}$

Def-2

If x and y are strongly negatively correlated then,

$$Sup(X \cup Y) \times Sup(X \cup Y) < Sup(X \cup Y) \times Sup(X \cup Y)$$

this definition also can't remove the problems of null transaction.

Example 7.5

So $Sup(A \cup \bar{B}) \times Sup(\bar{A} \cup B) = \frac{99}{200} \times \frac{99}{200}$

$= 0.245$

$Sup(A \cup B) < Sup(A \cup B) = \frac{1}{200} \times \frac{199}{200}$

$= 0.005$

$sup(A \cup B) \neq sup(A \cup B) \gg sup(A) \cup B)$
$\times sup(A \cup B)$

$\therefore$ A and B are negatively correlated

Now for 10^6 null transaction.

$sup(A \cup B) \times sup(A \cup B) = \frac{99}{10^6} \times \frac{99}{10^6}$

$= 9.8 \times 10^{-9}$

Again $sup(A \cup B) \times sup(A \cup B)$

$= \frac{99}{10^6} \times \frac{(10^6 - 99)}{10^6}$

$\simeq 1.99 \times 10^{-4}$

$\therefore$ No null transaction creates null invariant problem.

Def-3

If x and y are both frequent

and $(P(x|y) + P(y|x))/2 < \epsilon$

where ϵ is a negative pattern

threshold. then pattern, x & y

is negatively correlated pattern.

Example-7:

Given that,

Constraint Based Frequent Pattern mining:

Constraint are added to specify mining interest pattern.

It helps to confine the search space.

The constraints can include:

→ Knowledge type constraints:

specify the type of knowledge to be mined.

- association
- correlation
- classification
- clustering

→ Data constraints:

specify the set of task-relevant data.

→ Dimension / level constraints:

Specify the desired
- → dimensions of the data
- → the abstraction levels
- → the level of the concept hierarchies

→ used in the mining

→ Interestingness constraints:

specify thresholds on statistical measure of rule interestingness
- → support
- → confidence
- → correlation

→ Rule constraints:

Specify the form of, or conditions on, the rules to be mined

→ Constraints may be expressed as
- meta-rules / rule templates
- maximum or minimum number of predictions

Metarule Guided mining

Metarule allow users to specify the syntactic form of rules → interested in mining.

Q: How can metarules be used to guide the mining process?

Answer:

A metarule is a rule template of the form

$$P_1 \wedge P_2 \wedge \dots \wedge P_l \Rightarrow Q_1 \wedge Q_2 \wedge \dots \wedge Q_r$$

where P_i ($i = 1 \dots l$) and Q_j ($j = 1 \dots r$) are either instantiated predicates or predicate variable.

Method to find MetaRule:

1. Find frequent $(l+r)$ predicates based on min threshold
2. Push constraints deeply when possible into the mining process.
3. Use confidence, correlation and other measure when possible.

Constraint Based Pattern Generation.

1. Pruning Pattern Space
2. Pruning Data space

Pattern Pruning constraints:

Constraint that facilitate pattern space pruning is called pattern pruning constraints.

Data Pruning constraints:

Constraints that can be used for data space pruning is called data pruning constraints

Types of pattern space pruning Constraints:

5 categories

- → antimonotonic
- → monotonic
- → succinct
- → convertible
- → inconvertible

* Antimonotonic:

If constraint c is violated its further mining can be terminated.

* Monotonic

If constraint c satisfied, no need to check c again

* succinct:

Constraint c must be satisfied, so one can start with the data set satisfying C.

* Convertible:

C is not monotone nor antimonotone, but it can be converted into it, if items in the transaction can be properly ordered.

Example of Antimonotone:

count (S) ≤ v
max (S) ≤ v
min (S) ≥ v
sum (S) ≤ v (∀a ∈ S, a ≥ 0)
range (S) ≤ v

Example of Monotonic

$min(S) \leq v$

$max(S) \geq v$

$count(S) \geq v$

$sum(S) \geq v$

$range(S) \geq v$

Data Space Pruning Constraints:

two properties/types

→ data succinctness

→ data antimonotonicity

* **Data succinctness:**

Data space can be pruned at the initial pattern mining process.

Ex: mined pattern must contain digital camera

* Data anti-monotone:

If a transaction t does not satisfy c. t can be pruned from its farther mining.

Mining High-Dimensional Data

Can't mine by using conventional mining approach.

two approaches

method-1

* data set with a large number of dimensions but small number of rows
* pattern of very log lenght.

method-1

Extends a pattern growth approach by further exploring the vertical data format

to handle data of first t

Ex: useful for the analysis of gene expressions in bioinformatics.

* Column enumeration/horizontal data format:

D is viewed as a set of rows.

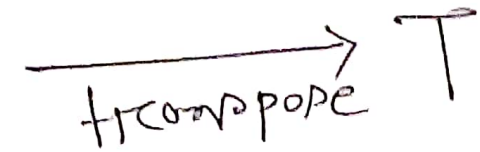
consists of an itemset

* row enumeration/vertical data format:

D is viewed as a set of row_IDs where the item appears in the traditional view of D.

Process:

→ The original data set, D can easily be transformed into a transposed data set T.



→ efficient pattern growth method can be developed

☐ Colossal pattern:

Pattern of very long length.

method: (Pattern-Fusion method)