Data Mining

INFS 4203/7203

Miao Xu

miao.xu@uq.edu.au

The University of Queensland, 2020 Semester 2

“Please be aware that this session is being recorded so it can be made available through Learn.UQ (Blackboard) to all students enrolled in the course. The reason we are recording the class presentations, discussions and chat room logs is because this provides a richer experience for all students and active classrooms help students’ learning.

The recording may be accessed and downloaded only by students enrolled in the course, including those students studying outside Australia.

If you would prefer not be captured either by voice or image in the recording, please let me know before the class starts. I recognise this will be the case for some students and it can be accommodated.”

Class recordings

If you are not wishing to be recorded:

- Turn off video and mute audio
- Use a proxy name for Zoom (student attendance will still be on record with the Course Coordinator)

Please note that students are not permitted to record teaching without the explicit permission of the Course Co-ordinator. This includes recording classes using Zoom.

For further information:

- PPL 3.20.06 Recording of Teaching at UQ
- UQ website: https://my.uq.edu.au/information-and-services/information-technology/software-and-web-apps/software-uq/zoom

"all members of the University community deserve respect from others in both formal and informal contexts"

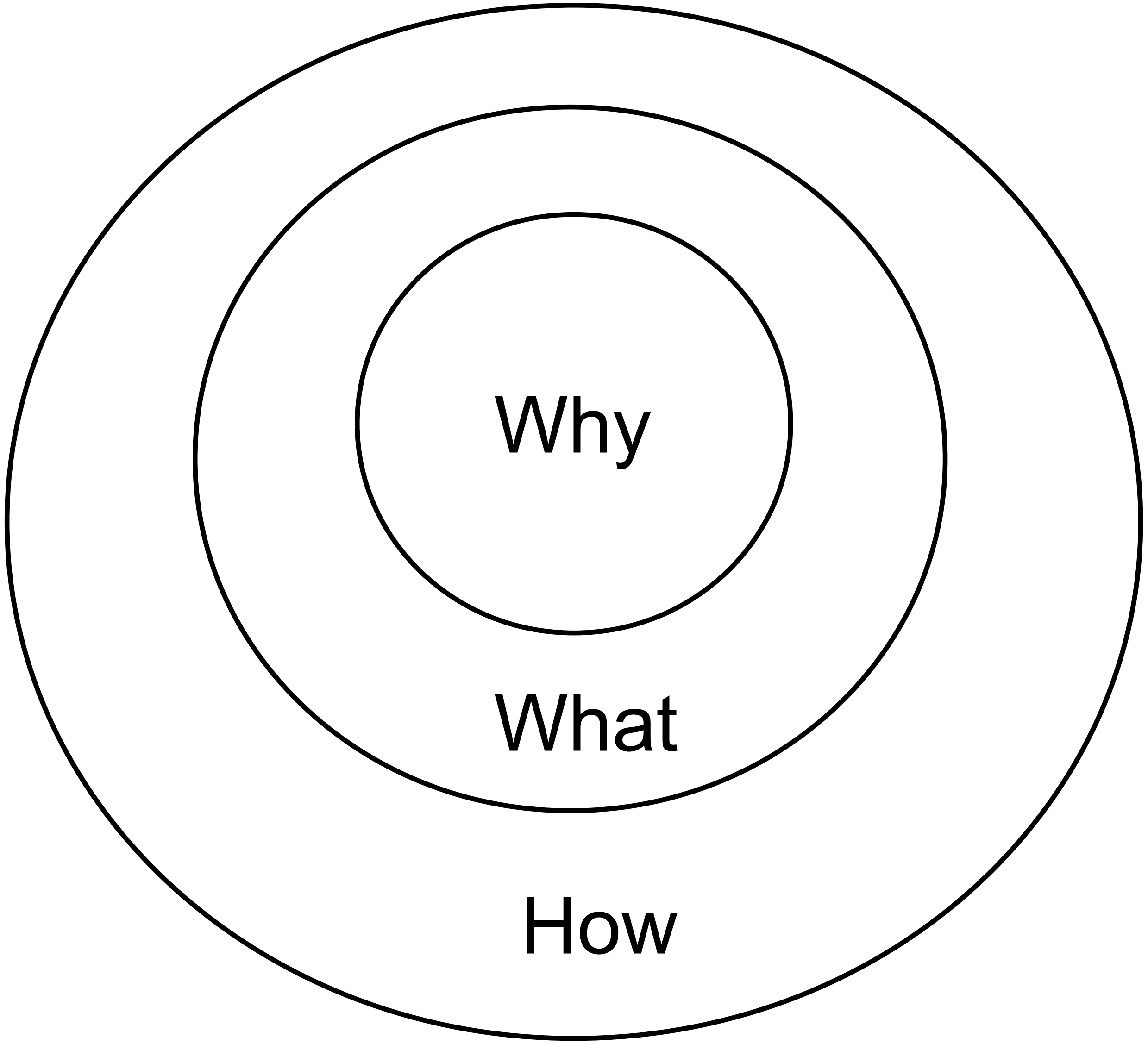
--the UQ Student Charter

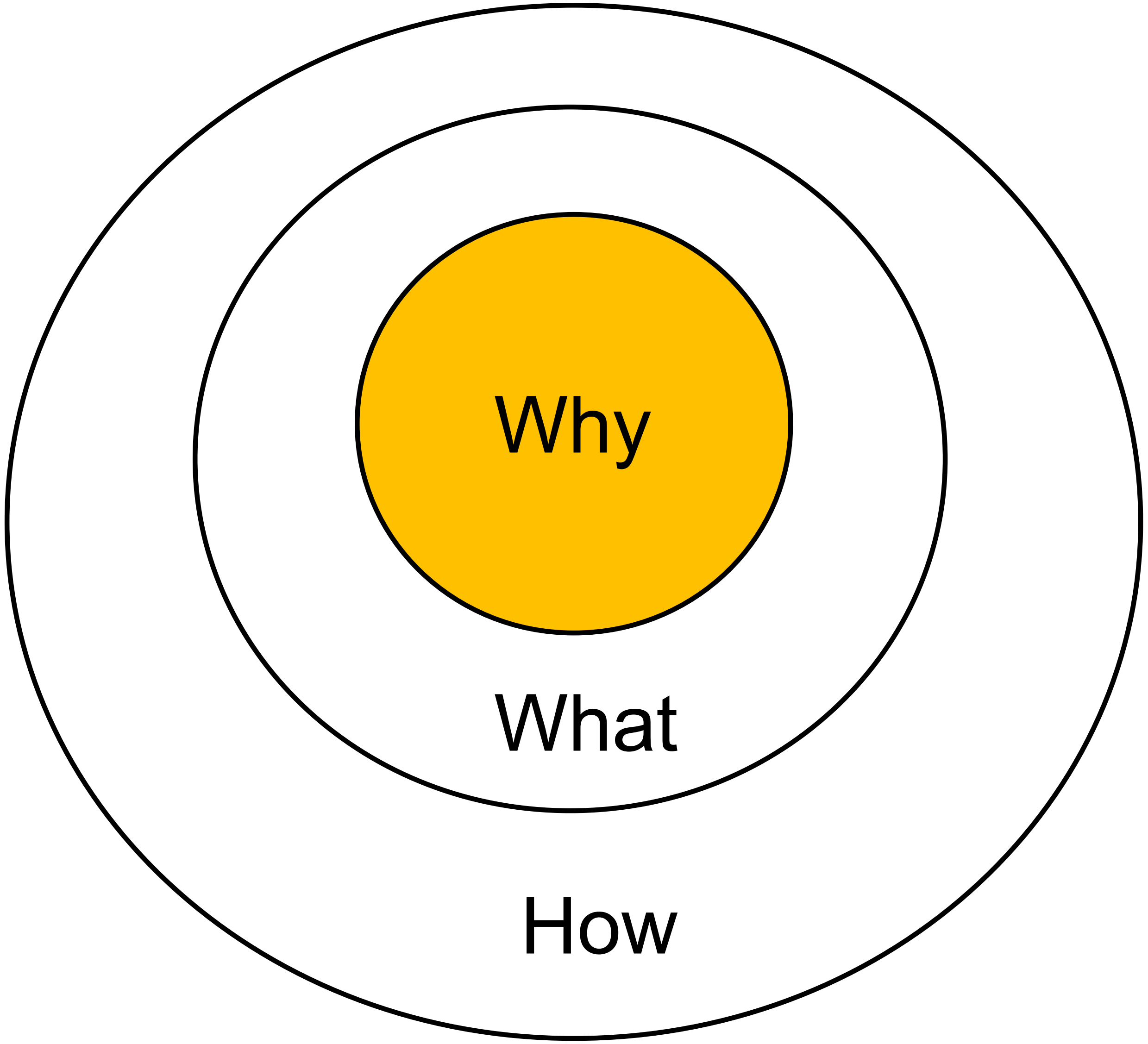
Please ensure the “to everyone” CHAT content is <u>accessible to everyone</u>, and <u>related to the course content</u>!

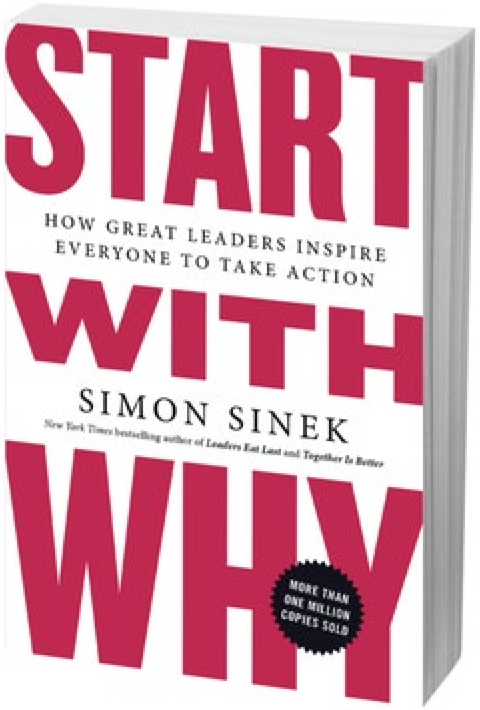
Last lecture

- Why data mining?
- What is data mining?
- What can data mining do?
- Who are using data mining?

Lecture 2: Association Rule Mining


Why
What
How


Why
What
How


START
HOW GREAT LEADERS INSPIRE
EVERYONE TO TAKE ACTION
WITH
SIMON SINEK
New York Times bestselling author of Leaders Eat Last and Together Is Better
WHY
MORE THAN ONE MILLION COPIES SOLD

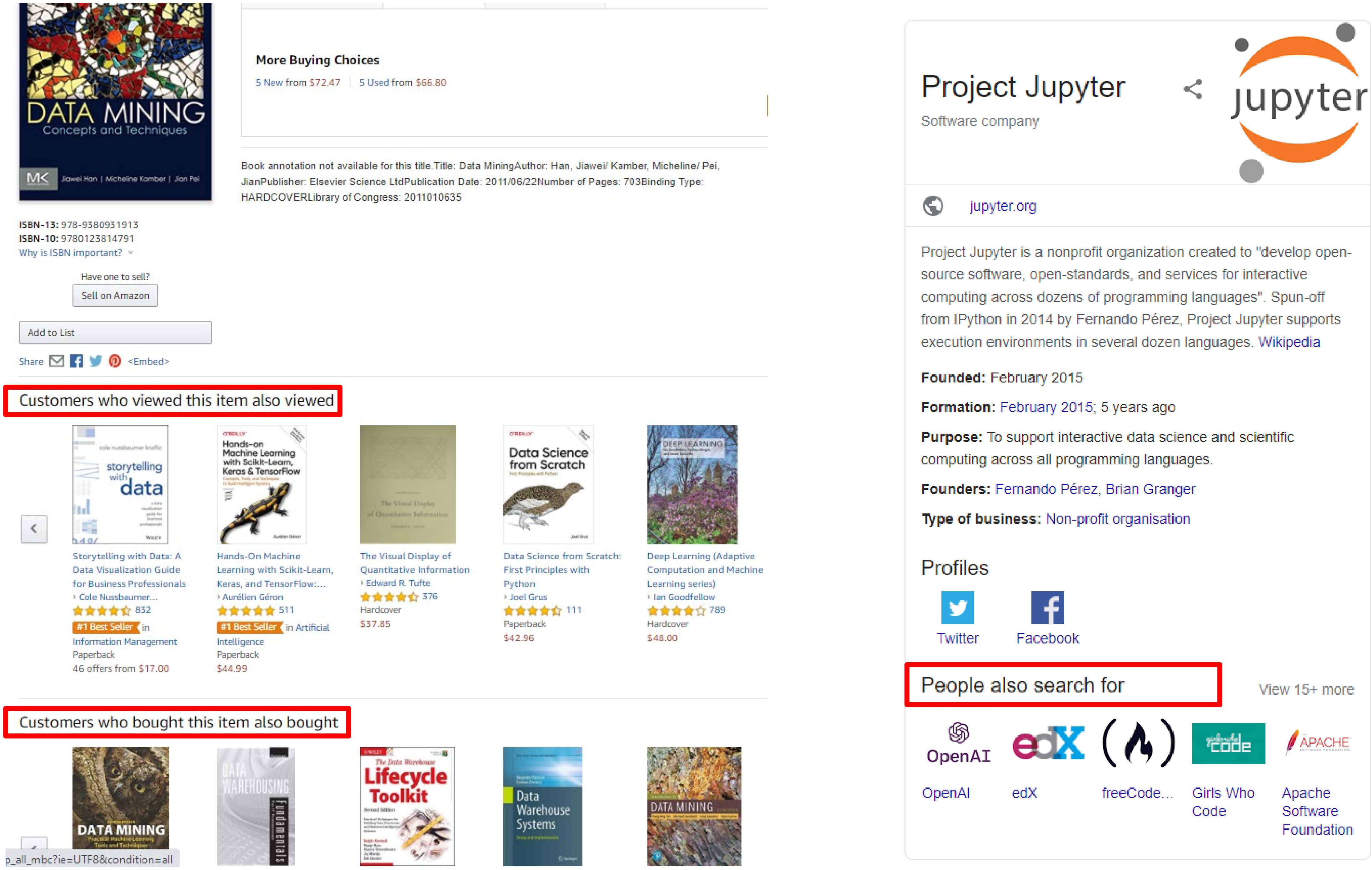
Market basket analysis

- Which items are frequently purchased together by customers?
- If customers buy {   }, what else will they often buy? How often?

- Given
  - A database of transactions
  - Each transaction is a list of items

TID	Items
1	{Bread, Milk}
2	{Bread, Diapers, Beer, Eggs}
3	{Milk, Diapers, Beer, Cola}
4	{Bread, Milk, Diapers, Beer}
5	{Bread, Milk, Diapers, Cola}

- Find
  - Frequent items purchased together
  - A rule that buying some items will lead to the buying of some other



DATA MINING
Concepts and Techniques
MK
Jiawei Han | Micheline Kamber | Jian Pei
More Buying Choices
5 New from $72.47 | 5 Used from $66.80
Book annotation not available for this title.Title: Data MiningAuthor: Han, Jiawei/ Kamber, Micheline/ Pei, JianPublisher: Elsevier Science LtdPublication Date: 2011/06/22Number of Pages: 703Binding Type: HARDCOVERLibrary of Congress: 2011010635
ISBN-13: 978-9380931913
ISBN-10: 9780123814791
Why is ISBN important?
Have one to sell?
Sell on Amazon
Add to List
Share <Embed>
Customers who viewed this item also viewed
Storytelling with Data: A Data Visualization Guide for Business Professionals
Cole Nussbaumer...
832
#1 Best Seller in Information Management
Paperback
46 offers from $17.00
Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow:...
Aurélien Géron
511
#1 Best Seller in Artificial Intelligence
Paperback
$44.99
The Visual Display of Quantitative Information
Edward R. Tufte
376
Hardcover
$37.85
Data Science from Scratch: First Principles with Python
Joel Grus
111
Paperback
$42.96
Deep Learning (Adaptive Computation and Machine Learning series)
Ian Goodfellow
789
Hardcover
$48.00
Customers who bought this item also bought
DATA MINING
DATA WAREHOUSING
Lifecycle Toolkit
Data Warehouse Systems
DATA MINING
p_all_mbc?ie=UTF8&condition=all
Project Jupyter
Software company
jupyter
jupyter.org
Project Jupyter is a nonprofit organization created to "develop open-source software, open-standards, and services for interactive computing across dozens of programming languages". Spun-off from IPython in 2014 by Fernando Pérez, Project Jupyter supports execution environments in several dozen languages. Wikipedia
Founded: February 2015
Formation: February 2015; 5 years ago
Purpose: To support interactive data science and scientific computing across all programming languages.
Founders: Fernando Pérez, Brian Granger
Type of business: Non-profit organisation
Profiles
Twitter
Facebook
People also search for
View 15+ more
OpenAI
edX
freeCode...
Girls Who Code
Apache Software Foundation

What else?

Traffic accident data

{Accident form=Collision with moving vehicles, Season=Summer, Hour=Deep night} => {Accident type=Fatal accident}

{Accident form=Collision with moving vehicles, Hour=Deep night} => {Accident type=Fatal accident}

{Week=Workday, Season=Summer, Hour=Deep night} => {Accident type=Fatal accident}

{Season=Summer, Hour=Deep night} => {Accident type=Fatal accident}

{Week=Workday, Hour=Deep night} => {Accident type=Fatal accident}

{Week=Workday, Season=Summer, Hour=Deep night} => {Accident form=Collision with moving vehicles}

{Hour=Deep night} => {Accident type=Fatal accident}

Qiu-ru Cai. Cause Analysis of Traffic Accidents on Urban Roads Based on an Improved Association Rule Mining Algorithm. IEEE Access 8: 75607-75615 (2020).

Medical data

Rule	Rule content	Support	Confidence
r_1	$(disease = meningitis) \to (tissue = brain)$	3	1
r_2	$(tissue = brain) \to (disease = meningitis)$	3	1
r_3	$(disease = liver\ cancer) \to (tissue = liver)$	2	1
r_4	$(sex = male) \wedge (disease = meningitis) \to (tissue = brain)$	2	1
r_5	$(sex = male) \wedge (tissue = brain) \to (disease = meningitis)$	2	1
r_6	$(sex = female) \to (tissue = brain)$	1	1
r_7	$(sex = female) \to (disease = meningitis)$	1	1
r_8	$(disease = cirrhosis) \to (tissue = liver)$	1	1
r_9	$(sex = male) \wedge (disease = liver\ cancer) \to (tissue = liver)$	1	1
r_{10}	$(sex = male) \wedge (tissue = liver) \to (disease = liver\ cancer)$	1	1
r_{11}	$(sex = female) \wedge (disease = meningitis) \to (tissue = brain)$	1	1
r_{12}	$(sex = female) \wedge (tissue = brain) \to (disease = meningitis)$	1	1
r_{13}	$(tissue = brain) \to (sex = male)$	2	2 / 3 = 0.67
r_{14}	$(sex = male) \to (tissue = brain)$	2	2 / 3 = 0.67
r_{15}	$(disease = meningitis) \to (sex = male)$	2	2 / 3 = 0.67
r_{16}	$(sex = male) \to (disease = meningitis)$	2	2 / 3 = 0.67
r_{17}	$(tissue = liver) \to (disease = liver\ cancer)$	2	2 / 3 = 0.67
r_{18}	$(tissue = brain) \to (disease = meningitis)$	2	2 / 3 = 0.67

https://pubmed.ncbi.nlm.nih.gov/31210270/

Others

- Text
- Bioinformatics
- …

COVID-19 e-print

Important: e-prints posted on arXiv are not peer-reviewed by arXiv; they should not be relied upon without context to guid
consulting multiple experts in the field.

[Submitted on 6 Apr 2020]

Discovering associations in COVID-19 related research papers

Iztok Fister Jr., Karin Fister, Iztok Fister

A COVID-19 pandemic has already proven itself to be a global challenge. It proves how vulnerable humanity can be. It has also
with this, our study analyses the abstracts of papers related to COVID-19 and coronavirus-related-research using association ru
method, called information cartography, was applied for extracting structured knowledge from a huge amount of association rule
situations throughout history.

Association rule mining to identify transcription factor interactions in genomic regions

Gaia Ceddia, Liuba Nausicaa Martino, Alice Parodi, Piercesare Secchi,
Stefano Campaner, Marco Masseroli

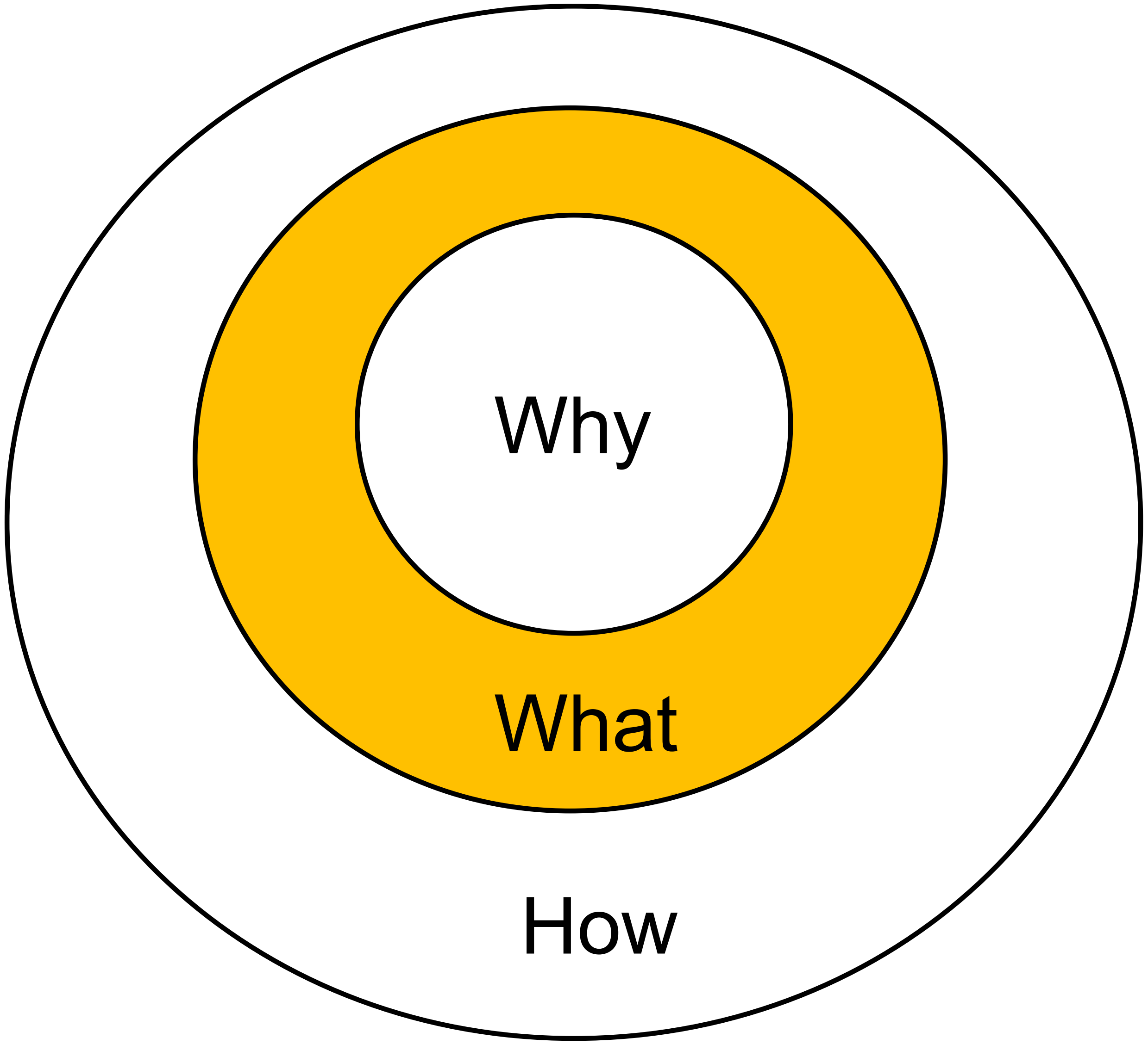
Bioinformatics, Volume 36, Issue 4, 15 February 2020, Pages 1007–1013,
https://doi.org/10.1093/bioinformatics/btz687

Published: 03 September 2019 **Article history** ▾

- Given
  - A database of lists
  - Each list is a list of items

TID	**Items**
1	{Bread, Milk}
2	{Bread, Diapers, Beer, Eggs}
3	{Milk, Diapers, Beer, Cola}
4	{Bread, Milk, Diapers, Beer}
5	{Bread, Milk, Diapers, Cola}

- Find
  - Frequent items appeared together in one list
  - A rule that some items appearing will lead to the appearing of some others

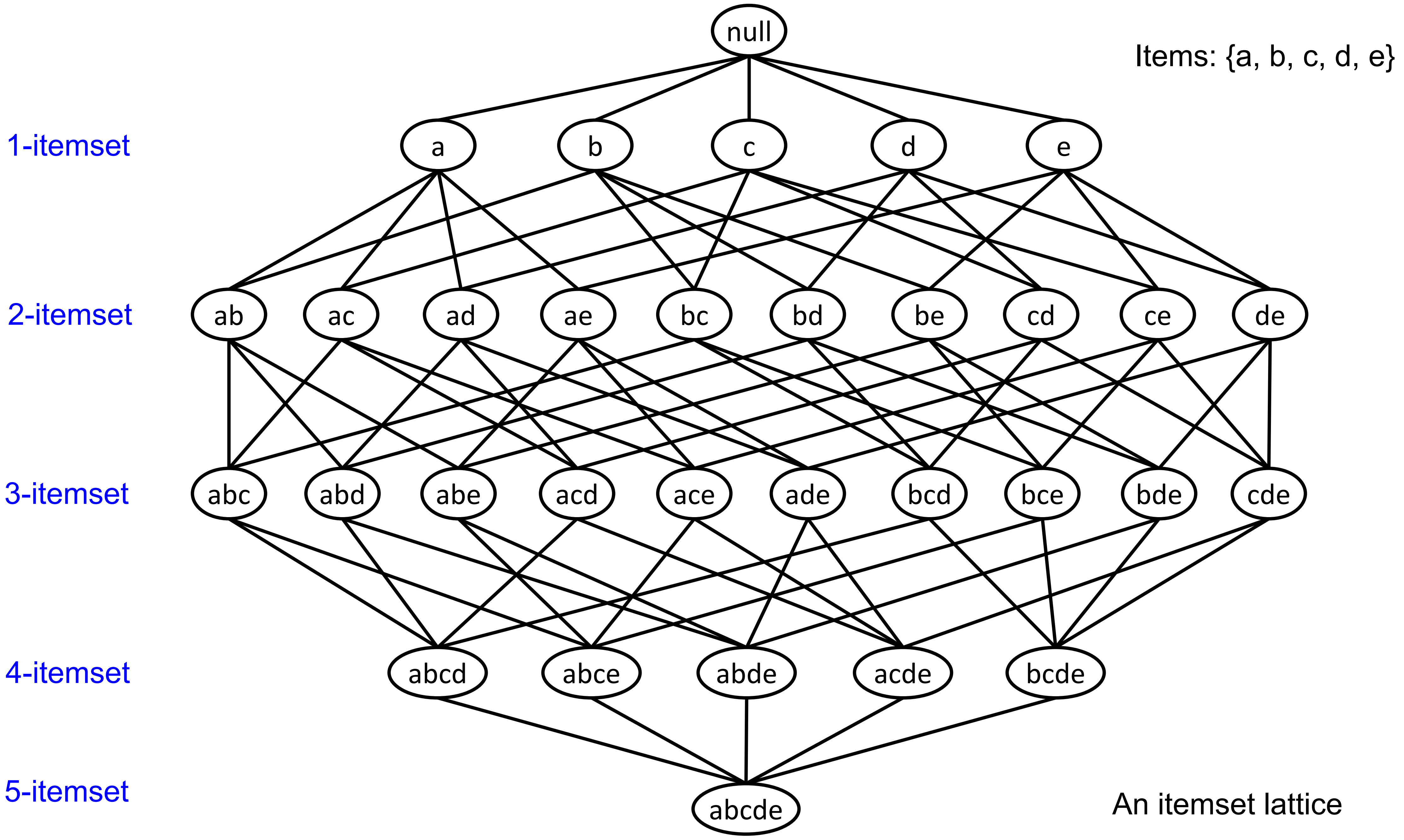

Why
What
How

- Given
  - A database of transactions
  - Each transaction is a list of items
- Find
  - Frequent items purchased together
  - A rule that buying some items will lead to the buying of others

Itemset

- Itemset
  - A set of items
  - A subset of all items
- *k*-itemset
  - An itemset containing *k* items

TID	**Items**
1	{Bread, Milk}
2	{Bread, Diapers, Beer, Eggs}
3	{Milk, Diapers, Beer, Cola}
4	{Bread, Milk, Diapers, Beer}
5	{Bread, Milk, Diapers, Cola}



An itemset lattice

Measurement on itemset: support

- Support count supp_cout(A) = $\#A$
  - How many transactions contain the itemset?
    - For {Diapers, Beer}: 3
    - For {Diapers, Beer, Milk}: 2
- Support rate supp_rate(A) = ${}^{\#A}/_{\#\ \text{Transaction}}$
  - The rate of transactions contain the itemset
    - For {Diapers, Beer}: 3/5 = 0.6 = 60%
    - For {Diapers, Beer, Milk}: 2/5 = 0.4 = 40%

TID	**Items**
1	{Bread, Milk}
2	{Bread, **Diapers, Beer**, Eggs}
3	{Milk, **Diapers, Beer**, Cola}
4	{Bread, Milk, **Diapers, Beer**}
5	{Bread, Milk, Diapers, Cola}

Frequent itemset: itemset with support larger than the specified ***minimum support threshold***.

turtle

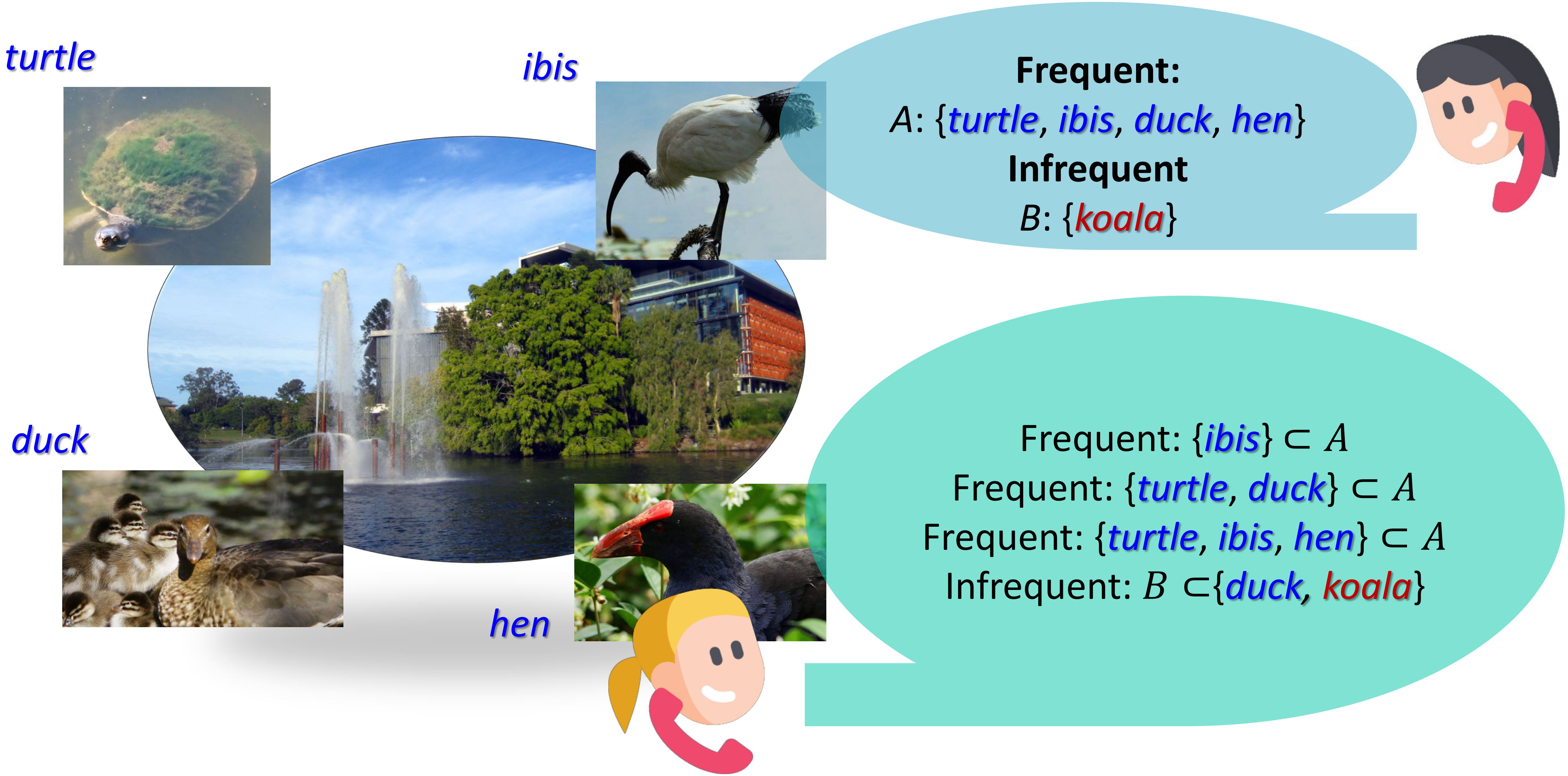
ibis

duck

hen

I see frequently ***turtle***,
ibis, ***duck*** and ***hen*** *but*
not ***koala***

Do you see:
ibis frequently?
turtle and ***duck*** frequently?
turtle, ***ibis*** and ***hen*** frequently?
duck and ***koala*** frequently?


turtle
ibis
duck
hen
Frequent:
A: {turtle, ibis, duck, hen}
Infrequent
B: {koala}
Frequent: {ibis} ⊂ A
Frequent: {turtle, duck} ⊂ A
Frequent: {turtle, ibis, hen} ⊂ A
Infrequent: B ⊂{duck, koala}

Properties of frequent itemset

- The Apriori Principle

If an itemset is frequent, then all of its subsets must also be frequent.

If {*turtle*, *ibis*, *duck*, *hen*} is frequent, then {*ibis*}, {*turtle*, *duck*}, {*turtle*, *ibis*, *hen*} are also frequent.

If an itemset if not frequent, then all its supersets are not frequent either.

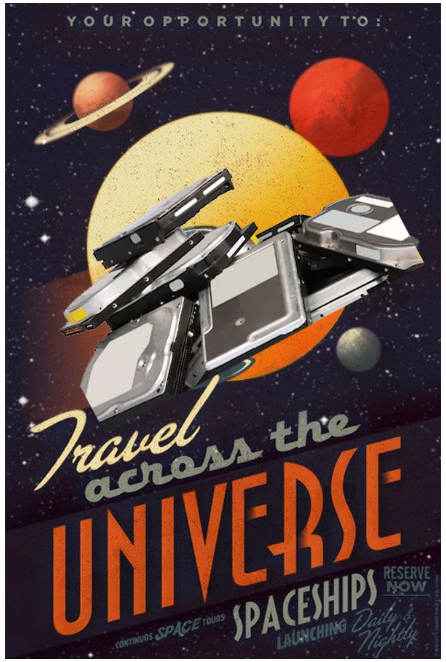
If {*koala*} is not frequent, then {*duck, koala*} is not frequent.

If the itemset {*item1, item2, …, item100*} is a frequent itemset, how many frequent itemsets does it contain?

$$2^{100} - 1 \approx 1.27 \times 10^{30}$$

If we put them in excel files, and store the files in 1T hard disk, the height of all disks should be $\mathbf{10^{13}}$ ***km****.*

1 light year = $9.46 \times 10^{12} km$

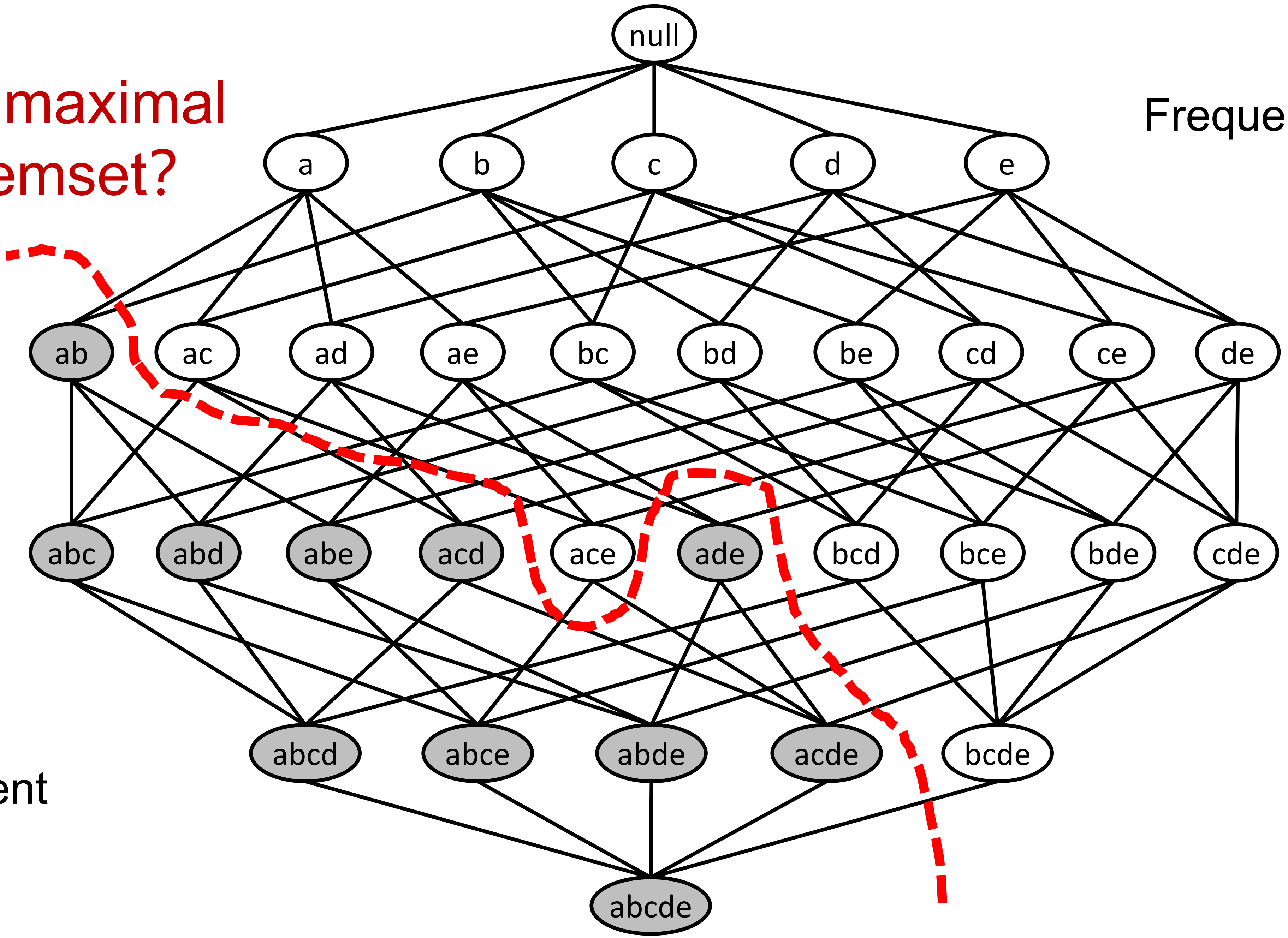


Compact representation for frequent itemset

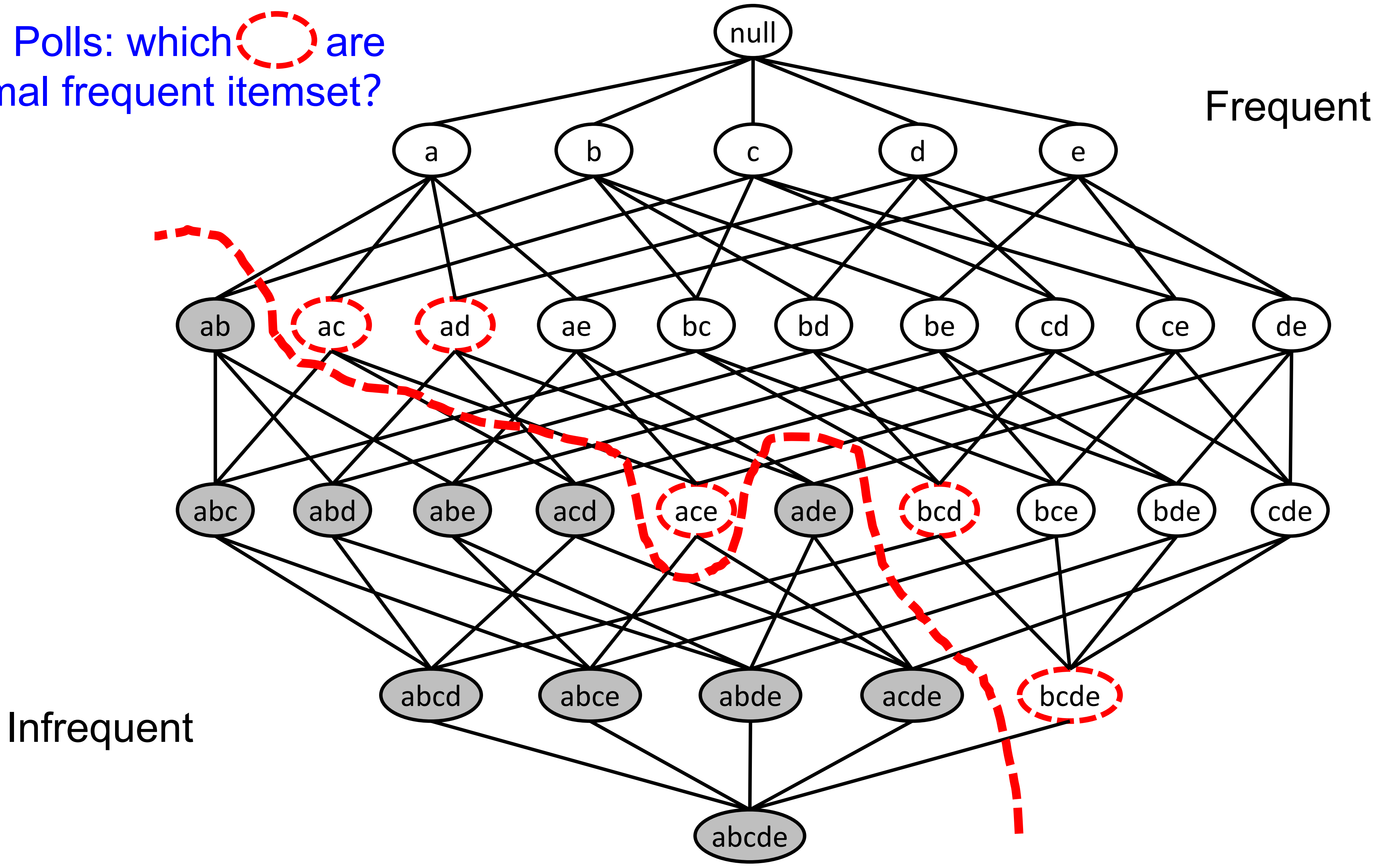
- Maximal frequent itemset (max-itemset)

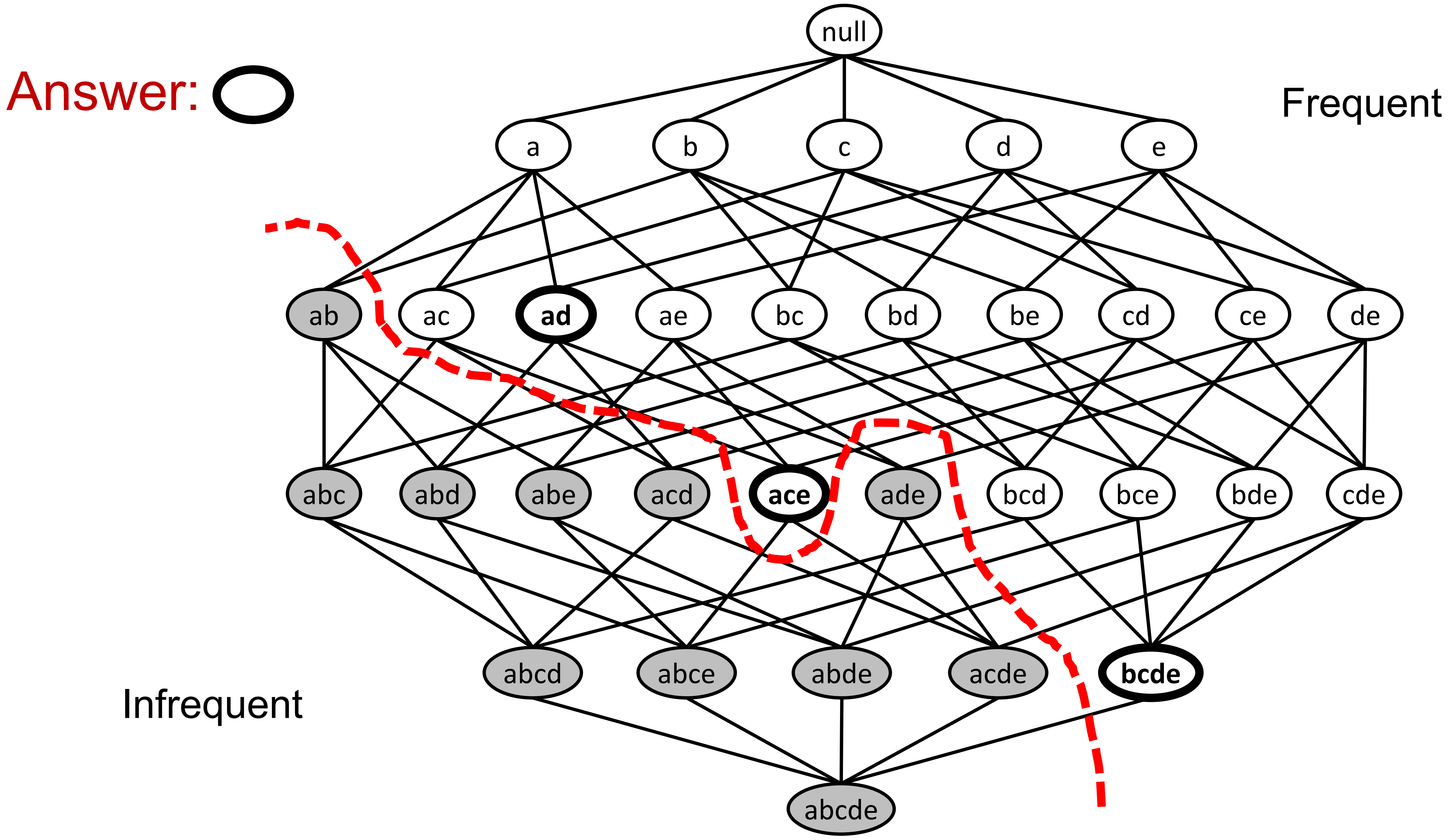
A frequent itemset is maximal if none of its supersets is frequent.

Which are maximal frequent itemset?



Zoom Polls: which are maximal frequent itemset?




Answer:
Frequent
Infrequent
null
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

- Given
  - A database of transactions
  - Each transaction is a list of items

- Find
  - Frequent items purchased together
  - A rule that buying some items will lead to the buying of others

Association rule

- An implication of the form

$$A \Rightarrow B \text{ or } A \rightarrow B$$

– A, B: non-empty itemset

– $A \cap B = \emptyset$

$$\{\text{Bread, Milk}\} \Rightarrow \{\text{Diapers}\}$$

$$\{\text{Diapers}\} \Rightarrow \{\text{Beer}\}$$

TID	**Items**
1	{Bread, Milk}
2	{Bread, Diapers, Beer, Eggs}
3	{Milk, Diapers, Beer, Cola}
4	{Bread, Milk, Diapers, Beer}
5	{Bread, Milk, Diapers, Cola}

Measurement on rules

- Support
  - Equals to support on itemset $A \cup B$

minimum support threshold

- Confidence
  - If A, what is the possibility of B?

$$\textbf{conf}(A \Rightarrow B) = \frac{\#\{A,B\}}{\#\{A\}}$$

  - {Diapers} ⇒ {Beer}: 3/4 = 0.75 = 75%
  - {Bread, Milk} ⇒ {Beer}: 1/3 = 0.333 = 33.3%

minimum confidence threshold

TID	Items
1	{Bread, Milk}
2	{Bread, **Diapers**, Beer, Eggs}
3	{Milk, **Diapers**, Beer, Cola}
4	{Bread, Milk, **Diapers**, Beer}
5	{Bread, Milk, **Diapers**, Cola}

$$A \Rightarrow B$$

{turtle, duck} → {koala}

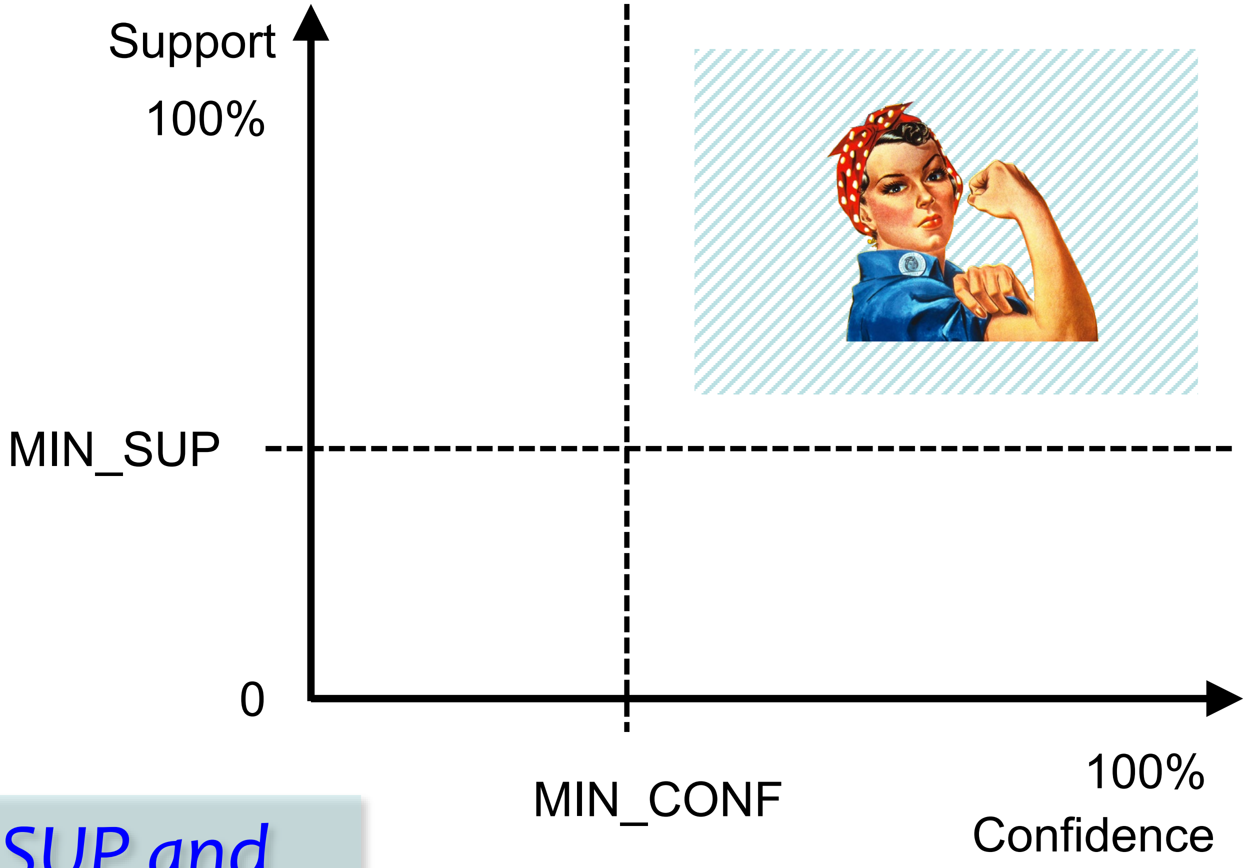
{turtle, koala} → {duck}

Activity: calculate the support and confidence

https://docs.google.com/spreadsheets/d/1Bcdr6FftEdo3UhwytRB9B3V2YDgRPWoG0uN2707OuF0/edit?usp=sharing

Strong association rule

- User-specified threshold
  - minimum support
    MIN_SUP
  - minimum confidence
    MIN_CONF



Rules that satisfy both MIN_SUP and MIN_CONF are called strong rules.

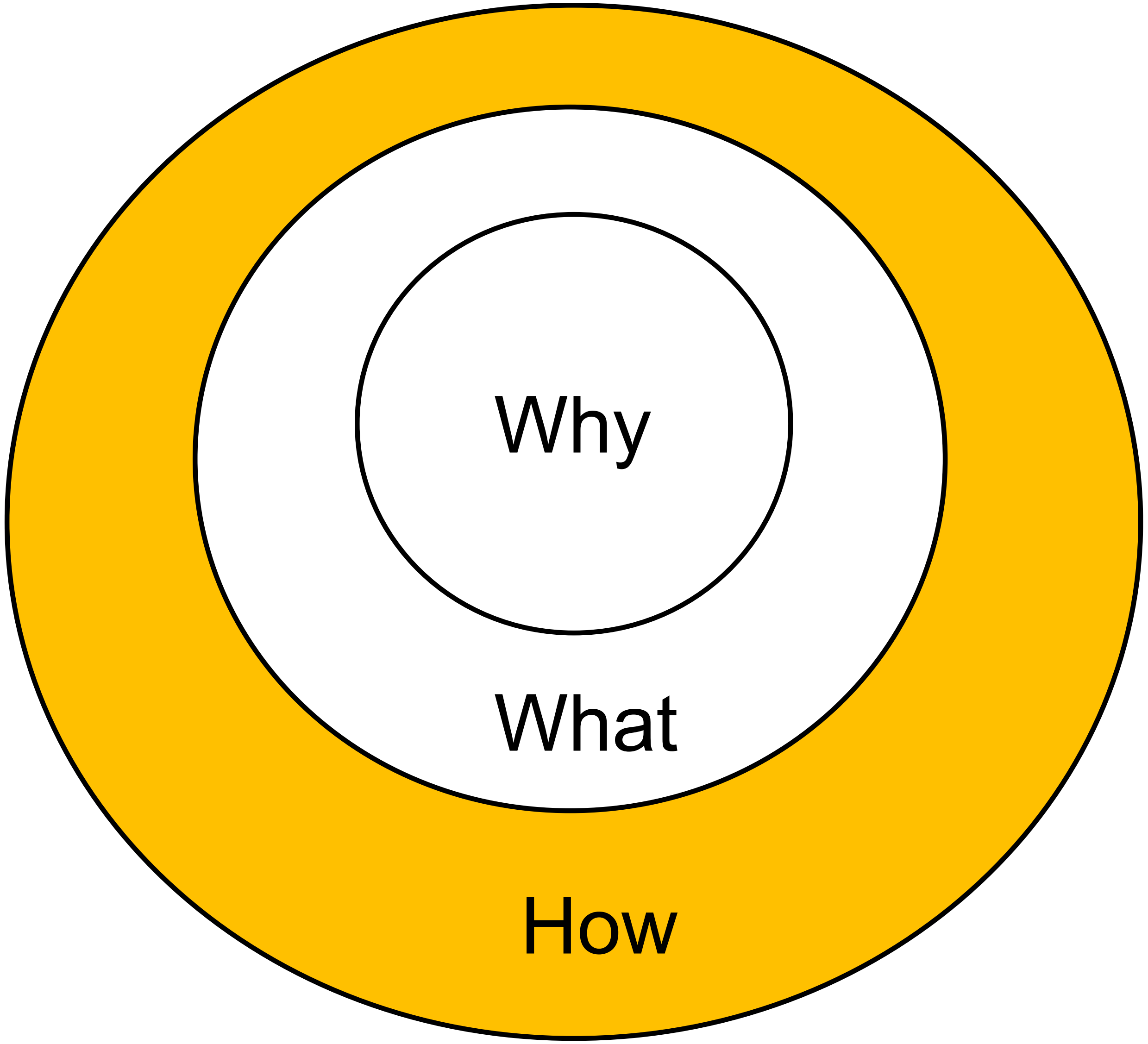
Question: true/false

If MIN_SUPP represents the minimum value threshold for support rate, then MIN_CONF needs to be larger than MIN_SUPP to make the MIN_CONF meaningful?

$$\mathrm{conf}(A \Rightarrow B) = \frac{\#\{\boldsymbol{A}, \boldsymbol{B}\}}{\#\{\boldsymbol{A}\}} \geq \mathrm{supp}(A \Rightarrow B) = \frac{\#\{\boldsymbol{A}, \boldsymbol{B}\}}{\#\boldsymbol{T}} \geq \mathrm{MIN\_SUPP}$$

Association rule mining

Given a set of transactions, find all strong rules showing the association relation between items.


Why
What
How

A naïve solution

- List all possible association rules
- Compute for each rule the
  - *s*: support
  - *c* : confidence
- Keep strong rules

TID	Items
1	{Bread, Milk}
2	{Bread, Diapers, Beer, Eggs}
3	{Milk, Diapers, Beer, Cola}
4	{Bread, Milk, Diapers, Beer}
5	{Bread, Milk, Diapers, Cola}

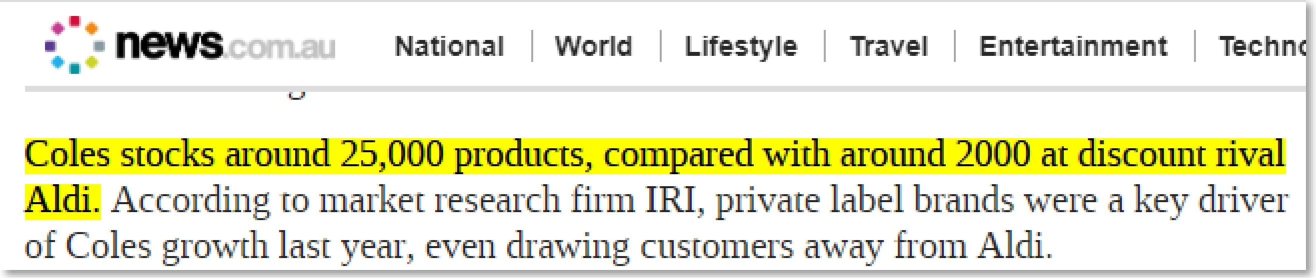
How many rules are there for three items? 12

How many rules are there for five items? 180

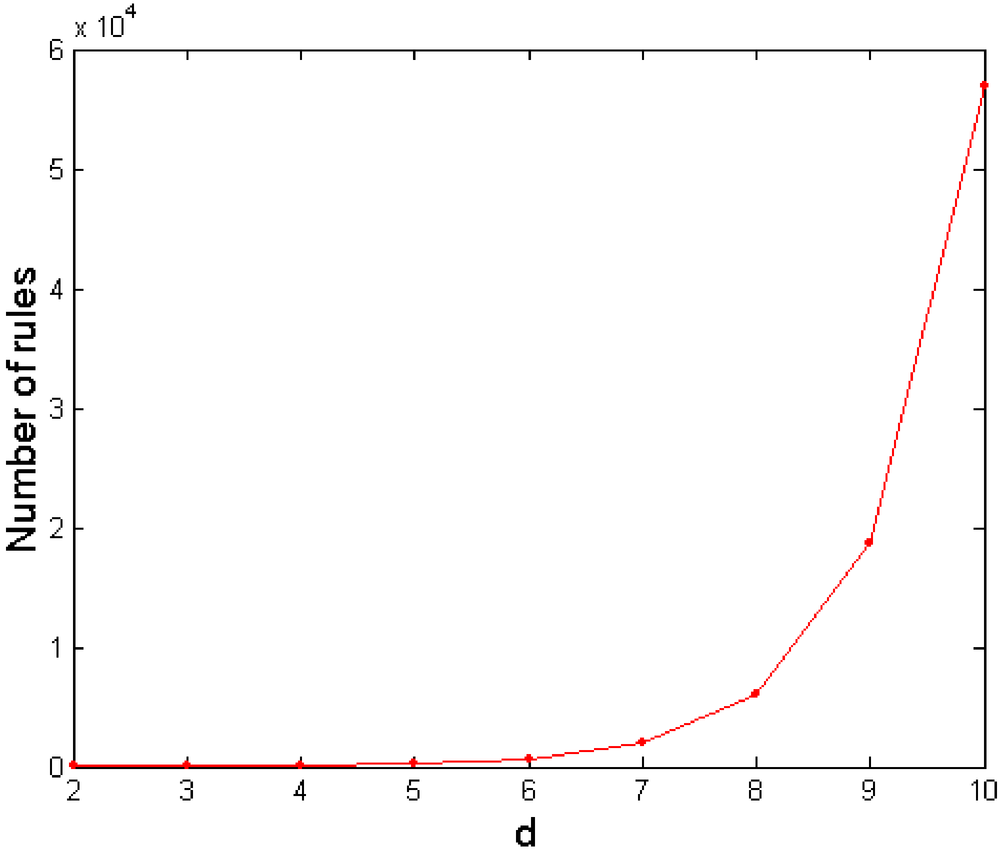
How many rules are there for six items? 602

How many rules are there for d items?

$$\sum_{k=2}^{d} C_d^k (2^k - 2) = 3^d - 2^{d+1} + 1$$


news.com.au National | World | Lifestyle | Travel | Entertainment | Techno

Coles stocks around 25,000 products, compared with around 2000 at discount rival Aldi. According to market research firm IRI, private label brands were a key driver of Coles growth last year, even drawing customers away from Aldi.



Observation

TID	**Items**
1	{Bread, Milk}
2	{Bread, Diapers, Beer, Eggs}
3	{Milk, Diapers, Beer, Cola}
4	{Bread, Milk, Diapers, Beer}
5	{Bread, Milk, Diapers, Cola}

Example of Rules:

{Milk, Diaper} ⇒ {Beer} (s=0.4, c=0.67)
{Milk, Beer} ⇒ {Diaper} (s=0.4, c=1.0)
{Diaper, Beer} ⇒ {Milk} (s=0.4, c=0.67)
{Beer} ⇒ {Milk, Diaper} (s=0.4, c=0.67)
{Diaper} ⇒ {Milk, Beer} (s=0.4, c=0.5)
{Milk} ⇒ {Diaper, Beer} (s=0.4, c=0.5)

- These rules are all partitions of {Milk, Diaper, Beer}.
- They have the same support but different confidence.

*Let's **decouple** the support and confidence requirement!*

A two-step approach

1. Frequent Itemset Generation
   - Generate all itemsets with

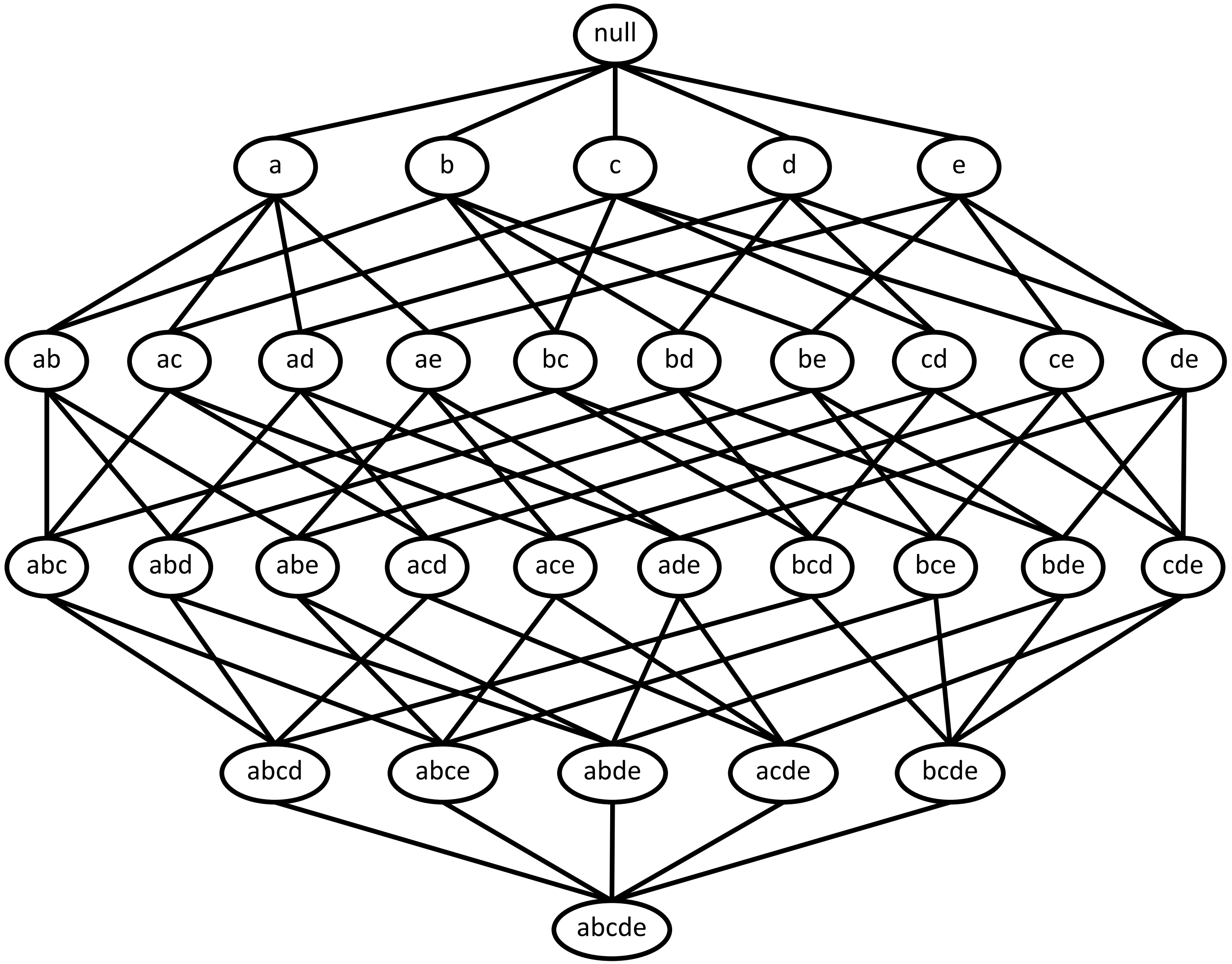
   support $\geq$ MIN_SUP

2. Strong rule Generation
   - Partition each frequent itemset into two parts
   - Generate rules from each partition and keep those with

   confidence $\geq$ MIN_CONF

Step 1: frequent itemset generation

- Brute force
  - Generate all itemsets
    - How much? $2^d - 1$
  - For each itemset:
    - Calculate its support by scanning all data



Step 1: frequent itemset generation

- Brute force
  - Generate all itemsets
    - How much? $2^d - 1$
  - For each itemset:
    - Calculate its support by scanning all data

n

TID	Items
1	{Bread, Milk}
2	{Bread, Diapers, Beer, Eggs}
3	{Milk, Diapers, Beer, Cola}
4	{Bread, Milk, Diapers, Beer}
5	{Bread, Milk, Diapers, Cola}

m

Time complexity? $O(nm2^d)$

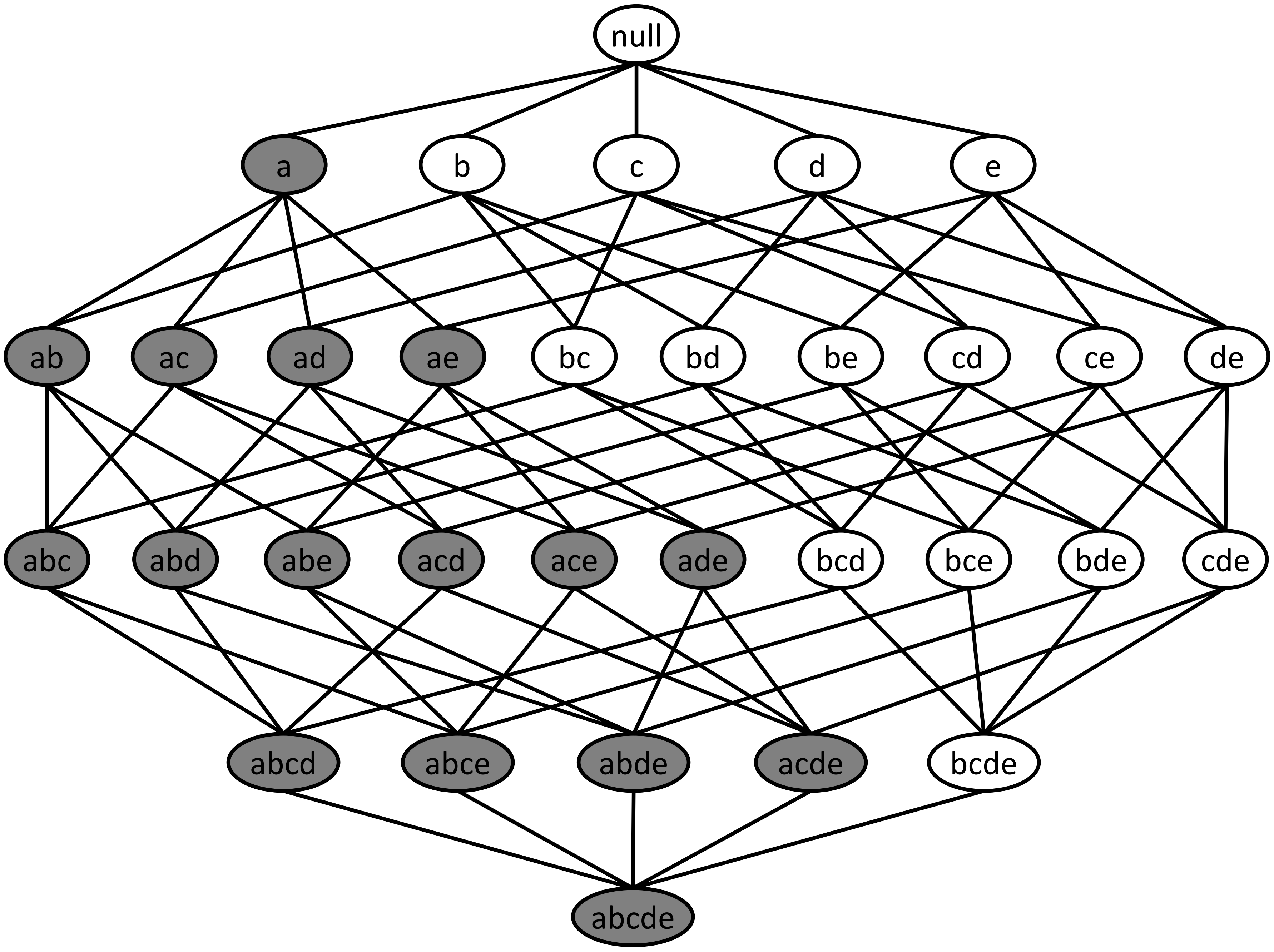
Segment overview

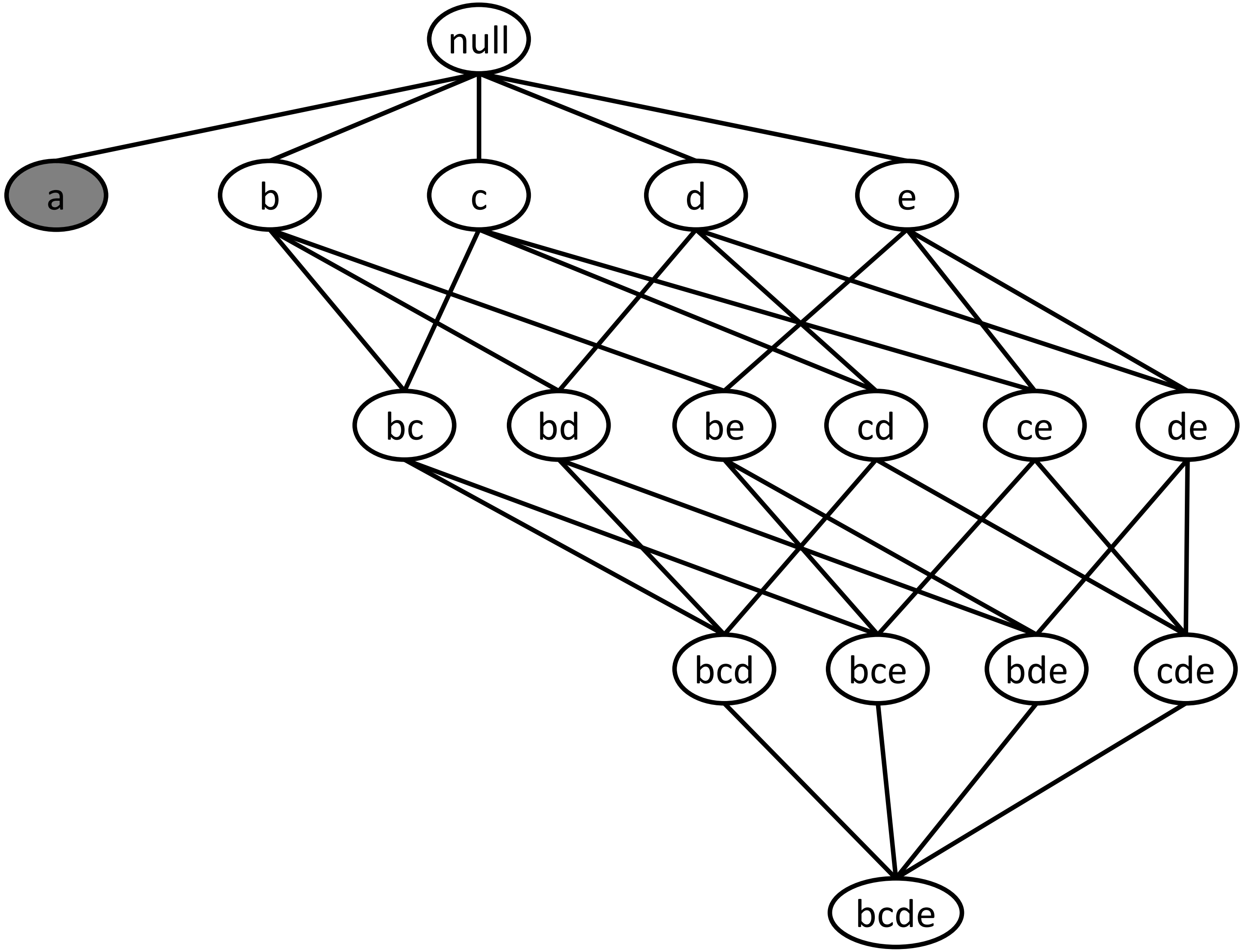
$ MILLION	FY19	FY18	CHANGE
Sales revenue	30,992.6	30,018.2	3.2%
EBIT	1,191.4	1,171.9	1.7%
EBIT margin (%)	3.8	3.9	(6bps)

The Apriori Principle for step 1

If an itemset is frequent, then all of its subsets must also be frequent.

If an itemset if not frequent, then all its supersets are not frequent either.


null
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


null
a
b
c
d
e
bc
bd
be
cd
ce
de
bcd
bce
bde
cde
bcde

- F_k: frequent k-itemsets
- L_k: candidate k-itemsets

- Algorithm
  - Let k=1
  - Generate F_1 = {frequent 1-itemsets}
  - Repeat until F_k is empty
    - **Candidate Generation**: Generate L_{k+1} from F_k
    - **Support Counting**: Count the support of each candidate in L_{k+1}
    - **Candidate Elimination**: leaving only those that are frequent to F_{k+1}

TID	**Items**
1	{Bread, Milk}
2	{Bread, Diapers, Beer, Eggs}
3	{Milk, Diapers, Beer, Cola}
4	{Bread, Milk, Diapers, Beer}
5	{Bread, Milk, Diapers, Cola}

L_1

1-itemset	**s**
Bread	4
Milk	4
Diapers	4
Beer	3
Cola	2
Eggs	1

F_1

1-itemset
Bread
Milk
Diapers
Beer

Generate L_2 from F_1

2-itemset	**s**
Bread, Milk	3
Bread, Diapers	3
Bread, Beer	2
Milk, Diapers	3
Milk, Beer	2
Diapers, Beer	3

MIN_SUP count = 3

F_2

2-itemset
Bread, Milk
Milk, Diapers
Diapers, Beer

Generated L_3 from F_2

F_3

3-itemset
NULL

Candidate generation

If an itemset is frequent, then all of its subsets must also be frequent.

F_2

2-itemset
Bread, Milk
Milk, Diapers
Diapers, Beer

+

F_1

1-itemset
Bread
Milk
Diapers
Beer

3-itemset	s
Bread, Milk, Diapers	2
Bread, Milk, Beer	1
Beer, Milk, Diapers	2
Bread, Diapers, Beer	2

TID	Items
1	{Bread, Milk}
2	{Bread, Diapers, Beer, Eggs}
3	{Milk, Diapers, Beer, Cola}
4	{Bread, Milk, Diapers, Beer}
5	{Bread, Milk, Diapers, Cola}

MIN_SUP count = 3

Candidate generation and pruning

If an itemset is frequent, then all of its subsets must also be frequent.

F_2

2-itemset
Bread, Milk
Milk, Diapers
Diapers, Beer

+

F_1

1-itemset
Bread
Milk
Diapers
Beer

3-itemset
Bread ... Diap
Bread, Mi
Beer, ... Diaper
Bread, Diapers, Beer

Pruning: check if all subsets of these 3-itemset are in F2

TID	Items
1	{Bread, Milk}
2	{Bread, Diapers, Beer, Eggs}
3	{Milk, Diapers, Beer, Cola}
4	{Bread, Milk, Diapers, Beer}
5	{Bread, Milk, Diapers, Cola}

L_1

1-itemset	s
Bread	4
Milk	4
Diapers	4
Beer	3
Cola	2
Eggs	1

F_1

1-itemset
Bread
Milk
Diapers
Beer
Cola

Generate L_2 from F_1

2-itemset	s	2-itemset	s
Bread, Milk	3	Bread, Cola	1
Bread, Diapers	3	Milk, Cola	2
Bread, Beer	2	Diapers, Cola	2
Milk, Diapers	3	Beer, Cola	1
Milk, Beer	2	Diapers, Beer	3

MIN_SUP count = **2**

F_2

2-itemset	2-itemset
Bread, Milk	Milk, Cola
Bread, Diapers	Diapers, Cola
Bread, Beer	Diapers, Beer
Milk, Diapers	
Milk, Beer	

Generated L_3 from F_2

F_2

2-itemset
Beer, Bread
Beer, Diapers
Beer, Milk
Bread, Cola
Bread, Diapers
Bread, Milk
Cola, Diapers
Cola, Milk
Diapers, Milk

F_1

1-itemset
Beer
Bread
Cola
Diapers
Milk

L_3

3-itemset
Beer, Bread, Cola
Beer, Bread, Diapers
Beer, Bread, Milk
Beer, Diapers, Milk
Bread, Cola, Diapers
Bread, Cola, Milk
Bread, Diapers, Milk
Cola, Diapers, Milk

F_2

2-itemset
Beer, Bread
Beer, Diapers
Beer, Milk
Bread, Cola
Bread, Diapers
Bread, Milk
Cola, Diapers
Cola, Milk
Diapers, Milk

F_1

1-itemset
Beer
Bread
Cola
Diapers
Milk

L_3

3-itemset
Beer, Bread, Cola
Beer, Bread, Diapers
Beer, Bread, Milk
Beer, Diapers, Milk
Bread, Cola, Diapers
Bread, Cola, Milk
Bread, Diapers, Milk
Cola, Diapers, Milk

Pruning

L_3

3-itemset
Beer, Bread, Diapers
Beer, Bread, Milk
Beer, Diapers, Milk
Bread, Cola, Diapers
Bread, Cola, Milk
Bread, Diapers, Milk
Cola, Diapers, Milk

Smarter way to generate L3?

Candidate generation by “join”

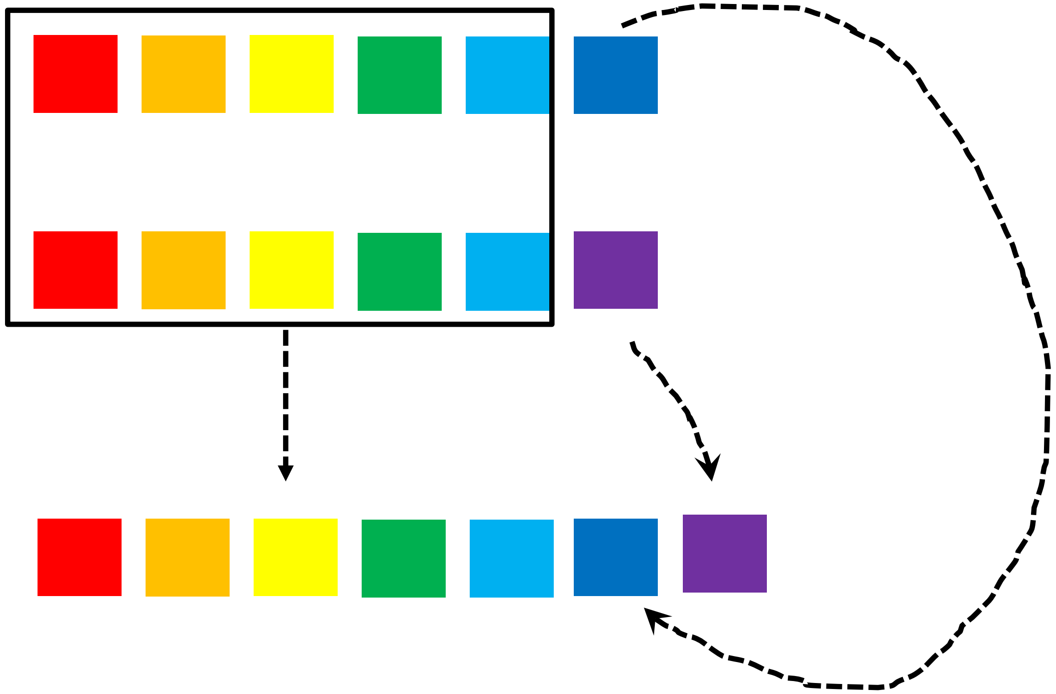
- Sorting the items in each frequent k-itemset
  - Lexicographic order
- If two frequent k-itemsets
  - share the first (k-1) items
  - different in the last item
- Concatenate the first (k-1) items and the last two items
  - A candidate (k+1)-itemset is generated.

- Illustration of a “join” operation

Frequent 6-itemset 1:

Frequent 6-itemset 2:

Join 1 and 2:



F_2

2-itemset
Beer, Bread
Beer, Diapers
Beer, Milk
Bread, Cola
Bread, Diapers
Bread, Milk
Cola, Diapers
Cola, Milk
Diapers, Milk

L_3

3-itemset
Beer, Bread, Diapers
Beer, Bread, Milk
Beer, Diapers, Milk
Bread, Cola, Diapers
Bread, Cola, Milk
Bread, Diapers, Milk
Cola, Diapers, Milk

Pruning

L_3

3-itemset
Beer, Bread, Diapers
Beer, Bread, Milk
Beer, Diapers, Milk
Bread, Cola, Diapers
Bread, Cola, Milk
Bread, Diapers, Milk
Cola, Diapers, Milk

Why we only need to do the joining operation to generate L_{k+1}, instead of merging every two frequent k-itemsets in F_k?

Hints:

1. Does the join operation generate duplicated candidates?
2. Will all frequent (k+1)-itemsets be contained within the generated L_{k+1}

Please find the answer by yourself and share/discuss on Piazza!

Summarization of Step 1

- The Apriori Algorithm
  - Let k=1
  - Generate F_1 = {frequent 1-itemsets}
  - Repeat until F_k is empty
    - **Candidate Generation**: Generate L_{k+1} from by Join operation on F_k
    - **Candidate Pruning**: Prune candidate itemsets in L_{k+1} containing subsets of length k that are infrequent
    - **Support Counting**: Count the support of each candidate in L_{k+1} by scanning the DB
    - **Candidate Elimination**: Eliminate candidates in L_{k+1} that are infrequent, leaving only those that are frequent => F_{k+1}

Discussion

- What are the factors impact the efficiency of Apriori algorithm?
  - Thresholds
  - Number of items
  - Number of transactions
  - Each transaction length

A two-step approach

1. Frequent Itemset Generation

   - Generate all itemsets with

   support $\geq$ MIN_SUP

2. Strong rule Generation

   - Partition each frequent itemset into two parts

   - Generate rules from each partition and keep those with

   confidence $\geq$ MIN_CONF

Step 2: Rule generation

- Algorithm
  - For each frequent itemset
    - For each partition {part1, part2} of the frequent itemset
      - **Candidate Generation**: Generate rule {part1⇒part2} and {part2 ⇒ part1}
      - **Confidence Counting**: Count the confidence of the rules by
        conf(part1⇒part2) = support(part1, part2)/support(part1)
        conf(part2 ⇒ part1) = support(part1, part2)/support(part1)
      - **Candidate Elimination**: leaving only those that are strong

– If {A,B,C,D} is a frequent itemset, partitions are:

{A, BCD}, {B, ACD}, {C, ABD}, {D, ABC}

{AB, CD}, {AC, BD}, {AD, BC}

– candidate rules:

$\{A \Rightarrow BCD\}$, $\{B \Rightarrow ACD\}$, $\{C \Rightarrow ABD\}$, $\{D \Rightarrow ABC\}$

$\{BCD \Rightarrow A\}$, $\{ACD \Rightarrow B\}$, $\{ABD \Rightarrow C\}$, $\{ABC \Rightarrow D\}$

$\{AB \Rightarrow CD\}$, $\{AC \Rightarrow BD\}$, $\{AD \Rightarrow BC\}$

$\{CD \Rightarrow AB\}$, $\{BD \Rightarrow AC\}$, $\{BC \Rightarrow AD\}$

– If $|F| = k$, then there are $2^k - 2$ candidate association rules (ignoring $L \rightarrow \varnothing$ and $\varnothing \rightarrow L$)

TID	Items
1	{Bread, Milk}
2	{Bread, Diapers, Beer, Eggs}
3	{Milk, Diapers, Beer, Cola}
4	{Bread, Milk, Diapers, Beer}
5	{Bread, Milk, Diapers, Cola}

F_3

3-itemset	s
Beer, Bread, Diapers	2
Beer, Diapers, Milk	2
Bread, Diapers, Milk	2
Cola, Diapers, Milk	2

MIN_SUP count = 2 MIN_CONF= 60%

	support	confidence
Beer→ Bread, Diapers	40%	66.7%
Bread, Diapers→ Beer	40%	66.7%
Bread→ Beer, Diapers	40%	50%
Beer, Diapers→ Bread	40%	66.7%
Diapers→ Beer, Bread	40%	50%
Beer, Bread→ Diapers	40%	100%

TID	Items
1	{Bread, Milk}
2	{Bread, Diapers, Beer, Eggs}
3	{Milk, Diapers, Beer, Cola}
4	{Bread, Milk, Diapers, Beer}
5	{Bread, Milk, Diapers, Cola}

F_3

3-itemset	s
Beer, Bread, Diapers	2
Beer, Diapers, Milk	2
Bread, Diapers, Milk	2
Cola, Diapers, Milk	2

MIN_SUP count = 2 MIN_CONF= 60%

	support	confidence
Beer→ Bread, Diapers	40%	66.7%
Bread, Diapers→ Beer	40%	66.7%
Bread→ Beer, Diapers	40%	50%
Beer, Diapers→ Bread	40%	66.7%
Diapers→ Beer, Bread	40%	50%
Beer, Bread→ Diapers	40%	100%

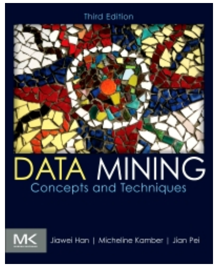
Summary

Association Rule Mining

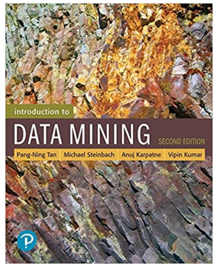
- Why
  - Market basket analysis, text, bioinformatics…
- What
  - Frequent itemset: high support
  - Strong rule: high support and high confidence
- How
  - The Apriori Algorithm
  - Rule generation

Recommended reading (not required)

- [Han et al., 2012]
  - Sec. 6.1-6.2



- [Tan et al., 2019]
  - Sec. 5.1-5.4



- [Aggarwal, 2015]
  - Sec. 4.1-4.4