Advanced Topics in Information systems

SE204

Lecture 4

Dr. Nelly Amer

Data mining.

- **What is data mining?**
- **Why data mining?**
- **Data Mining steps.**
- **Data analysis, Data Mining and Data science.**
- **Data Mining Models.**
- **Frequent pattern mining.**

What is data mining?

Data mining is the process of discovering hidden patterns, correlations, trends, and knowledge from large amounts of data .

The data sources can include databases, data warehouses, the Web, other information repositories, or data that are streamed into the system dynamically.

Examples:

- In market basket analysis can discover that customers who buy bread are also likely to purchase milk.
- Watch a cooking video, and YouTube suggests more recipes.
- Facebook show ads based on what you like.

Data mining.

Not all information discovery tasks are considered to be data mining, Examples include queries, e.g., looking up individual records in a database or finding web pages that contain a particular set of keywords. This is because such tasks can be accomplished through simple interactions with a database management system or an information retrieval system. These systems rely on traditional computer science techniques, which include query processing algorithms, for retrieving information from data, **while data mining** focus on discovering **hidden, interesting, and useful patterns**, or **hidden correlations** in data, **not Simple queries, reports, or statistics that retrieve existing information**.

≠

Why data mining.

traditional data analysis techniques have often encountered practical difficulties in meeting the challenges posed by big data applications. The following are some of the specific challenges that motivated the development of data mining.

- Scalability
- High Dimensionality
- Heterogeneous
- Complex Data
- Data Ownership and Distribution
- Non-traditional Analysis

Why data mining.

Scalability

Because of advances in data generation and collection, data sets with sizes of terabytes, petabytes, or even Exabyte are becoming common. If data mining algorithms are to handle these massive data sets, they must be scalable.

Many data mining algorithms employ special search strategies to handle exponential search problems.

High Dimensionality

It is now common to encounter data sets with hundreds or thousands of attributes instead of the handful common a few decades ago, and Traditional data analysis techniques that were developed for low-dimensional data often do not work well for such high-dimensional data.

Examples:

1- In bioinformatics, progress in microarray technology has produced gene expression data involving thousands of features.

2- Data sets with spatial components also tend to have high dimensionality. For example, consider a data set that contains measurements of temperature at various locations. If the temperature measurements are taken repeatedly for an extended period, the number of dimensions (features) increases in proportion to the number of measurements taken

Why data mining.

Heterogeneous

Traditional data analysis methods often deal with data sets containing attributes of the same type, either continuous or categorical. As the role of data mining in business, science, medicine, and other fields has grown, so has the need for techniques that can handle heterogeneous attributes.

Complex Data

Recent years have also seen the emergence of more complex data objects. Examples of such non-traditional types of data include **web and social media** data containing text, hyperlinks, images, audio, videos, DNA data with complex structure, and climate data that consists of measurements (temperature, pressure, etc.) at various times and locations on the Earth's surface.

Why data mining.

Data Ownership and Distribution

Sometimes, the data needed for an analysis is not stored in one location or owned by one organization. Instead, the data is geographically distributed among resources belonging to multiple entities. This requires the development of distributed data mining techniques. The key challenges faced by distributed data mining algorithms include the following:

(1) how to reduce the amount of communication needed to perform the distributed computation.

(2) how to effectively consolidate the data mining results obtained from multiple sources.

(3) how to address data security and privacy issues.

Why data mining.

Non-traditional Analysis

The traditional statistical approach is based on a hypothesize-and-test paradigm. In other words, a hypothesis is proposed, an experiment is designed to gather the data, and then the data is analyzed with respect to the hypothesis. Unfortunately, this process is extremely labor-intensive. Current data analysis tasks often require the generation and evaluation of thousands of hypotheses, and consequently, the development of some data mining techniques has been motivated by the desire to automate the process of hypothesis generation and evaluation.

Data Mining steps

There are seven steps as the following :

1- Data cleaning

Data cleaning: it is the process to handle noise, incomplete and inconsistent data.

Incomplete. When some of the attribute values are missed.

Noisy. When data contains errors or outliers, for example, some of the data points in a dataset may contain extreme values that can severely affect the dataset's range.

Inconsistent. refers to the lack of uniformity in data format or content, for example if records do not start with a capital letter, discrepancies are present.

2- **Data integration**

The multiple data sources are **combined from different databases**. It is the mixture of heterogeneous and homogeneous types of data which gets stored in data warehouse.

3-Data selection

Data relevant to the analysis task are retrieved from the database.

Data Mining steps

4-Data transformation

In this data transformed into forms appropriate for mining by performing summary or aggregation operations.

5-Data mining

An essential process where intelligent methods are applied in order to extract data patterns

6-Pattern evaluation

To identify the truly interesting patterns representing knowledge base on some interesting measures.

7-Knowledge

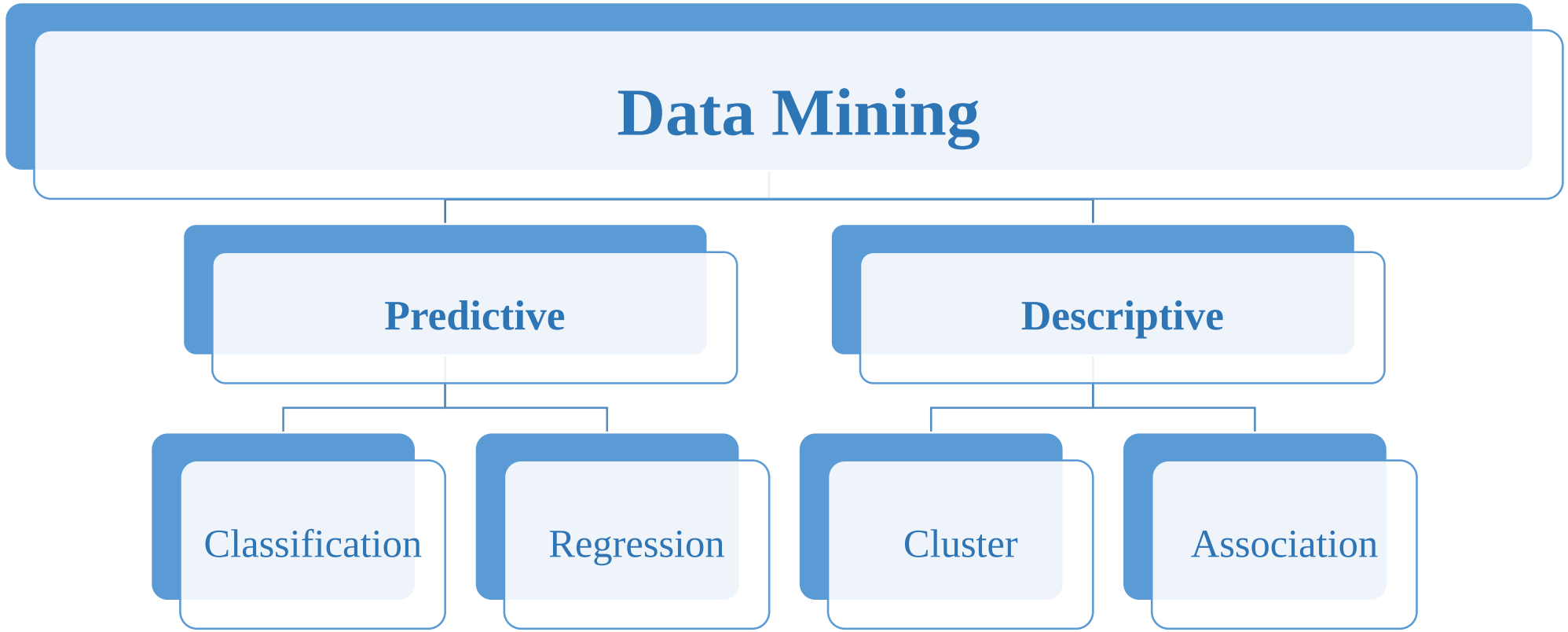
Where visualization & knowledge representation techniques are used to present the mined knowledge to the user.

Data analysis , Data Mining and Data science.

What is the difference?

Data Mining models.

Data mining tasks are generally divided into two major categories:



Data Mining models

Predictive models:

The objective of these models is to **predict the value of a particular attribute based on the values of other attributes**.

The attribute to be predicted is commonly known as the target or **dependent** variable, while the attributes used for making the prediction are known as the **independent** variables.

There are **two types of predictive modeling tasks**: classification, which is used for discrete target variables, and regression, which is used for continuous target variables.

Descriptive models

the objective is to find **human-interpretable patterns** (correlations, trends, clusters) that describe the data.

Predictive models

In **classification**, the goal is to assign input data to specific, predefined categories. The output classification is typically a label or a class from a set of predefined options.

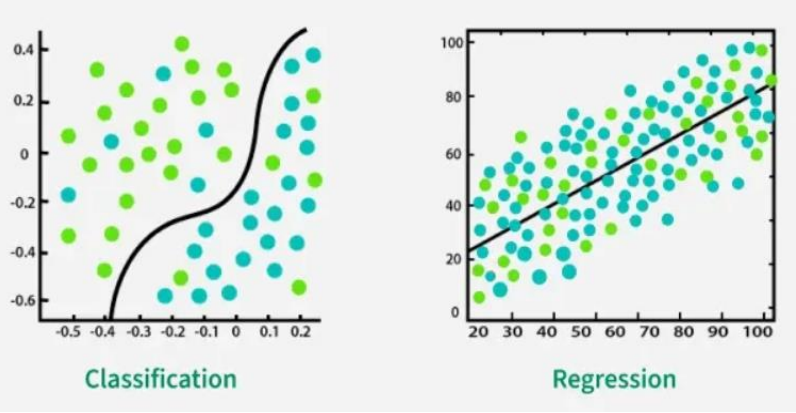
In **regression**, the goal is to establish a relationship between input variables and the output. The output in regression is a real-valued number that can vary within a range.

Both approaches **require labeled data for training but differ in their objectives**

classification aims to find decision boundaries that separate classes, whereas regression focuses on finding the best-fitting line to predict numerical outcomes.

Examples on classification:

It can determine whether an email is spam or not, classify images as “cat” or “dog,” or predict weather conditions like “sunny,” “rainy,” or “cloudy.” with decision boundary , classify IRIS flowers.



Examples on regression:

models are used to predict house prices based on features like size and location, or forecast stock prices over time with straight fit line.

Descriptive Model

The descriptive model classifies customers or prospects into groups based on the analyzing relationship between the data as in Clustering, Summarization, Association rule, Sequence discovery, etc.

Clustering:

A cluster is a group of similar objects or data points that share common characteristics, clustering means Finding groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups

Examples:

Group genes and proteins that have similar functionality

Descriptive Model

Association:

Association is used to discover patterns that describe strongly associated features in the data. The discovered patterns are typically represented in the form of implication rules or feature subsets. Because of the exponential size of its search space, the goal of association analysis is to extract the most interesting patterns in an efficient manner. Useful applications of association analysis include finding groups of genes that have related functionality, identifying web pages that are accessed together, or understanding the relationships between different elements of Earth's climate system.

Example (Market Basket Analysis).

TID	*Items*
1	**Bread, Coke, Milk**
2	**Beer, Bread**
3	**Beer, Coke, Diaper, Milk**
4	**Beer, Bread, Diaper, Milk**
5	**Coke, Diaper, Milk**

For example, we may discover the rule {Diapers}→{Milk}, which suggests that customers who buy diapers also tend to buy milk.

Frequent Pattern

A **frequent pattern** are patterns that occur frequently in data.

Types of Frequent Patterns:

Frequent Itemsets : **A group of items that appear together** often in transactions.
Example: In a supermarket, **{Milk, Bread}** frequently appear in customer purchases.

Frequent Sequential Patterns : A sequence of **events** that frequently occur in a specific order.
Example: A customer often buys **Laptop → Mouse → Keyboard** in that order.

Frequent structures –Repeated patterns in chemical and biological data, XML data, software program traces, and Web browsing behaviors , and GPS data.

We will focus on frequent Itemsets

Frequent Pattern mining

Frequent patterns mining: is a Data Mining subject with the objective of **extracting frequent itemsets from a dataset**.

It leads to the discovery of interesting associations and correlations within data.

The uses of frequent pattern mining:

Recommendation Systems (e.g., Amazon’s "Customers also bought...")

Market Basket Analysis (e.g., Which items are bought together?)

Fraud Detection (e.g., Finding unusual spending patterns)

Medical Diagnosis (e.g., Common symptoms in a disease)

Basic concepts

Itemset

A collection of one or more items

Example: {Milk, Bread, Diaper}

k-itemset

An itemset that contains k items

Support count (σ)

Frequency of occurrence of an itemset

σ({Milk, Bread, Diaper}) = 2

Support

Frequency of occurrence of an itemset / all transaction

s({Milk, Bread, Diaper}) = 2/5

TID	*Items*
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Frequent Itemset

Frequent Itemset

An itemset whose support is greater than or equal to a minsup threshold

Example:

As seen in the following transaction table, given the minsup = 60%, which of the following itemset is frequent:

{Bread}, {Eggs}, {Bread, Diaper}, {Bread, Milk}, {Bread, Milk, Diaper}

TID	*Items*
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Solution

Itemset	Support
{Bread}	4/5=80%
{Eggs}	1/5=20%
{Bread, Diaper}	3/5=60%
{Bread, Milk}	3/5=60%
{Bread, Milk, Diaper}	2/5=40%

Since minsup = 60%, then {Bread}, {Bread, Diaper}, {Bread, Milk} are frequent itemsets

, and {Eggs}, {Bread, Milk, Diaper} are not frequent itemsets

Superset

A superset is a set that contains **all elements of another set**, along with possibly more elements.

If an **itemset X** contains all elements of **itemset Y**, then **X is a superset of Y**.

Example. 1:

Given the set {A, B, C, D}, what are the supersets of {C}

Solution

supersets of {C} are:

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

Example. 2:

Given the set {A, B, C, D}, what are the supersets of {A, B}

Solution

supersets of {A, B} are:

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

closed frequent itemset

A **closed frequent itemset:** is a frequent **itemset** that doesn't have a **superset** with the **same support** in the dataset.

Given a frequent itemset, how do you check if it is closed frequent Item set or not, follow the following steps:

1-Find all supersets of this frequent itemset

2-Compute the support of each superset (support=frequency of the set/the number of transactions)

3-

if any superset (larger itemset) has the same support, then the frequent itemset is not closed.

if all supersets (larger itemset) have not the same support, then the frequent itemset is closed.

closed frequent itemset example.1

Given the transactions in the table, determine if the frequent itemset {**Milk**} is closed frequent itemset or not

Transaction ID	Milk	Bread	Butter	Cheese
1	✓	✓	✓	✗
2	✓	✓	✗	✓
3	✓	✓	✓	✗
4	✗	✓	✓	✓

Solution:

1- find all supersets of the frequent itemset {Milk} :
{Milk, Bread}, {Milk, Butter}, {Milk, Cheese} ,
,{Milk, Bread, Butter} ,{Milk, Bread, Cheese},{Milk, Butter, Cheese},
{Milk, Bread, Butter, Cheese}

2- compute Support for each superset:

Superset	support
{Milk, Bread}	3/4=75%
{Milk, Butter}	2/4=50%
{Milk, Cheese}	1/4=25%
{Milk, Bread, Butter}	2/4=50%
{Milk, Bread, Cheese}	1/4=25%
{Milk, Butter, Cheese}	0/4=0%
{Milk, Bread, Butter, Cheese}	0/4=0%

3-the support of S({**Milk**})= 3/4=75%, **it is clear that** S({**Milk**})= S({**Milk, Bread**})=75%, **Then the frequent itemset** {**Milk**} **is not closed .**

closed frequent itemset example. 2

Given the transactions in the following table, determine if the frequent itemset {**Bread}** is closed frequent itemset or not

Solution:

1- find all supersets of the frequent itemset {Bread} :

{Bread, Milk}, {Bread, Butter}, {Bread, Cheese} ,
,{Bread, Milk, Butter} ,{Bread, Milk, Cheese},{Bread, Butter, Cheese},
{Milk, Bread, Butter, Cheese}

Transaction ID	Milk	Bread	Butter	Cheese
1	✓	✓	✓	✗
2	✓	✓	✗	✓
3	✓	✓	✓	✗
4	✗	✓	✓	✓

2- compute Support for each superset:

Superset	support
{Bread, Milk}	3/4=75%
{Bread, Butter}	3/4=75%
{Bread, Cheese}	2/4=50%
{Bread, Milk, Butter}	2/4=50%
{Bread, Milk, Cheese}	1/4=25%
{Bread, Butter, Cheese}	1/4=0%
{Milk, Bread, Butter, Cheese}	0/4=0%

3-the support of S({**Bread})= 4/4=100 %, it is clear that all supersets have different support**

Then frequent itemset({ Bread} is closed

closed frequent itemset examples.3

Given the transactions in the table, determine if the frequent itemset {**Milk, Bread**} is closed frequent itemset or not

Transaction ID	Milk	Bread	Butter	Cheese
1	✓	✓	✓	✗
2	✓	✓	✗	✓
3	✓	✓	✓	✗
4	✗	✓	✓	✓

Solution:

1- find all supersets of the frequent itemset {Milk, Bread} :

{Milk, Bread, Butter}, {Milk, Bread, Cheese} , {Milk, Bread, Butter, Cheese}

2- compute Support for each superset:

Superset	support
{Milk, Bread, Butter}	2/4=50%
{Milk, Bread, Cheese}	1/4=25%
{Milk, Bread, Butter, Cheese}	0/4=0%

3-the support of S({**Milk, Bread**})= **3/4=75%, it is clear that all supersets have different support**

Then frequent itemset({Milk, Bread} is closed .

Max frequent itemset

A **maximal frequent pattern** (or **max frequent itemset**): is a **frequent itemset that has no superset** that is also frequent.

Given a frequent itemset, how do you check if it is max frequent Item set or not, follow the following steps:

1-Find all supersets of this itemset

2-Compute the support of each superset (support=frequency of the set/the number of transactions)

3-

Checks all superset (larger itemset):

if any superset (larger itemset)is also frequent (i.e. its support >= minsupport), then the frequent itemset is not max.

if all supersets (larger itemset) are not frequent, then the frequent itemset is max.

Max frequent itemset example. 1

Given the transactions in the table, **and minsupport=50%** determine if the frequent itemset {**Milk, Bread**} is Max frequent itemset or not

Transaction ID	Milk	Bread	Butter	Cheese
1	✓	✓	✓	✗
2	✓	✓	✗	✓
3	✓	✓	✓	✗
4	✗	✓	✓	✓

Solution:

1- find all supersets of the frequent itemset {Milk, Bread} :

{Milk, Bread, Butter}, {Milk, Bread, Cheese} , {Milk, Bread, Butter, Cheese}

2- compute Support for each superset:

Superset	support
{Milk, Bread, Butter}	2/4=50%
{Milk, Bread, Cheese}	1/4=25%
{Milk, Bread, Butter, Cheese}	0/4=0%

3- it is clear that the itemset **{Milk, Bread, Butter} is frequent, (because S({Milk, Bread, Butter}=50%=minsupport)**
Then the frequent itemset {Milk, Bread} is not max

Max frequent itemset example. 2

Given the transactions in the table, and **minsupport=60%** determine if the frequent itemset {**Milk, Bread}** is Max frequent itemset or not

Transaction ID	Milk	Bread	Butter	Cheese
1	✓	✓	✓	✗
2	✓	✓	✗	✓
3	✓	✓	✓	✗
4	✗	✓	✓	✓

Solution:

1- find all supersets of the frequent itemset {Milk, Bread} :

{Milk, Bread, Butter}, {Milk, Bread, Cheese} , {Milk, Bread, Butter, Cheese}

2- compute Support for each superset:

Superset	support
{Milk, Bread, Butter}	2/4=50%
{Milk, Bread, Cheese}	1/4=25%
{Milk, Bread, Butter, Cheese}	0/4=0%

3- it is clear that all supersets are not frequent (since all their support is less than the minsupport , then the frequent itemset {Milk, Bread}is max.

Max frequent itemset example. 3

Given the transactions in the table, and minsupport=50% determine if the frequent itemset {**Milk, Bread, Butter}** is Max frequent itemset or not

Transaction ID	Milk	Bread	Butter	Cheese
1	✓	✓	✓	✗
2	✓	✓	✗	✓
3	✓	✓	✓	✗
4	✗	✓	✓	✓

Solution:

1- find all supersets of the frequent itemset {Milk, Bread, Butter} :

{Milk, Bread, Butter, Cheese}

2- compute Support for each superset:

Superset	support
{Milk, Bread, Butter, Cheese}	0/4=0%

3- it is clear that all supersets are not frequent, then the frequent itemset

{**Milk, Bread, Butter}** is max.

Max frequent itemset example.4

Given the transactions in the table, and minsupport=50% determine if the frequent itemset { **Bread, Cheese}** is Max frequent itemset or not

Transaction ID	Milk	Bread	Butter	Cheese
1	✓	✓	✓	✗
2	✓	✓	✗	✓
3	✓	✓	✓	✗
4	✗	✓	✓	✓

Solution:

1- find all supersets of the frequent itemset {Bread, Cheese} :

{ **Bread, Cheese**, Milk}, { **Bread, Cheese** Butter} , {Milk, Bread, Butter, Cheese}

2- compute Support for each superset:

Superset	support
{ Bread, Cheese, Milk}	1/4=25%
{ Bread, Cheese Butter}	1/4=0%
{Milk, Bread, Butter, Cheese}	0/4=0%

3- it is clear that all supersets are not frequent, then the frequent itemset { **Bread, Cheese}** is max.

Assignment #4

1- What is the difference between data analysis, data mining, and data science?

2- Given the following transactions, and the minimum support=60%, determine which of the following itemsets are frequent {a}, {c}, {f}, {e, f}, {a, e, f}

tid	items
1	a, b, c, d
2	b, c, e, f
3	a, d, e, f
4	a, e, f
5	b, d, f

3-Given the transactions in the table, determine if the frequent itemset {**A, B}** is closed frequent itemset or not

Transaction ID	A	B	C	D
1	✓	✓	✓	✗
2	✓	✓	✗	✓
3	✓	✓	✓	✗
4	✗	✓	✓	✓

Assignment #4 cont.

4-Given the transactions in the table, and **minsupport=60%** determine if the frequent itemset {**A, B**} is Max frequent itemset or not

Transaction ID	A	B	C	D
1	✓	✓	✓	✗
2	✓	✓	✗	✓
3	✓	✓	✓	✗
4	✗	✓	✓	✓

References.

[1] P. N. Tan, M. Steinbach , A. Karpatne , V. Kumar, “introduction to data mining”, (2nd edition) , Pearson, 2018.

[2] J. Han, and M. Kamber, “ data mining: concepts and techniques”, 2011.

[3] J. Aguilar-Ruiz, D. Rodríguez -Baena, R. Alves, “frequent pattern mining.” in: W. Ddubitzky, O. Wolkenhauer, K. Hyun Cho, H. Yokota. (eds), encyclopedia of systems biology. springer, 2013.

[4] C. C. Aggarwal, “association pattern mining”, in: “data mining”, springer, 2015.

[5] M. Sharma, “Data Mining Prediction Techniques in Health Care Sector”, Journal of Physics: Conference Series, vol. 2267, pp. 1-9, 2021.

Advanced Topics in Information systems

SE204

Lecture 5

Dr. Nelly Amer

Data mining.

- **Association rules**
- **Apriori algorithm**

Association rules

An association rule is **an implication expression of the form X→Y**

, where X and Y are disjoint itemsets, i.e., X∩Y=∅.

It represents the relationships between two itemsets.

For example:

the following rule can be extracted from the data set shown in the following Table:

Transaction ID	Milk	Bread	Butter	Cheese
1	✓	✓	✓	✗
2	✓	✓	✗	✓
3	✓	✓	✓	✗
4	✗	✓	✓	✓

{Bread} → {Butter}

it means that customers who buy bread are likely to also buy butter.

{Milk, Bread} → {Butter}

It means that if a customer buys **milk and bread**, they are likely to also buy **butter.**

Association Rule cont.

The strength of an association rule can be measured in terms of its **support** and **confidence** as the following:

Support (s)

Fraction of transactions that contain both X and Y

$s(X \rightarrow Y)=\sigma(X \cup Y)/T$

Confidence (c)

Measures how often items in Y appear in transactions that contain X

, $c(X \rightarrow Y)=\sigma(X \cup Y)/\sigma(X)$.

Example.1:

{Bread} → {Milk}

$S(\{Bread\} \rightarrow \{Milk\}) = \sigma(Bread, Milk)/T = 3/5 = 60\%$

$C(\{Bread\} \rightarrow \{Milk\}) = \sigma(Bread, Milk)/\sigma(Bread) = 3/4 = 75\%$

TID	*Items*
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Association Rule cont.

Example. 2:

{Milk} → {Diaper}

S(**{Milk} → {Diaper}**)= σ(Milk, Diaper) / T =3/5=60%

C(**{Milk} → {Diaper}**)= σ(Milk, Diaper)/σ(Milk) =3/4=75%

TID	*Items*
1	**Bread, Milk**
2	**Bread, Diaper, Beer, Eggs**
3	**Milk, Diaper, Beer, Coke**
4	**Bread, Milk, Diaper, Beer**
5	**Bread, Milk, Diaper, Coke**

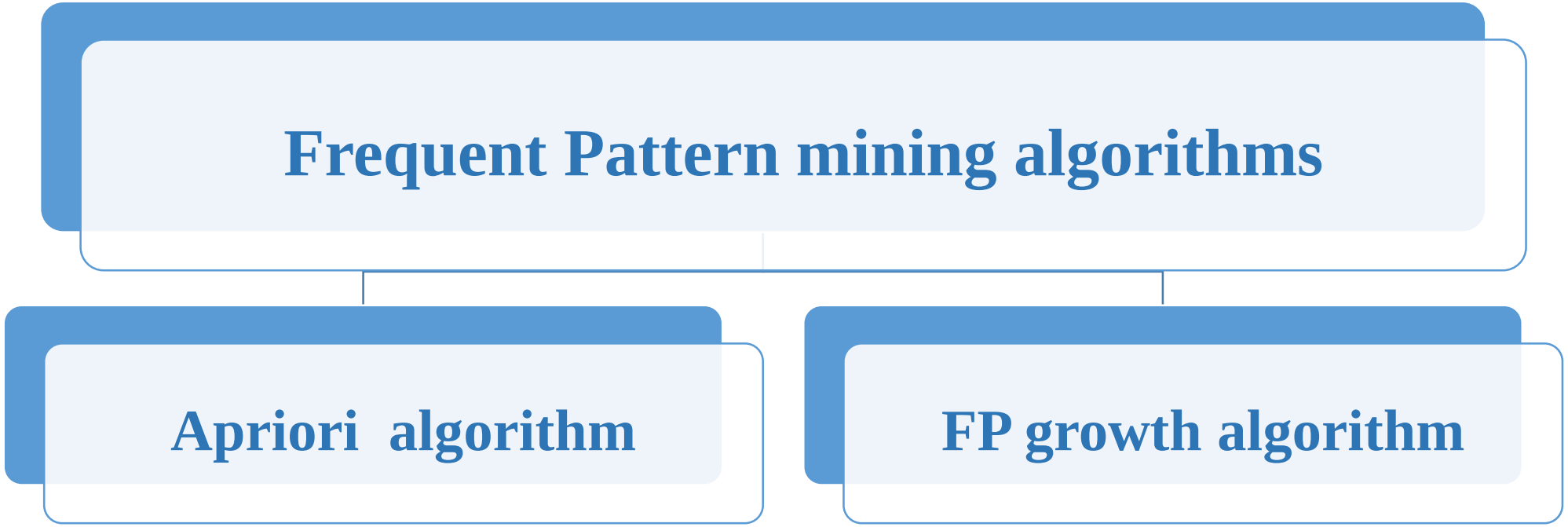
Example. 3:

{Milk, Diaper} → {Beer}

$$s = \frac{\sigma(\text{Milk}, \text{Diaper}, \text{Beer})}{|\text{T}|} = \frac{2}{5} = 0.4$$

$$c = \frac{\sigma(\text{Milk}, \text{Diaper}, \text{Beer})}{\sigma(\text{Milk}, \text{Diaper})} = \frac{2}{3} = 0.67$$

Frequent Pattern mining algorithms



Apriori algorithm

The **Apriori Algorithm** is a popular algorithm used in **association rule mining** used to detect **frequent patterns** (or frequent itemsets) in a dataset, especially in **transactional databases** like market basket analysis.

It helps in identifying relationships between items in large datasets, such as products frequently bought together in a store.

The algorithm follows the **Apriori principle**, which states:

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

This means that if a combination of items appears frequently in transactions, then individual items in that combination must also be frequent.

Apriori algorithm : Pseudo code

Algorithm *Apriori*(Transactions: $\mathcal{T}$, Minimum Support: *minsup*)
begin
  $k = 1$;
  $\mathcal{F}_1 = \{$ All Frequent 1-itemsets $\}$;
  while $\mathcal{F}_k$ is not empty **do begin**
    Generate $\mathcal{C}_{k+1}$ by joining itemset-pairs in $\mathcal{F}_k$;
    Prune itemsets from $\mathcal{C}_{k+1}$ that violate downward closure;
    Determine $\mathcal{F}_{k+1}$ by support counting on $(\mathcal{C}_{k+1}, \mathcal{T})$ and retaining
      itemsets from $\mathcal{C}_{k+1}$ with support at least *minsup*;
    $k = k + 1$;
  end;
  return$(\cup_{i=1}^{k} \mathcal{F}_i)$;
end

Apriori algorithm example. 1

Given the following transactions, and the minimum support=50%, find the frequent itemsets using Apriori algorithm

Tid	Items Bought			
1	Milk	Bread	Butter	
2	Milk	Bread		
3	Bread	Butter		
4	Milk	Bread	Butter	Eggs
5	Bread	Butter		

Apriori algorithm example.1

ID	Items Bought			
1	Milk	Bread	Butter	
2	Milk	Bread		
3	Bread	Butter		
4	Milk	Bread	Butter	Eggs
5	Bread	Butter		

the minimum support=50%

Determine the candidate item c1

C1(candidate item)

Item	count	support
Milk	3	3/5=60%
Bread	5	5/5=100%
Butter	4	4/5=80%
Eggs	1	1/5=20%

Determine F1 by Applying the minimum support on C1

F1(frequent)

Item	count	support
Milk	3	60%
Bread	5	100%
Butter	4	80%

Generate C2 from F1

C2(candidate item)

Item	count	support
Milk, Bread	3	60%
Milk, Butter	2	40%
Bread, Butter	4	80%

Determine F2 by Applying the minimum support on C2

F2(Frequent item)

Item	Count	support
Milk, Bread	3	60%
Bread, Butter	4	80%

Generate C3 from F2

C3(candidate item)

Item	count	support
Milk, Bread, Butter	2	40%

Determine F3 by Applying the minimum support on C3

F3(Frequent item)

Item	count	support
Null		

stop

Association rules.

After applying the Apriori algorithm, you can determine the association rules by following the next steps:

Step 1:

Generate Possible Association Rules, starting from **F2** or more (F3,F4,……) we will generate the association rules.

Step 2:

Compute the support and the confidence of each association rule and

Support (S)=S(X→Y)=σ(X∪Y)/T Measures how **frequent** itemset appears in all transactions.

Confidence (C)= C(X→Y)=σ(X∪Y)/σ(X) Measures how **reliable** an association rule is. checking **how often** the consequent (Y) appears when the antecedent (X) is present.

Step 3:

Determine Association Rules, by choose the association rules that satisfy the two following conditions:

Association rule support >= minimum support and Association rule confidence>= minimum confidence

important note: The minimum support for choosing frequent itemsets and the minimum support for checking association rules can be different.

Association rules.

ID	Items Bought			
1	Milk	Bread	Butter	
2	Milk	Bread		
3	Bread	Butter		
4	Milk	Bread	Butter	Eggs
5	Bread	Butter		

By applying the Apriori algorithm, with minimum support=50%

You got

F2(Frequent item)

Item	Count	support
Milk, Bread	3	60%
Bread, Butter	4	80%

With minimum support=60%, and minimum confidence= 80%, determine the association rule.

Solution:

Step 1: Generate Possible Association Rules

F2={{Milk, Bread}, {Bread, Butter} }

From frequent itemset **{Milk, Bread}**, we can get the association rules:

Milk → Bread

Bread → Milk

From frequent itemset **{Bread, Butter}**, we can get the association rules:

Butter → Bread

Bread → Butter

Association rules cont.

ID	Items Bought			
1	Milk	Bread	Butter	
2	Milk	Bread		
3	Bread	Butter		
4	Milk	Bread	Butter	Eggs
5	Bread	Butter		

Step 2: Compute the support and the confidence of each association rule

Support (S)=S(X→Y)=σ(X∪Y)/T, **Confidence (C)=C(X→Y)=σ(X∪Y)/σ(X).**

Milk → Bread

s(Milk → Bread)=σ(Milk , Bread)/T =3/5 = 60%

c(Milk → Bread)= σ(Milk , Bread) / σ(Milk) =3/3=100%

Interpretation:

"Every customer who bought **Milk** also bought **Bread**."

Bread→ Milk

s(Bread → Milk)=σ(Milk , Bread)/T =3/5 = 60%

c(Bread → Milk)= σ(Milk , Bread) / σ(Bread) =3/5=60%

Interpretation:

If a customer buys **Bread**, there is a 60% chance they will also buy **Milk**

Butter→ Bread

s(Butter → Bread)=σ(Butter , Bread)/T =4/5 = 80%

c(Butter → Bread)= σ(Butter, Bread) / σ(Butter) =4/4=100%

Interpretation:

"Every customer who bought **Butter** also bought **Bread**."

Bread→ Butter

s(Bread → Butter)=σ(Butter, Bread)/T =4/5= 80%

c(Bread → Butter)= σ(Butter , Bread) / σ(Bread) =4/5=80%

Interpretation:

If a customer buys **Bread**, there is a 80% chance they will also buy **Butter**

Association rules cont.

Step 3:

Determine Association Rules, by choose the association rules that satisfy the two following conditions:

Association rule support >= minimum support (60%)

Association rule confidence >= minimum confidence (80%)

Possible association rules

Possible association rules	support	confidence
Milk----->Bread	60%	100%
Bread----->Milk	60%	60%
Butter----->Bread	80%	100%
Bread----->Butter	80%	80%

By comparing with minimum support=60% , minimum confidence=80%

We got

Association rules

Association rules	support	confidence
Milk----->Bread	60%	100%
Butter----->Bread	80%	100%
Bread----->Butter	80%	80%

Apriori algorithm example. 2

Given the following **transactions table, and the minimum support=50%, determine the frequent itemsets using the Apriori algorithm, and hence determine the association rules with support 50% and confidence 80%.**

Tid	Items Bought			
1	A	C	D	
2	B	C	E	
3	A	B	C	E
4	B	E		

Apriori algorithm example.2 cont.

Tid	Items Bought			
1	A	C	D	
2	B	C	E	
3	A	B	C	E
4	B	E		

Determine the candidate item c1

C1(candidate item)

Item	Count	Support
A	2	2/4=50%
B	3	3/4=75%
C	3	3/4=75%
D	1	1/4=25%
E	3	3/4=75%

Determine F1 by Applying the minimum support on C1

F1(Frequent item)

Item	Count	Support
A	2	50%
B	3	75%
C	3	75%
E	3	75%

Generate C2 from F1

C2(candidate item)

Item	Count	Support
A, B	1	1/4=25%
A, C	2	2/4=50%
A, E	1	1/4=25%
B, C	2	2/4=50%
B, E	3	3/4=75%
C, E	2	2/4=50%

Determine F2 by Applying the minimum support on C2

F2(Frequent item)

Item	Count	Support
A, C	2	50%
B, C	2	50%
B, E	3	75%
C, E	2	50%

Generate C3 from F2

C3(candidate item)

Item	Count	Support
A, C, B	1	1/4=25%
A, C, E	1	1/4=25%
B, C, E	2	2/4=50%
B, E, A	1	1/4=25 %

Determine F3 by Applying the minimum support on C3

F3(Frequent item)

Item	Count	Support
B, C, E	2	2/4=50%

Generate C4 from F3

C4(candidate item)

Item	Count	Support

stop

Apriori algorithm example.2 cont.

Determine Association rules.

Tid	Items Bought			
1	A	C	D	
2	B	C	E	
3	A	B	C	E
4	B	E		

By applying the Apriori algorithm, with minimum support=50%

You got

F2(Frequent item)

Item	Count	Support
A, C	2	50%
B, C	2	50%
B, E	3	75%
C, E	2	50%

,

F3(Frequent item)

Item	Count	Support
B, C, E	2	2/4=50%

With minimum support=50%, and minimum confidence= 80%, determine the association rule.

Solution

Step 1: Generate Possible Association Rules

Generate possible association rules from F2 , F3

F2={{A, C}, {B, C} , (B, E}, {C, E} }

F3={ {B, C, E} }

Determine Association rules cont.

Step.1 cont.

Possible Association Rules from F2={{A, C}, {B, C} , (B, E}, {C, E} }

From frequent itemset **{A, C}**, we can get the association rules:

A → C
C → A

From frequent itemset **{B, C}**, we can get the association rules:

B → C
C → B

From frequent itemset **{B, E}**, we can get the association rules:

B → E
E → B

From frequent itemset **{C, E}**, we can get the association rules:

C → E

E → C

Step.1 cont.

Possible Association Rules from F3={ {B, C, E} }

From frequent itemset **{B, C, E}**, we can get the following possible association rules:

$\{B\} \rightarrow \{C, E\}$

$\{C\} \rightarrow \{B,E\}$

$\{E\} \rightarrow \{B,C\}$

$\{B, C\} \rightarrow \{E\}$

$\{B, E\} \rightarrow \{C\}$

$\{C , E\} \rightarrow \{B\}$

Step. 2.

Step 2: Compute the support and the confidence of each association rule

Tid	Items Bought			
1	A	C	D	
2	B	C	E	
3	A	B	C	E
4	B	E		

Support (s) = $S(X \rightarrow Y)=\sigma(X \cup Y)/T$, Confidence (c) = $C(X \rightarrow Y)=\sigma(X \cup Y)/\sigma(X)$.

A → C

$s(A \rightarrow C)=\sigma(A, C)/T=2/4 = 50\%$

$c(A \rightarrow C)= \sigma(A, C) / \sigma(A) =2/2=100\%$

Interpretation:
"Every customer who bought **A** also bought **C.**

C → A

$s(C \rightarrow A)=\sigma(A, C)/T =2/4 = 50\%$

$c(C \rightarrow A)= \sigma(A, C) / \sigma(C) =2/3=66\%$

Interpretation:
If a customer buys C, there is a 66% chance they will also buy A.

B → C

$s(B \rightarrow C)=\sigma(B, C)/T =2/4 = 50\%$

$c(B \rightarrow C)= \sigma(B, C) / \sigma(B) =2/3=66\%$

Interpretation:
If a customer buys B, there is a 66% chance they will also buy C

C → B

$s(C \rightarrow B)=\sigma(B, C)/T =2/4 = 50\%$

$c(C \rightarrow B)= \sigma(B, C) / \sigma(C) =2/3=66\%$

Interpretation:
If a customer buys C, there is a 66% chance they will also buy B

Step.2 cont.

Tid	Items Bought			
1	A	C	D	
2	B	C	E	
3	A	B	C	E
4	B	E		

B → E

$s(B \rightarrow E)=\sigma(B, E)/T = 3/4 = 75\%$

$c(B \rightarrow E)= \sigma(B, E) / \sigma(B) = 3/3=100\%$

Interpretation:
"Every customer who bought **B** also bought **E.**

E → B

$s(E \rightarrow B)=\sigma(B, E)/T = 3/4 = 75\%$

$c(E \rightarrow B)= \sigma(B, E) / \sigma(E) = 3/3=100\%$

Interpretation:
"Every customer who bought **E** also bought B**.**

C → E

$s(C \rightarrow E)=\sigma(C, E)/T = 2/4 = 50\%$

$c(C \rightarrow E)= \sigma(C, E) / \sigma(C) = 2/3=66\%$

Interpretation:
If a customer buys C, there is a 66% chance they will also buy E

E → C

$s(E \rightarrow C)=\sigma(C, E)/T = 2/4 = 50\%$

$c(E \rightarrow C)= \sigma(C, E) / \sigma(E) = 2/3=66\%$

Interpretation:
If a customer buys E, there is a 66% chance they will also buy C

Step.2 cont.

Tid	Items Bought			
1	A	C	D	
2	B	C	E	
3	A	B	C	E
4	B	E		

{B} → {C, E}

$s(B \rightarrow C, E) = \sigma(B, C, E)/T = 2/4 = 50\%$

$c(B \rightarrow C, E) = \sigma(B, C, E) / \sigma(B) = 2/3 = 66\%$

Interpretation:
If a customer buys B, there is a 66% chance they will also buy C and E

{C, E} → {B}

$s(C, E \rightarrow B) = \sigma(B, C, E)/T = 2/4 = 50\%$

$c(C, E \rightarrow B) = \sigma(B, C, E) / \sigma(C, E) = 2/2 = 100\%$

Interpretation:
"Every customer who bought **C and E** also bought B.

{C} → {B, E}

$s(C \rightarrow B, E) = \sigma(B, C, E)/T = 2/4 = 50\%$

$c(C \rightarrow B, E) = \sigma(B, C, E) / \sigma(C) = 2/3 = 66\%$

Interpretation:
If a customer buys C, there is a 66% chance they will also buy B and E

{B, E} → {C}

$s(B, E \rightarrow C) = \sigma(B, C, E)/T = 2/4 = 50\%$

$c(B, E \rightarrow C) = \sigma(B, C, E) / \sigma(B, E) = 2/3 = 66\%$

Interpretation:
If a customer buys B and E, there is a 66% chance they will also buy C.

Step.2 cont.

Tid	Items Bought			
1	A	C	D	
2	B	C	E	
3	A	B	C	E
4	B	E		

{E} → {B, C}

$s(E \rightarrow B, C) = \sigma(B, C, E)/T = 2/4 = 50\%$

$c(E \rightarrow B, C) = \sigma(B, C, E) / \sigma(E) = 2/3 = 66\%$

Interpretation:
If a customer buys E, there is a 66% chance they will also buy B and C

{B, C} → {E}

$s(B, C \rightarrow E) = \sigma(B, C, E)/T = 2/4 = 50\%$

$c(B, C \rightarrow E) = \sigma(B, C, E) / \sigma(B, C) = 2/2 = 100\%$

Interpretation:
"Every customer who bought **B and C** also bought E.

Step. 3

Step 3:

Determine Association Rules, by choose the association rules that satisfy the two following conditions:

Association rule support >= minimum support (50%)

Association rule confidence >= minimum confidence (80%)

Possible association rules

Possible association rules	Support	Confidence
A -----> C	50%	100%
C -----> A	50%	66%
B -----> C	50%	66%
C -----> B	50%	66%
B -----> E	75%	100%
E -----> B	75%	100%
C -----> E	50%	66%
E -----> C	50%	66%
B -----> C ,E	50%	66%
C -----> B, E	50%	66%
E -----> B, C	50%	66%
C, E -----> B	50%	100%
B, E -----> C	50%	66%
B, C -----> E	50%	100%

Comparing with minimum support=50% , minimum confidence=80%

Association rules

Association rules	Support	Confidence
A -----> C	50%	100%
B -----> E	75%	100%
E -----> B	75%	100%
C, E -----> B	50%	100%
B, C -----> E	50%	100%

Apriori algorithm drawbacks.

Computationally Expensive : Scans the dataset multiple times.

Too Many Rules : Can generate redundant or unimportant patterns.

Not suitable for high dimensional data : If there are many items, it becomes slow.

Determining the minimum Support is Difficult: choosing minimum support and confidence values affects the results, where **Too high** lead to **Important rules may be lost**, **Too low** lead to **Too many rules**, including useless ones.

Assignment #5

1- Given the following transactions, and the minimum support=40%, determine the frequent patterns (itemsets) using Apriori algorithm, and hence determine the strongest association rules with support 40% and confidence 60%.

tid	Set of items
1	$\{Bread, Butter, Milk\}$
2	$\{Eggs, Milk, Yogurt\}$
3	$\{Bread, Cheese, Eggs, Milk\}$
4	$\{Eggs, Milk, Yogurt\}$
5	$\{Cheese, Milk, Yogurt\}$

References.

[1] P. NING TAN, M. STEINBACH , A. KARPATNE , V. KUMAR, “INTRODUCTION TO DATA MINING”, (2nd Edition) (2nd. ed.), Pearson, 2018.

[2] Han, J., Pei, J., & Kamber, M, “ Data Mining: Concepts and Techniques” (3rd ed.). Morgan Kaufmann, 2011.

[3]Aguilar-Ruiz, J., Rodríguez -Baena, D., Alves, R. , “Frequent Pattern Mining.” In: Dubitzky, W., Wolkenhauer, O., Cho, KH., Yokota, H. (eds) Encyclopedia of Systems Biology. Springer, New York, 2013.

[4] C. C. Aggarwal, “Association Pattern Mining”, in: “Data Mining”, Springer, 2015.

Advanced Topics in Information systems

SE204

Lecture 6

Dr. Nelly Amer

Data mining.

- **FP Growth algorithm**
- **Python for frequent pattern mining**

FP- Growth algorithm

FP (Frequent Pattern) growth algorithm is based on Frequent Pattern (FP) Tree which is a popular method of mining association rules.

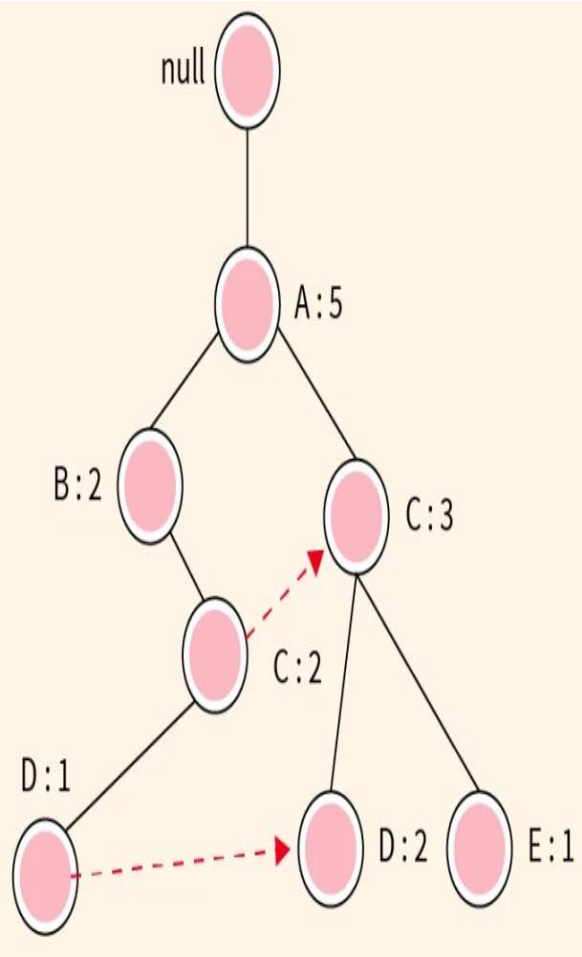
The entire transactional database is encoded in a compact prefix tree structure. Each transaction is read and represented as a path/branch of the tree.

Each node usually stores three information: name, link and count.

The parent of all nodes is a “root” node which stores a “NULL” value.

After the FP-Tree generation, Conditional Pattern (CP) Trees are generated for each item in the FP-Tree to find the frequent item sets.

FP (Frequent Pattern) growth algorithm avoided many of the shortcomings of traditional association rule mining methods such as candidate set generation and multi-database scanning.



FP Growth algorithm steps

1- Compute the frequency for each item

2- Discard the items not satisfy the minsup count

3- Arrange the items descending according to their support count

4- Arrange the items in each transactions table according to step.3

5- Construct FP Tree

6- For each item X compute the conditional pattern base(Paths from the root to occurrences of item X in the FP-tree with their counts, which are determined by the count of item X at that location.

7- For each item compute the conditional FP tree by sum up occurrences of each item in the conditional pattern base and remove **infrequent items** (below the min-support count).

8- Generate the frequent patterns

FP Growth algorithm example. 1

Given the following transactions, and the minimum support count=3, find the frequent itemsets using FP- Growth algorithm

Tid	Items Bought			
1	Milk	Bread	Butter	
2	Milk	Bread		
3	Bread	Butter		
4	Milk	Bread	Butter	Eggs
5	Bread	Butter		

FP Growth algorithm example.1 cont.

ID	Items Bought			
1	Milk	Bread	Butter	
2	Milk	Bread		
3	Bread	Butter		
4	Milk	Bread	Butter	Eggs
5	Bread	Butter		

the minsupport count=3

1- Compute the frequency for each item

Item	count
Milk	3
Bread	5
Butter	4
Eggs	1

2- Discard the items not satisfy the minsup count

3- Arrange the items descending according to their support count

Item	count
Bread	5
Butter	4
Milk	3

4- Arrange the items in each transactions table according to step.3

ID	Items Bought			
1	Milk	Bread	Butter	
2	Milk	Bread		
3	Bread	Butter		
4	Milk	Bread	Butter	Eggs
5	Bread	Butter		

Reorder the items

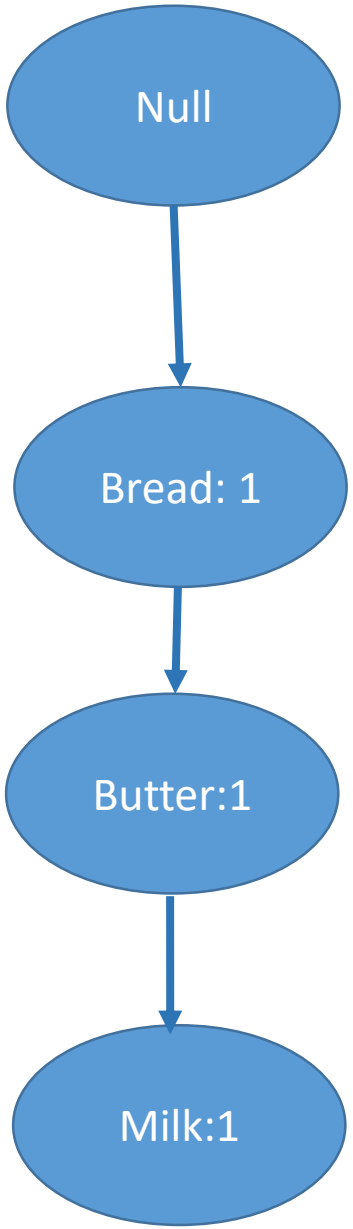
ID	Items Bought		
1	Bread	Butter	Milk
2	Bread	Milk	
3	Bread	Butter	
4	Bread	Butter	Milk
5	Bread	Butter	

FP Growth algorithm example.1 cont.

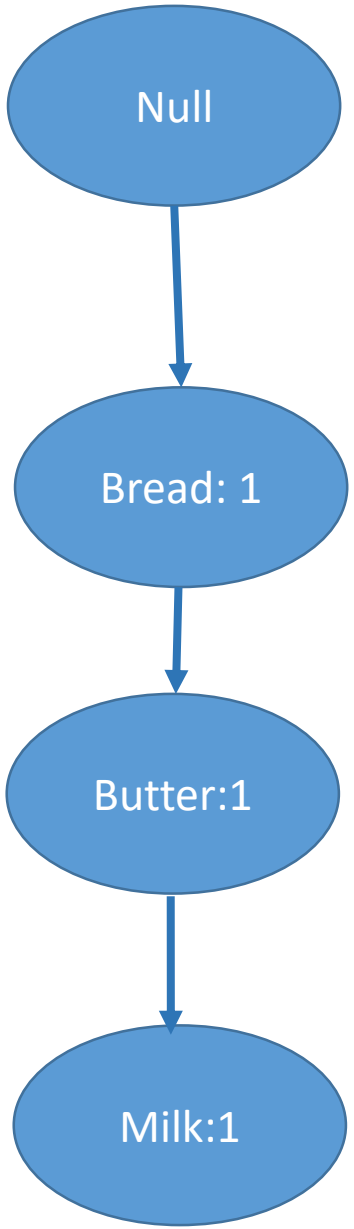
ID	Items Bought		
1	Bread	Butter	Milk
2	Bread	Milk	
3	Bread	Butter	
4	Bread	Butter	Milk
5	Bread	Butter	

The first transaction:

Bread, Butter, Milk



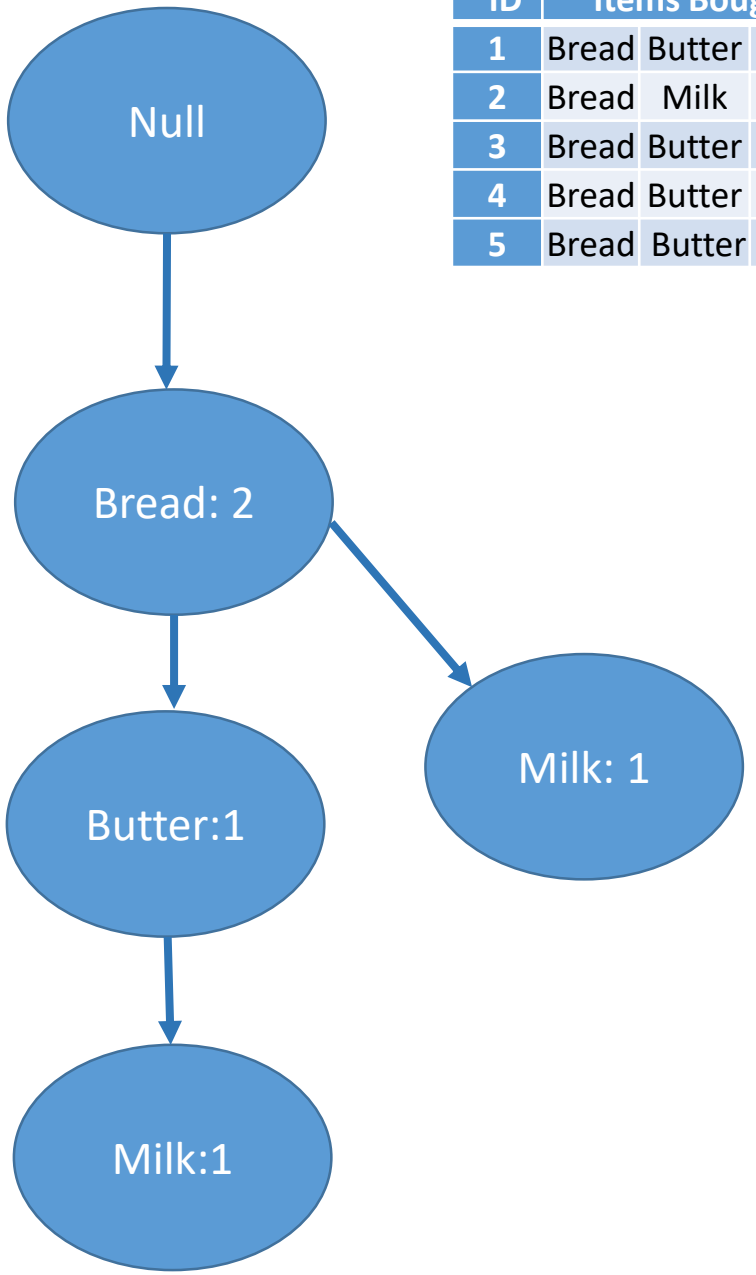
FP Growth algorithm example.1 cont.



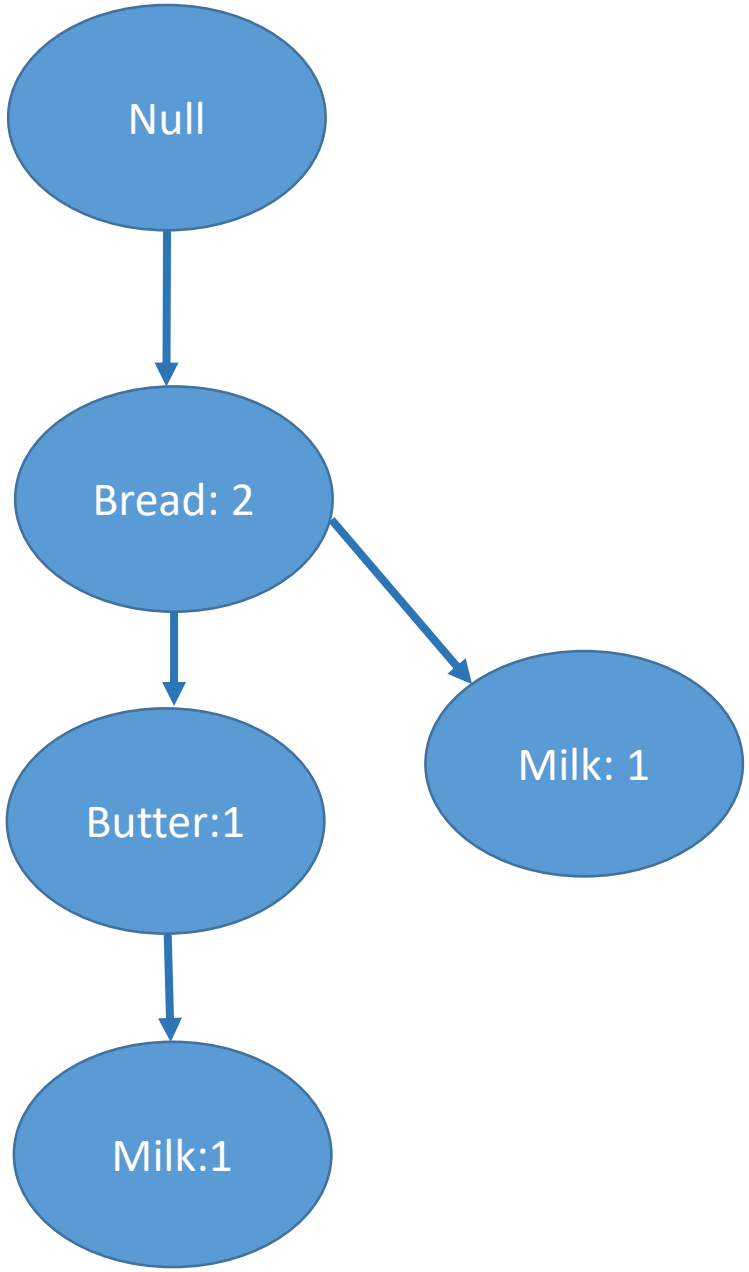
The second transaction:

Bread, Milk

ID	Items Bought		
1	Bread	Butter	Milk
2	Bread	Milk	
3	Bread	Butter	
4	Bread	Butter	Milk
5	Bread	Butter	

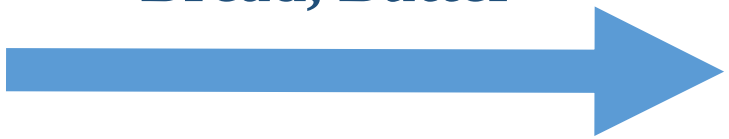


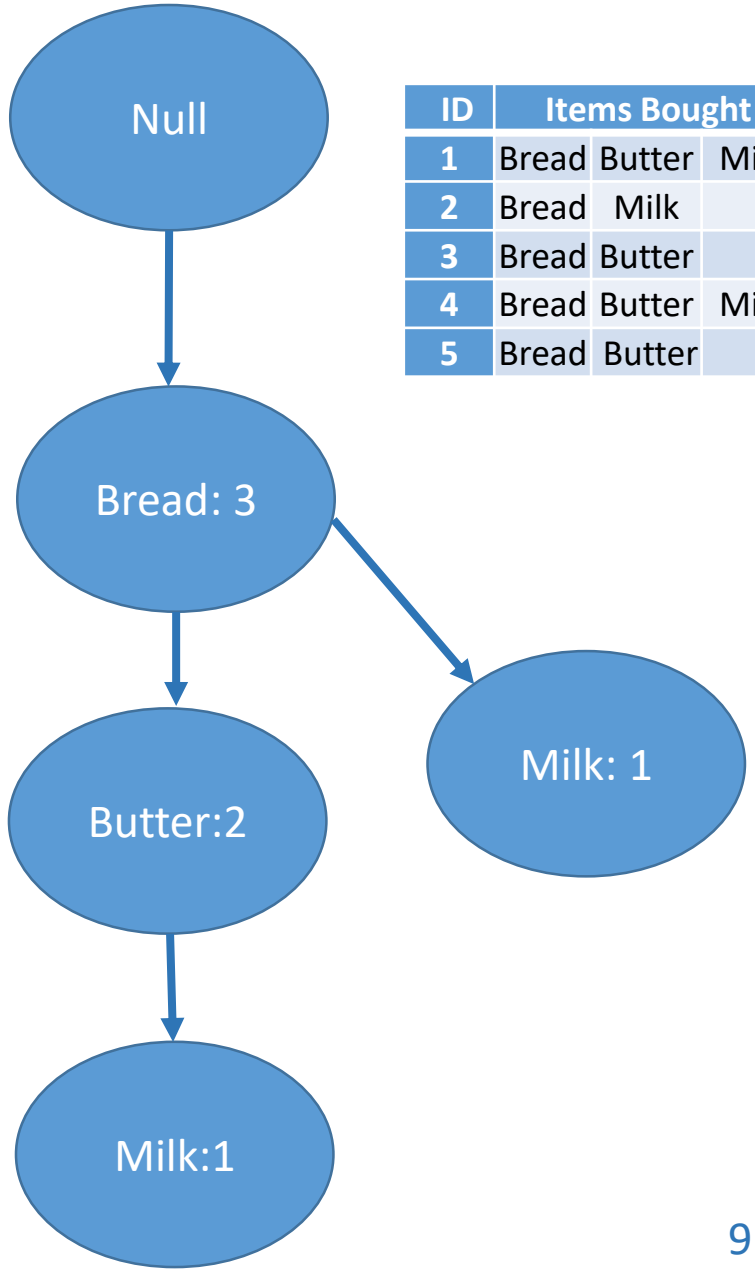
FP Growth algorithm example.1 cont.



The third transaction:

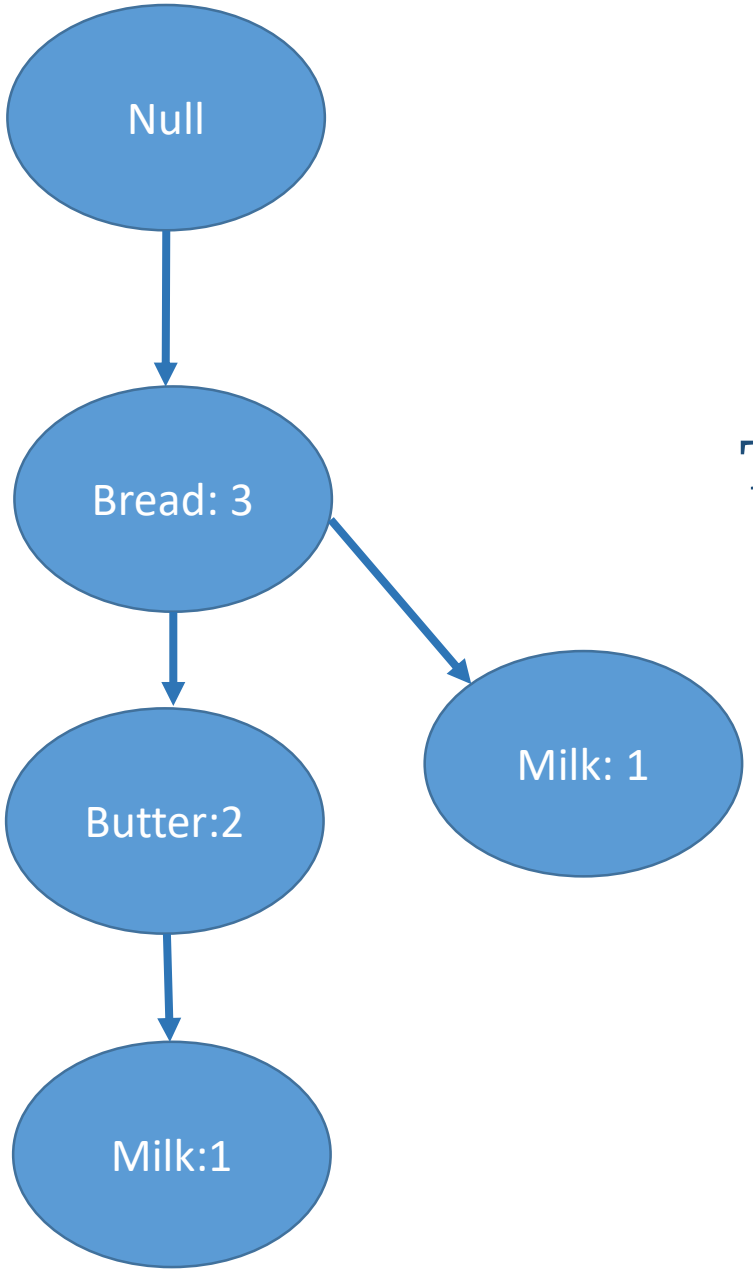
Bread, Butter





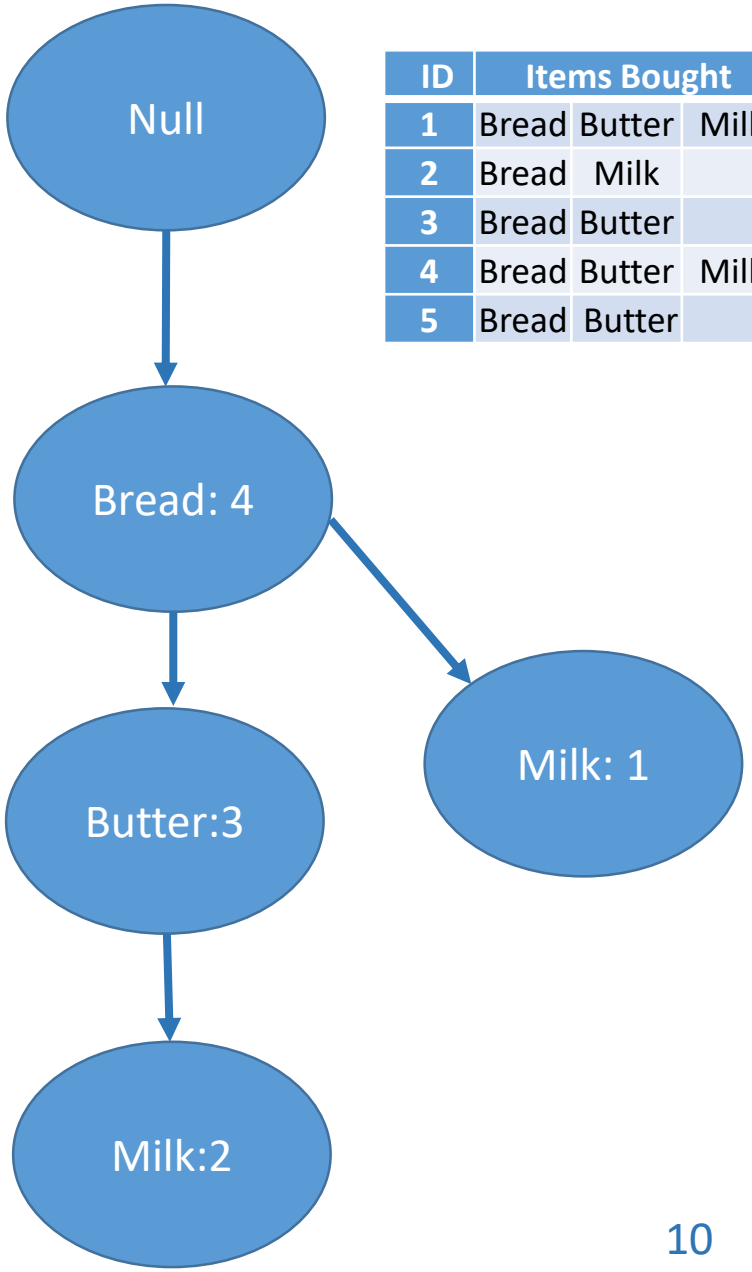
ID	Items Bought		
1	Bread	Butter	Milk
2	Bread	Milk	
3	Bread	Butter	
4	Bread	Butter	Milk
5	Bread	Butter	

FP Growth algorithm example.1 cont.



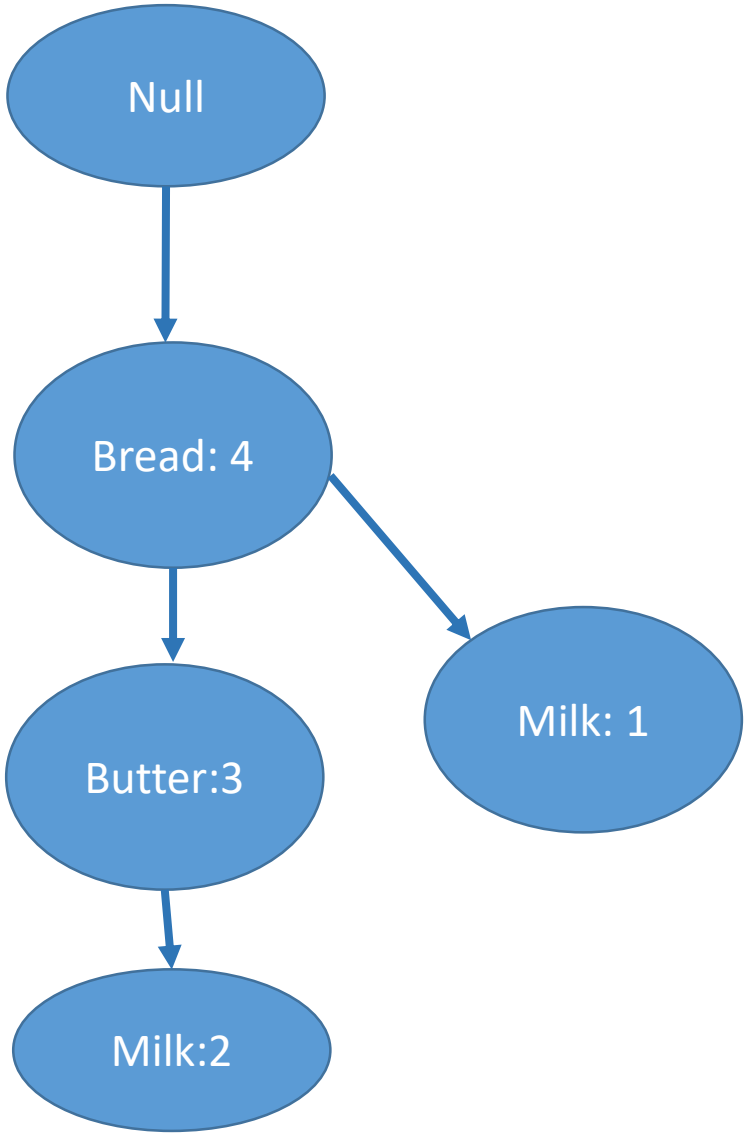
The fourth transaction:

Bread, Butter, Milk



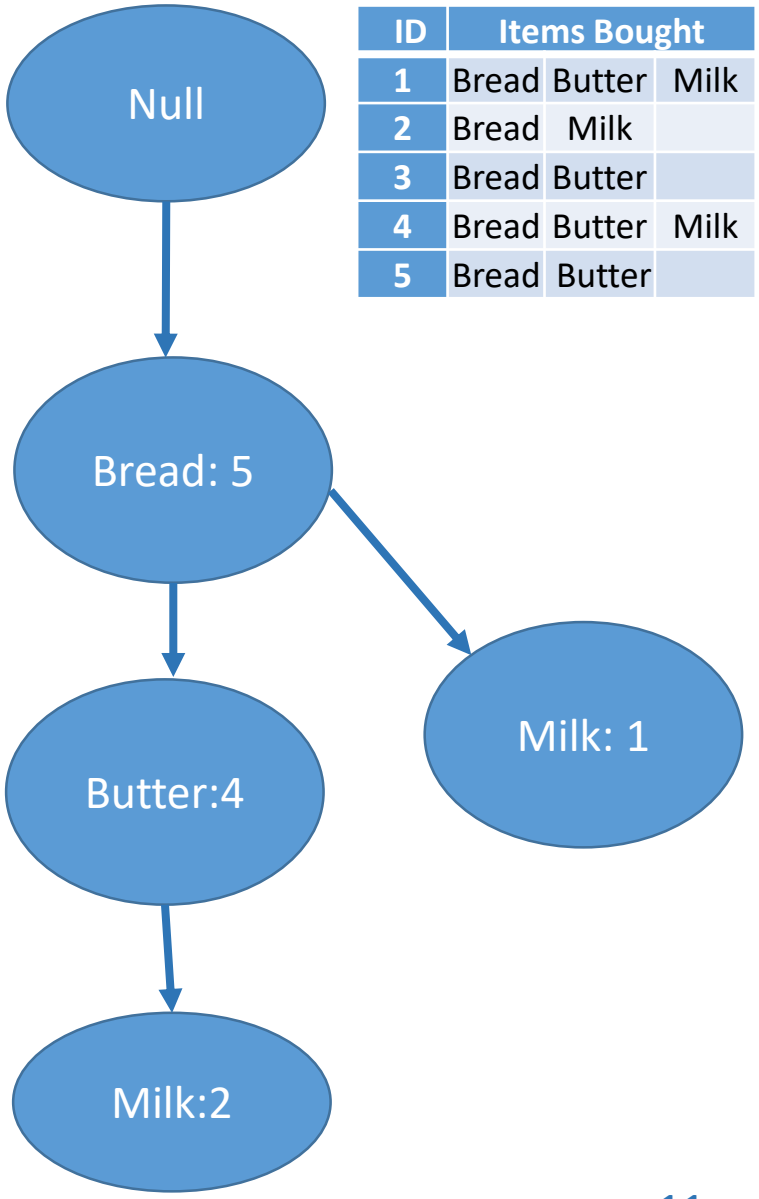
ID	Items Bought		
1	Bread	Butter	Milk
2	Bread	Milk	
3	Bread	Butter	
4	Bread	Butter	Milk
5	Bread	Butter	

FP Growth algorithm example.1 cont.



The fifth transaction:

Bread, Butter

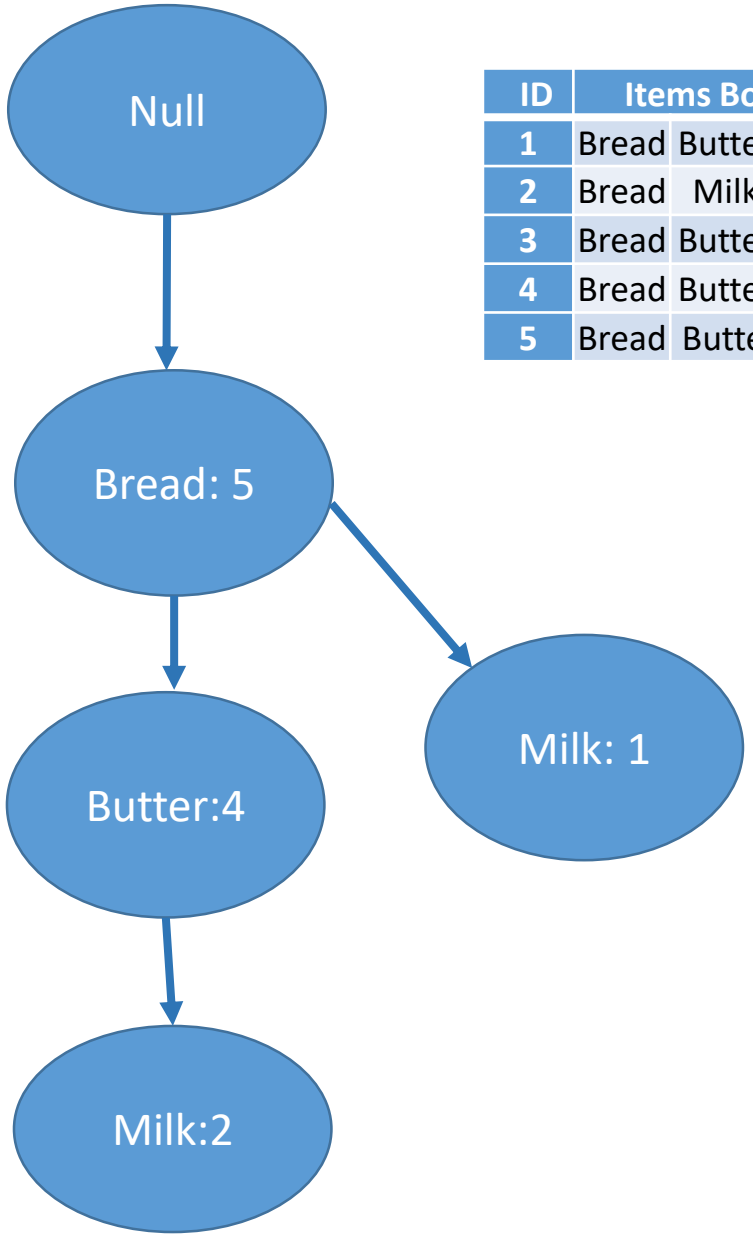


ID	Items Bought		
1	Bread	Butter	Milk
2	Bread	Milk	
3	Bread	Butter	
4	Bread	Butter	Milk
5	Bread	Butter	

Hint: look at the count for each item in the final constructed FP tree and the count support made in step 1

FP Growth algorithm example.1 cont.

Item	Conditional pattern base	Conditional F P tree	Frequent pattern generation
Milk	{{Bread:1}, {Butter, Bread:2}}	<Bread:3, Butter:2>	{Milk, Bread:3}
Butter	{Bread:4}	<Bread:4>	{Butter, Bread:4}
Bread	null		



ID	Items Bought		
1	Bread	Butter	Milk
2	Bread	Milk	
3	Bread	Butter	
4	Bread	Butter	Milk
5	Bread	Butter	

After that determine strongest association rules as in lecture #5.

FP Growth algorithm example.2

Given the following **transactions table, and the minimum support count=2, determine the frequent itemsets using the FP- Growth algorithm.**

Tid	Items Bought			
1	A	C	D	
2	B	C	E	
3	A	B	C	E
4	B	E		

FP Growth algorithm example.2 cont.

Tid	Items Bought			
1	A	C	D	
2	B	C	E	
3	A	B	C	E
4	B	E		

1- Compute the frequency for each item

Item	count
A	2
B	3
C	3
D	1
E	3

2- Discard the items not satisfy the minsup

3- Arrange the items descending according to their support

Item	count
B	3
C	3
E	3
A	2

4- Arrange the items in each transactions table according to step.3

Tid	Items Bought			
1	A	C	D	
2	B	C	E	
3	A	B	C	E
4	B	E		

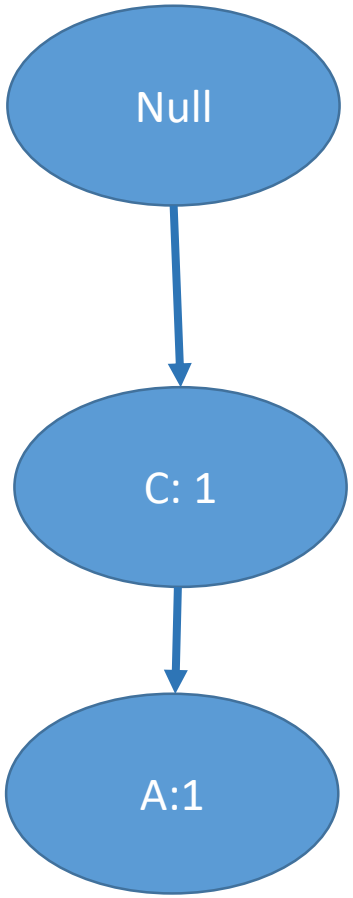
Reorder the items

Tid	Items Bought			
1	C	A		
2	B	C	E	
3	B	C	E	A
4	B	E		

FP Growth algorithm example. 2 cont.

The first transaction:

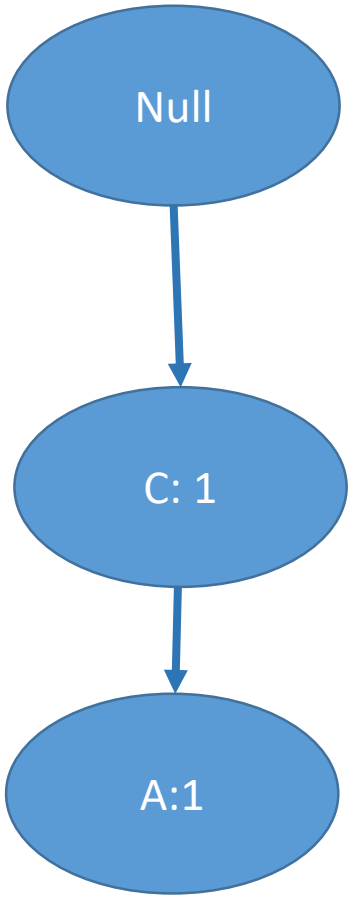
C, A



Tid	Items Bought			
1	C	A		
2	B	C	E	
3	B	C	E	A
4	B	E		

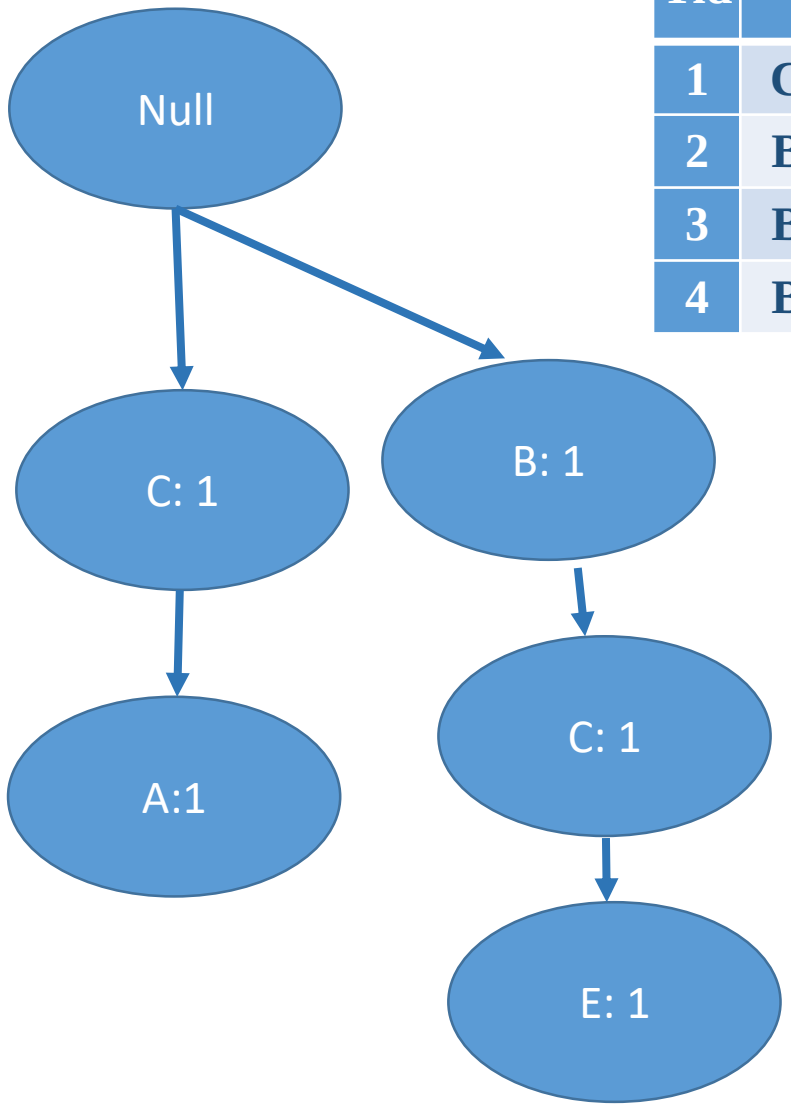
FP Growth algorithm example.2 cont.

Tid	Items Bought			
1	C	A		
2	B	C	E	
3	B	C	E	A
4	B	E		



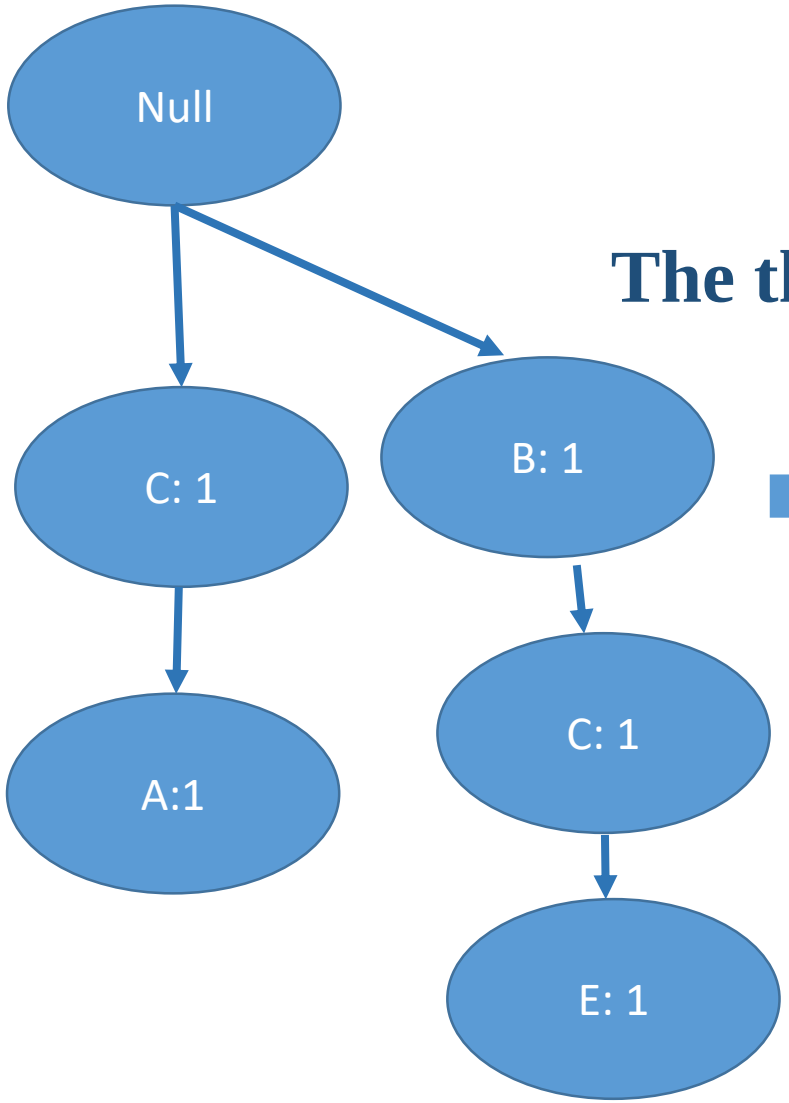
The second transaction:

B, C, E



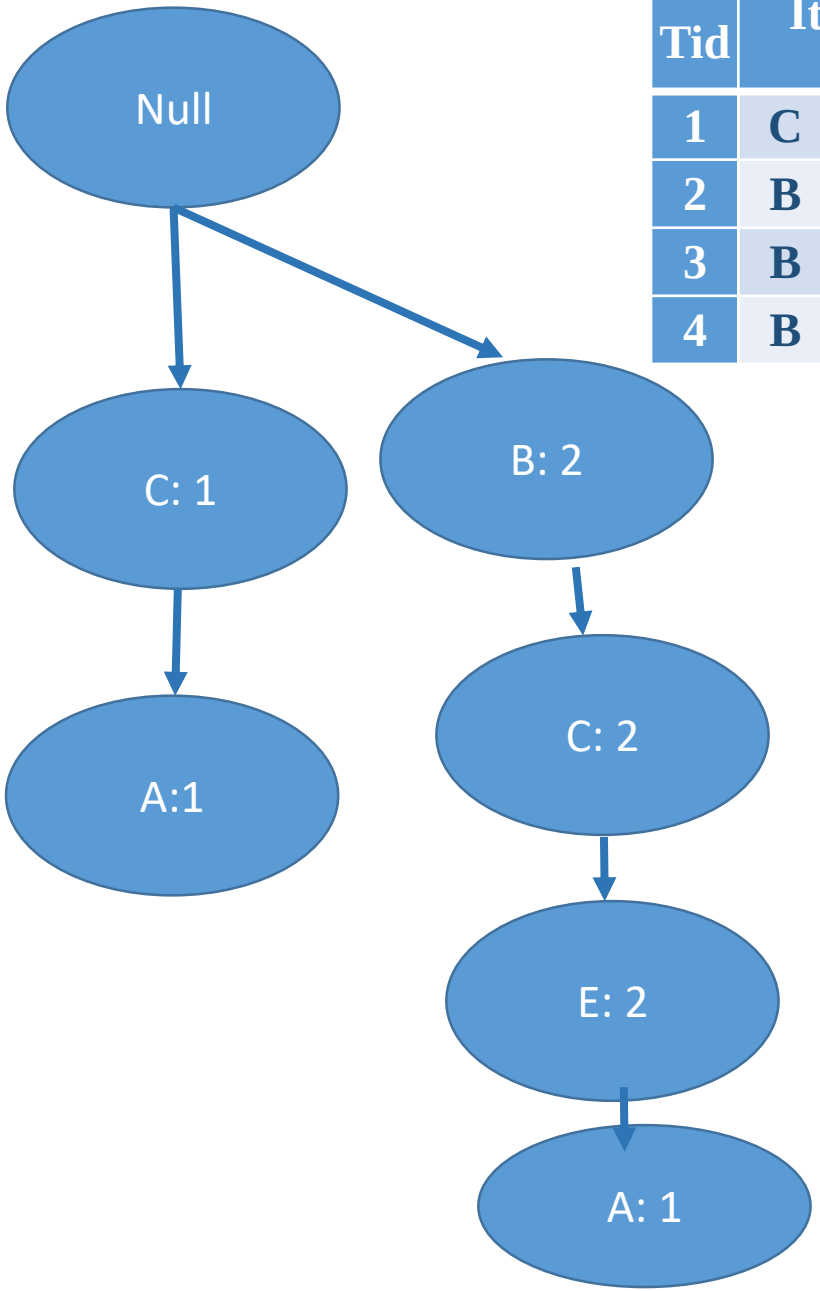
FP Growth algorithm example.2 cont.

Tid	Items Bought			
1	C	A		
2	B	C	E	
3	B	C	E	A
4	B	E		

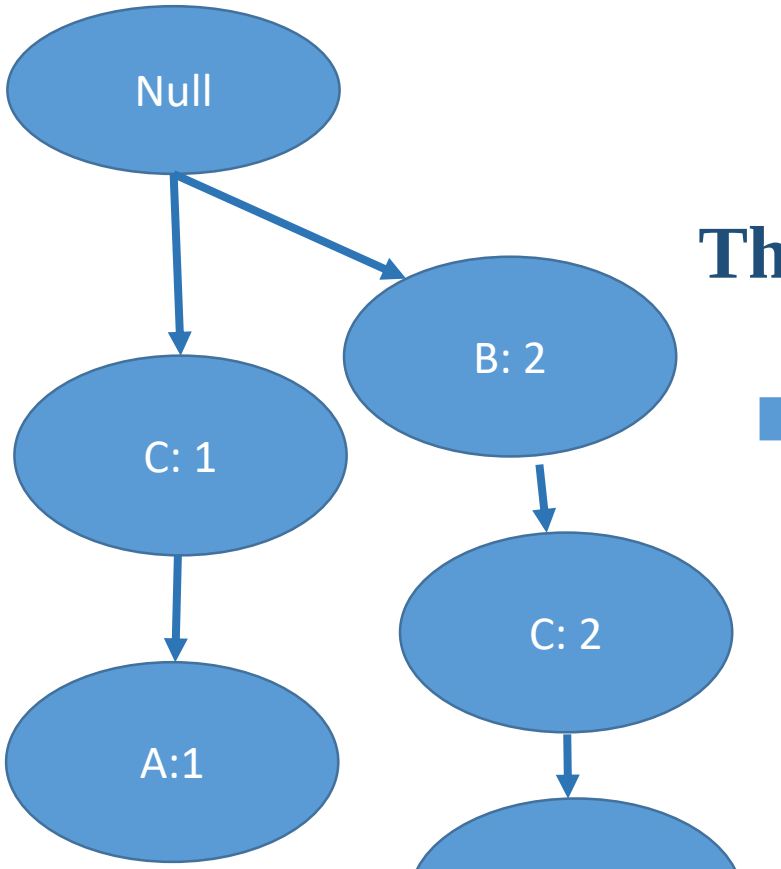


The third transaction:

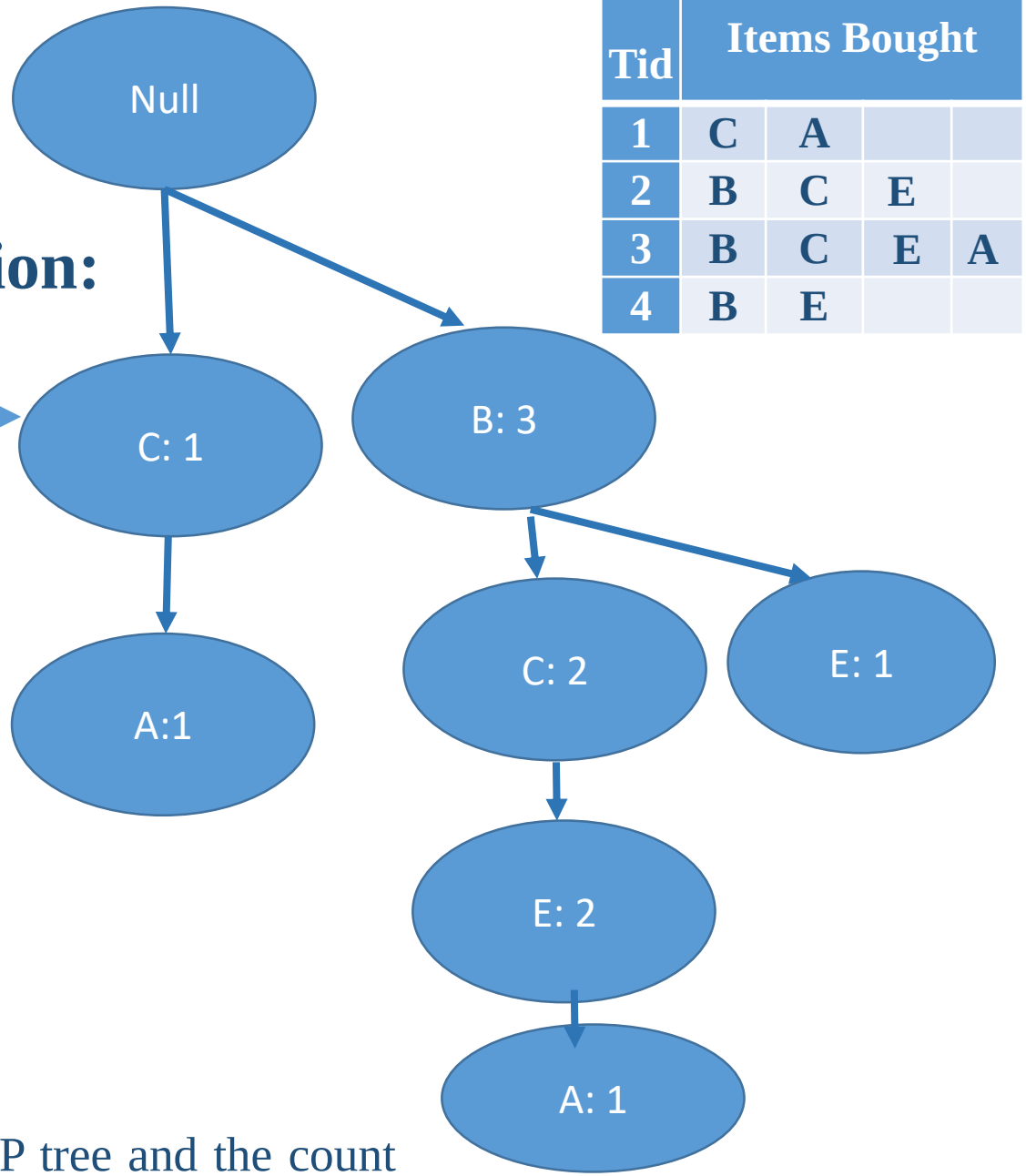
B, C, E, A



FP Growth algorithm example.2 cont.



The fourth transaction:
B, E



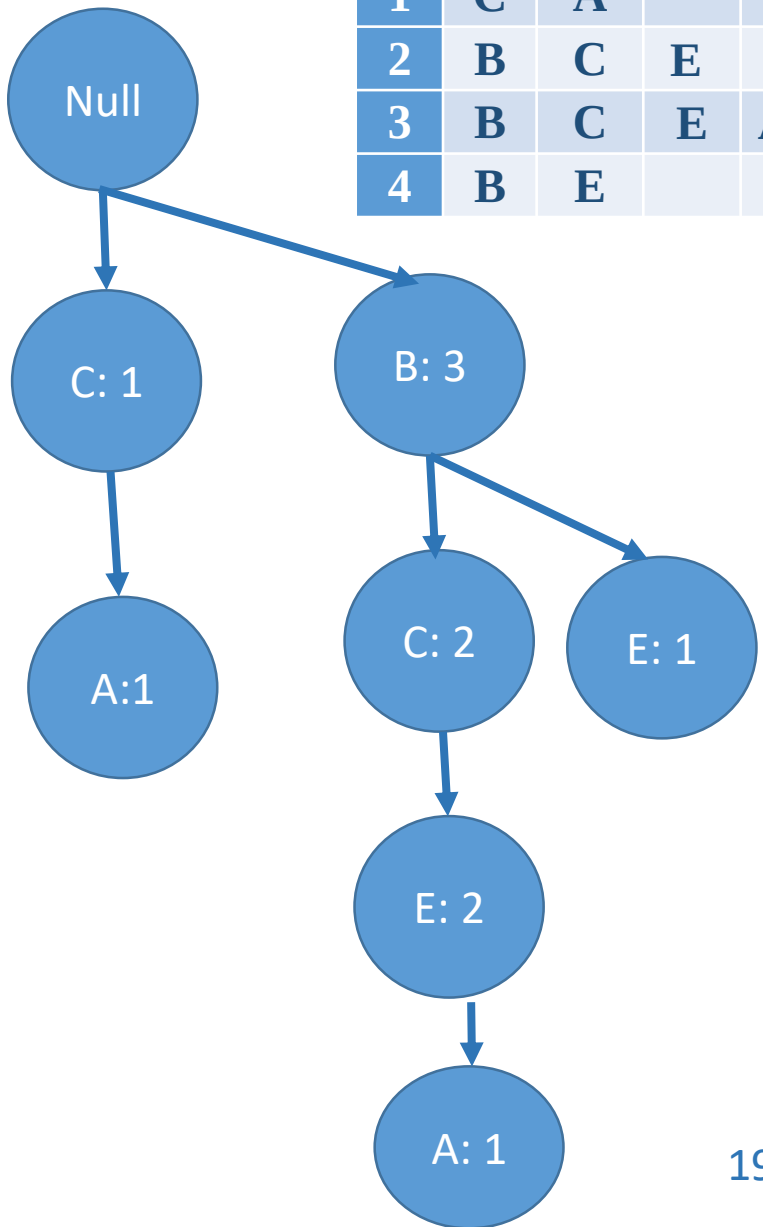
Tid	Items Bought			
1	C	A		
2	B	C	E	
3	B	C	E	A
4	B	E		

Hint: look at the count for each item in the final constructed FP tree and the count support made in step 1

FP Growth algorithm example.2 cont.

Tid	Items Bought			
1	C	A		
2	B	C	E	
3	B	C	E	A
4	B	E		

Item	Conditional pattern base	Conditional F P tree	Frequent pattern generation
A	{{B, C, E:1}, {C:1}}	<B:1>, <C:2>, <E:1>	{A, C:2}
E	{B, C:2}, {B:1}	<B:3>, <C:2>	{E, B: 3}, {E, C:2}, {E, B, C:2}
C	{B:2}	<B:2>	{C, B:2}
B	null		



After that determine strongest association rules as in lecture #5.

FP Growth algorithm example.3

Given the following **transactions table, and the minimum support count=2, determine the frequent itemsets using the FP- Growth algorithm.**

Tid	Items Bought			
1	I1	I2	I5	
2	I2	I4		
3	I2	I3		
4	I1	I2	I4	
5	I1	I3		
6	I2	I3		
7	I1	I3		
8	I1	I2	I3	I5
9	I1	I2	I3	

FP Growth algorithm example.3 cont.

Tid	Items Bought			
1	I1	I2	I5	
2	I2	I4		
3	I2	I3		
4	I1	I2	I4	
5	I1	I3		
6	I2	I3		
7	I1	I3		
8	I1	I2	I3	I5
9	I1	I2	I3	

1- Compute the frequency for each item

Item	count
I1	6
I2	7
I3	6
I4	2
I5	2

2- Discard the items not satisfy the minsup

3- Arrange the items descending according to their support

Item	count
I2	7
I1	6
I3	6
I4	2
I5	2

4- Arrange the items in each transactions table according to step.3

Tid	Items Bought			
1	I1	I2	I5	
2	I2	I4		
3	I2	I3		
4	I1	I2	I4	
5	I1	I3		
6	I2	I3		
7	I1	I3		
8	I1	I2	I3	I5
9	I1	I2	I3	

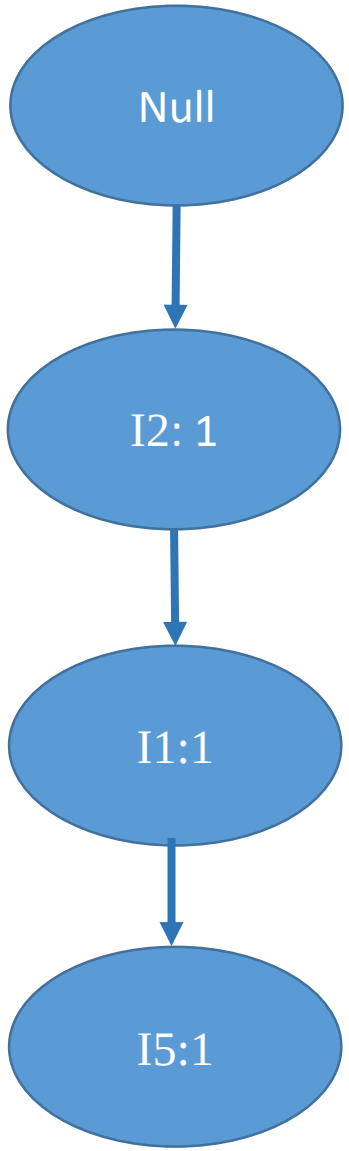
Reorder the items

Tid	Items Bought			
1	I2	I1	I5	
2	I2	I4		
3	I2	I3		
4	I2	I1	I4	
5	I1	I3		
6	I2	I3		
7	I1	I3		
8	I2	I1	I3	I5
9	I2	I1	I3	

FP Growth algorithm example. 3 cont.

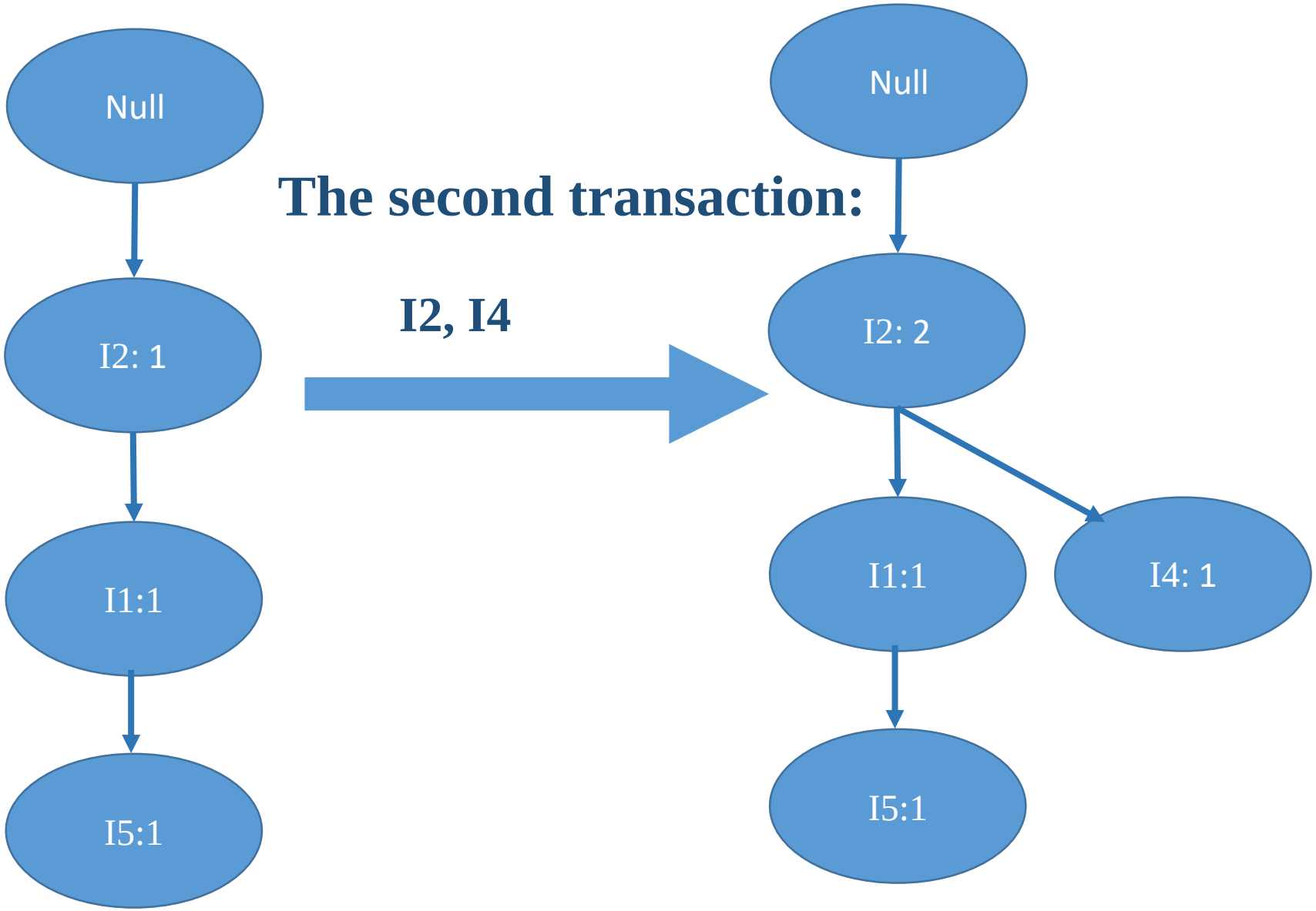
The first transaction:

I2, I1, I5



Tid	Items Bought			
1	I2	I1	I5	
2	I2	I4		
3	I2	I3		
4	I2	I1	I4	
5	I1	I3		
6	I2	I3		
7	I1	I3		
8	I2	I1	I3	I5
9	I2	I1	I3	

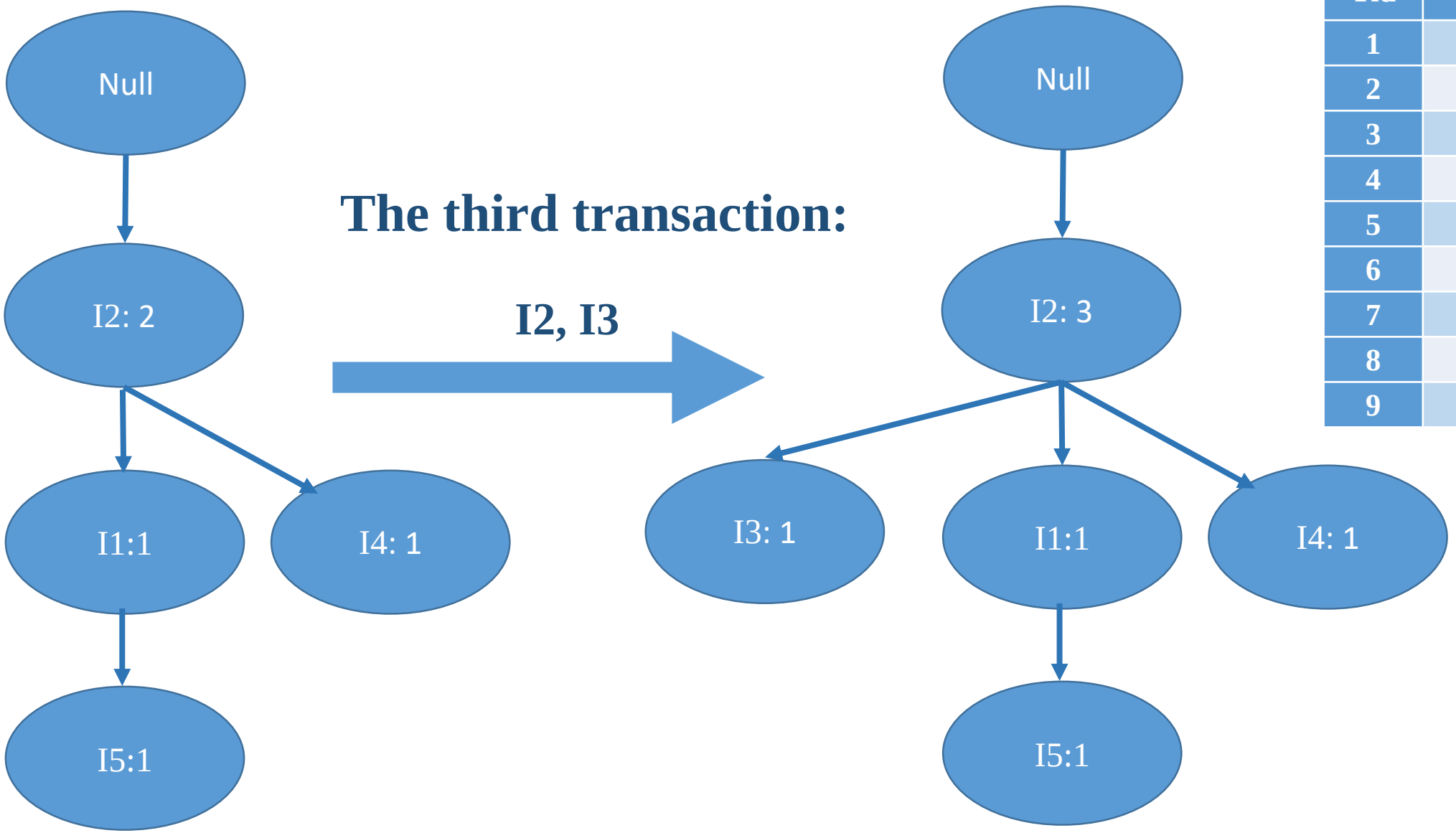
FP Growth algorithm example. 3 cont.



Tid	Items Bought			
1	I2	I1	I5	
2	I2	I4		
3	I2	I3		
4	I2	I1	I4	
5	I1	I3		
6	I2	I3		
7	I1	I3		
8	I2	I1	I3	I5
9	I2	I1	I3	

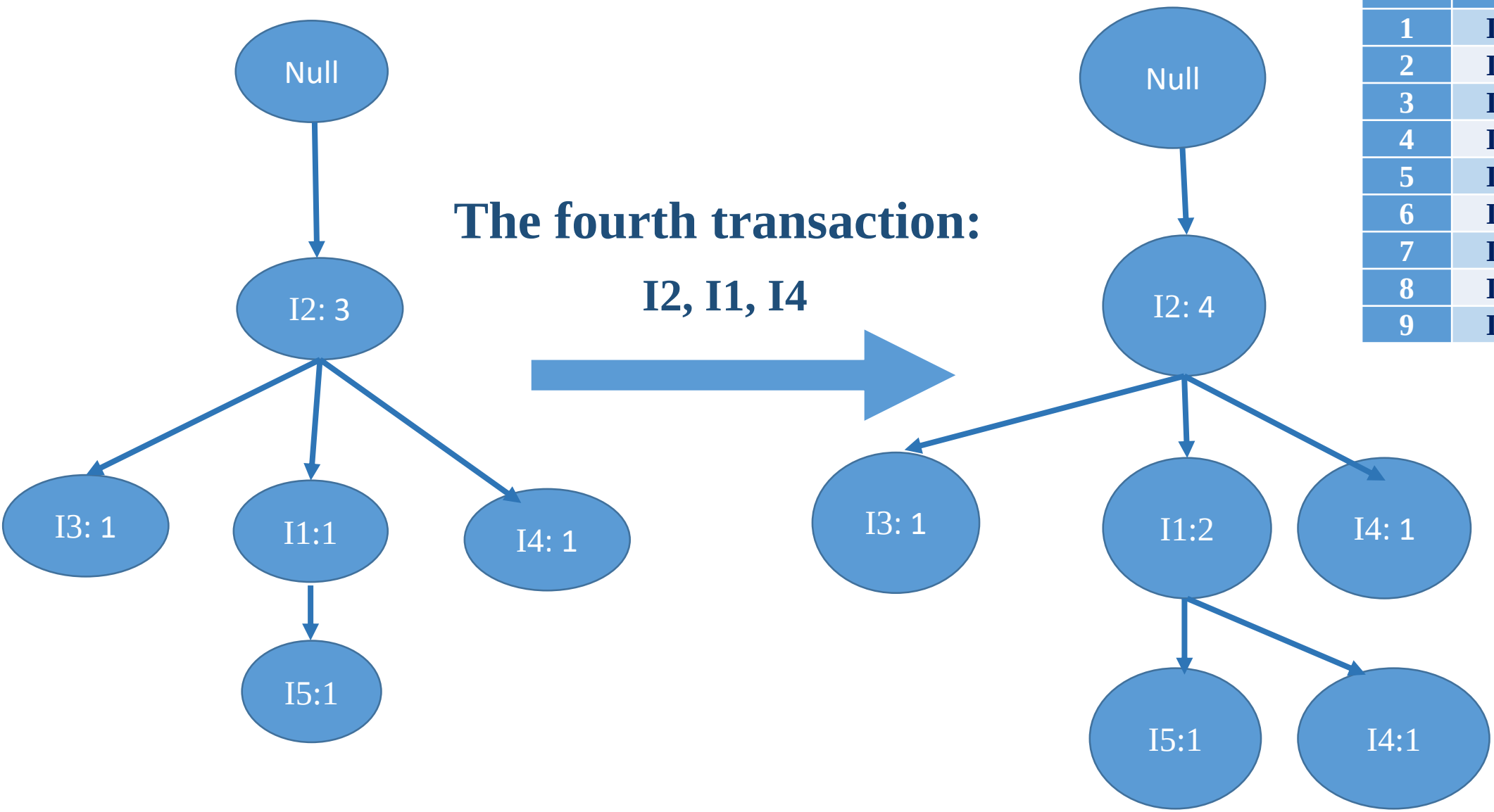
FP Growth algorithm example. 3 cont.

Tid	Items Bought			
1	I2	I1	I5	
2	I2	I4		
3	I2	I3		
4	I2	I1	I4	
5	I1	I3		
6	I2	I3		
7	I1	I3		
8	I2	I1	I3	I5
9	I2	I1	I3	



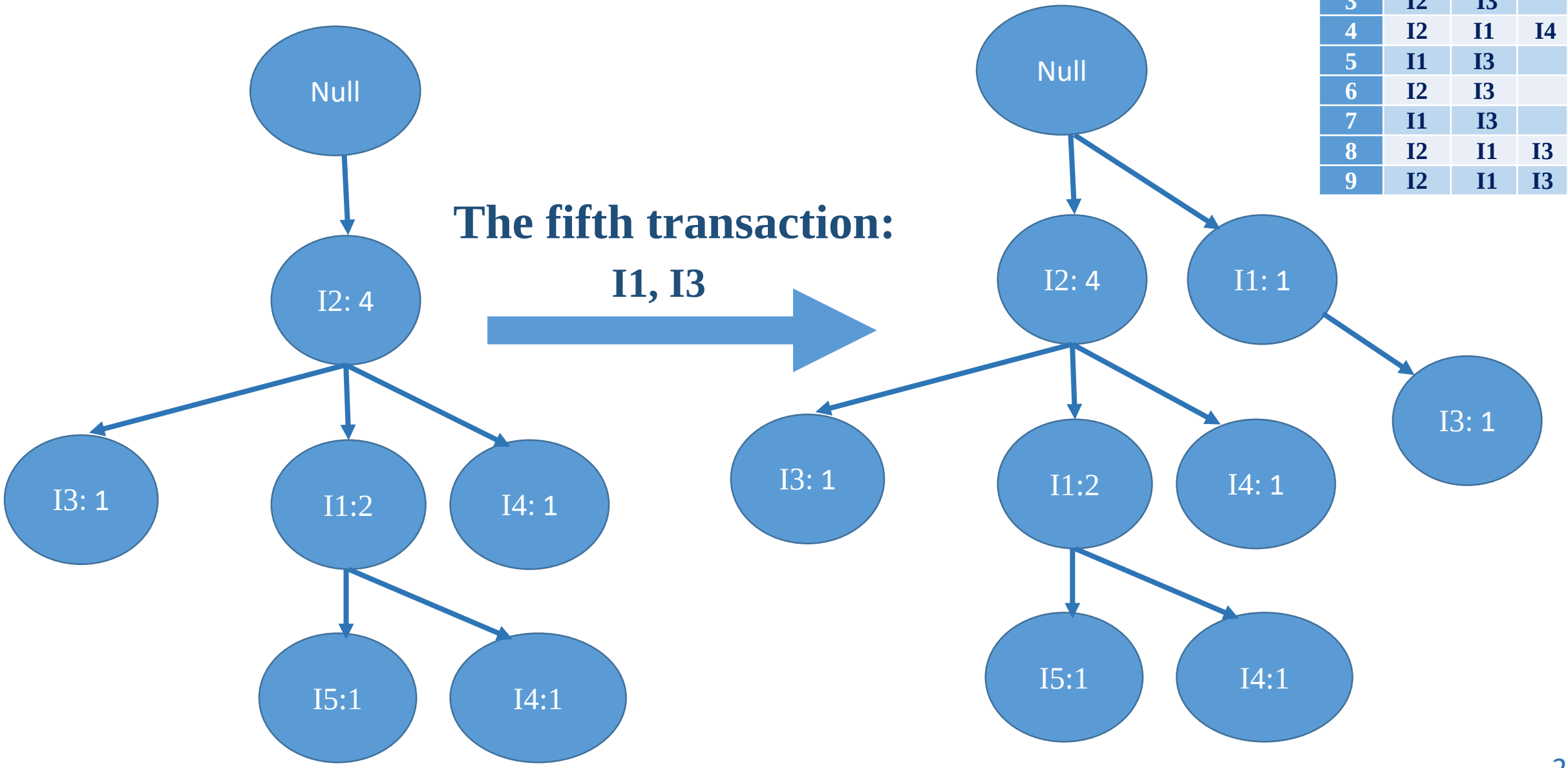
FP Growth algorithm example. 3 cont.

Tid	Items Bought			
1	I2	I1	I5	
2	I2	I4		
3	I2	I3		
4	I2	I1	I4	
5	I1	I3		
6	I2	I3		
7	I1	I3		
8	I2	I1	I3	I5
9	I2	I1	I3	



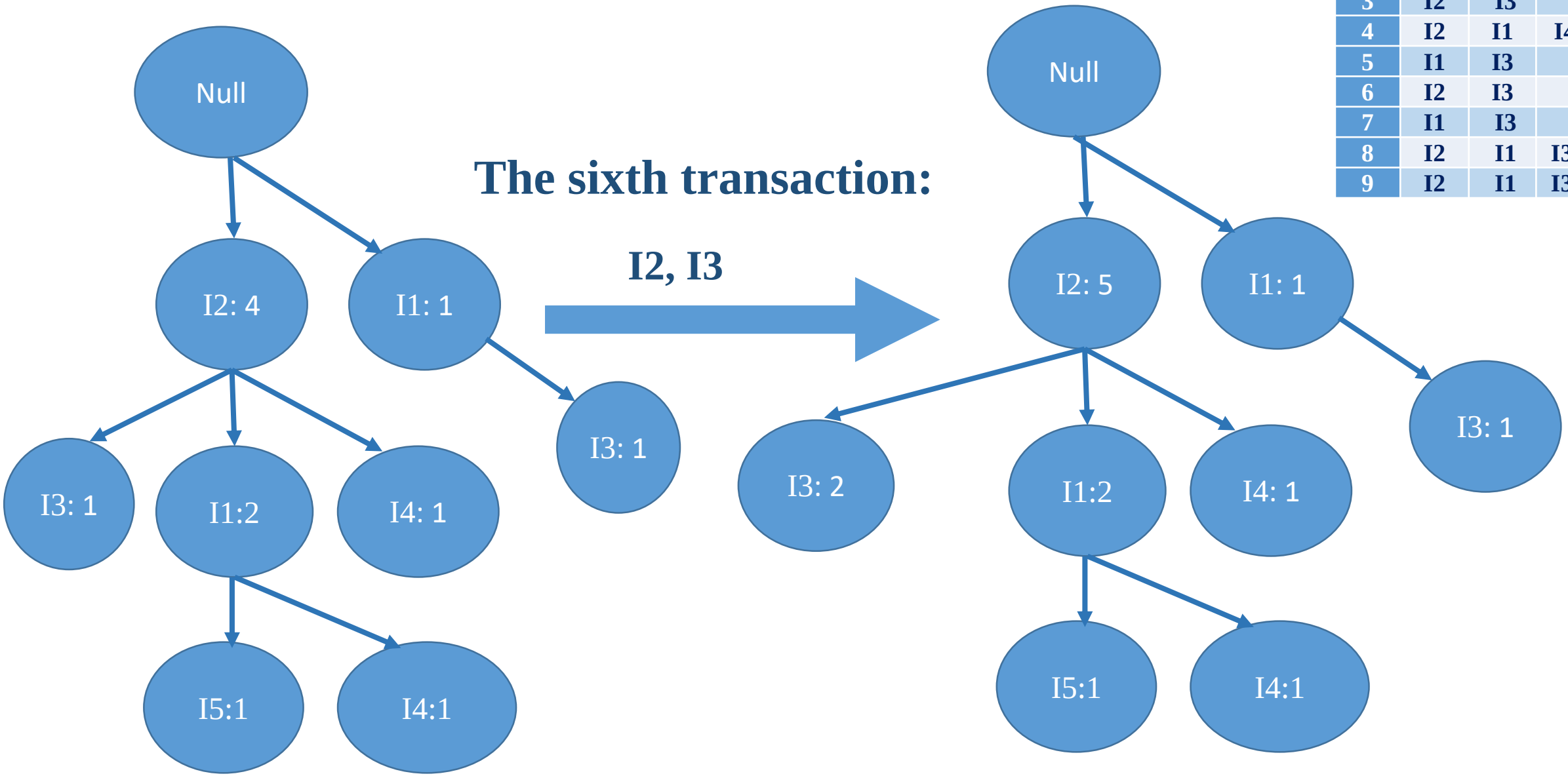
FP Growth algorithm example. 3 cont.

Tid	Items Bought			
1	I2	I1	I5	
2	I2	I4		
3	I2	I3		
4	I2	I1	I4	
5	I1	I3		
6	I2	I3		
7	I1	I3		
8	I2	I1	I3	I5
9	I2	I1	I3	



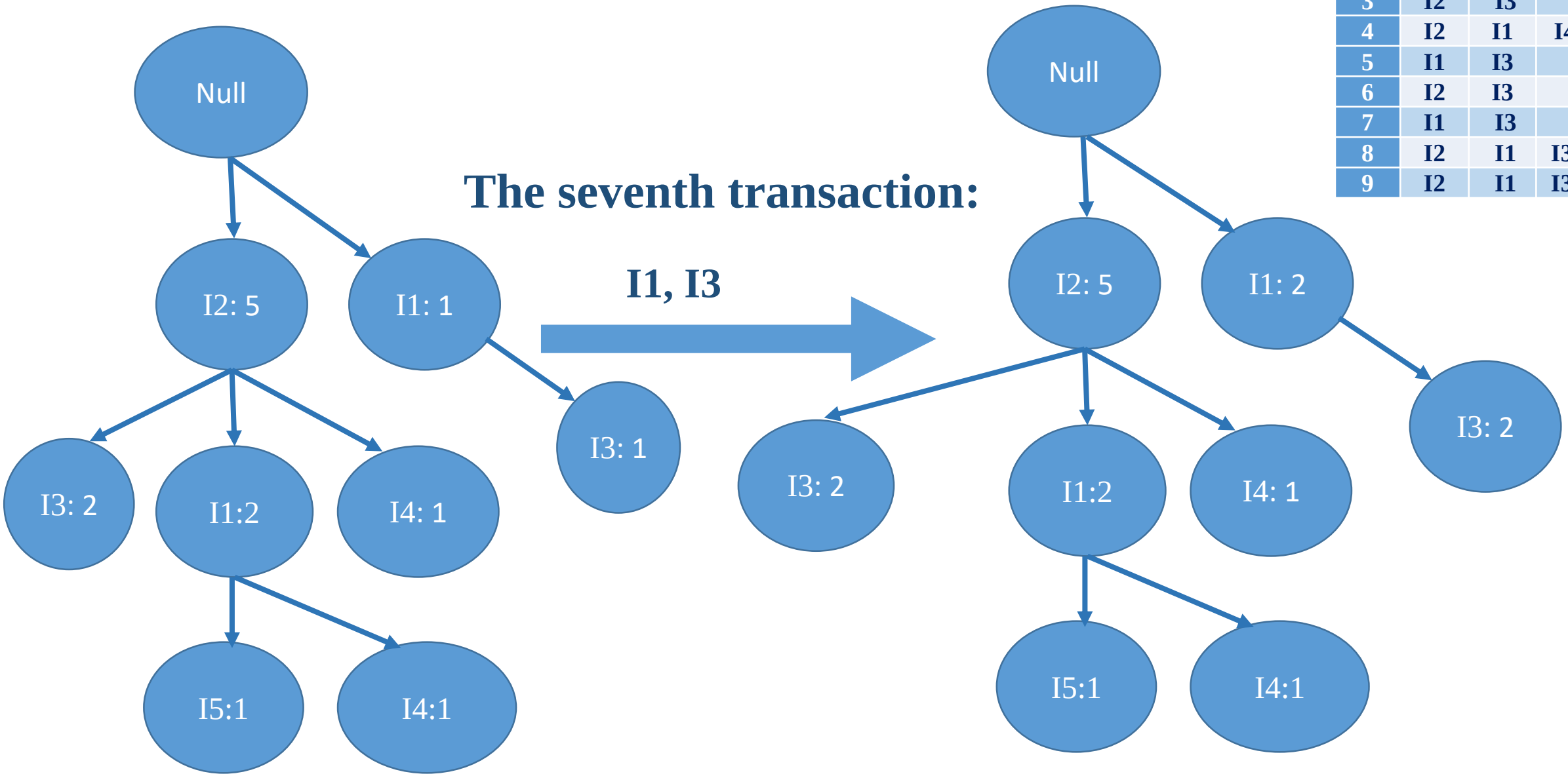
FP Growth algorithm example. 3 cont.

Tid	Items Bought			
1	I2	I1	I5	
2	I2	I4		
3	I2	I3		
4	I2	I1	I4	
5	I1	I3		
6	I2	I3		
7	I1	I3		
8	I2	I1	I3	I5
9	I2	I1	I3	



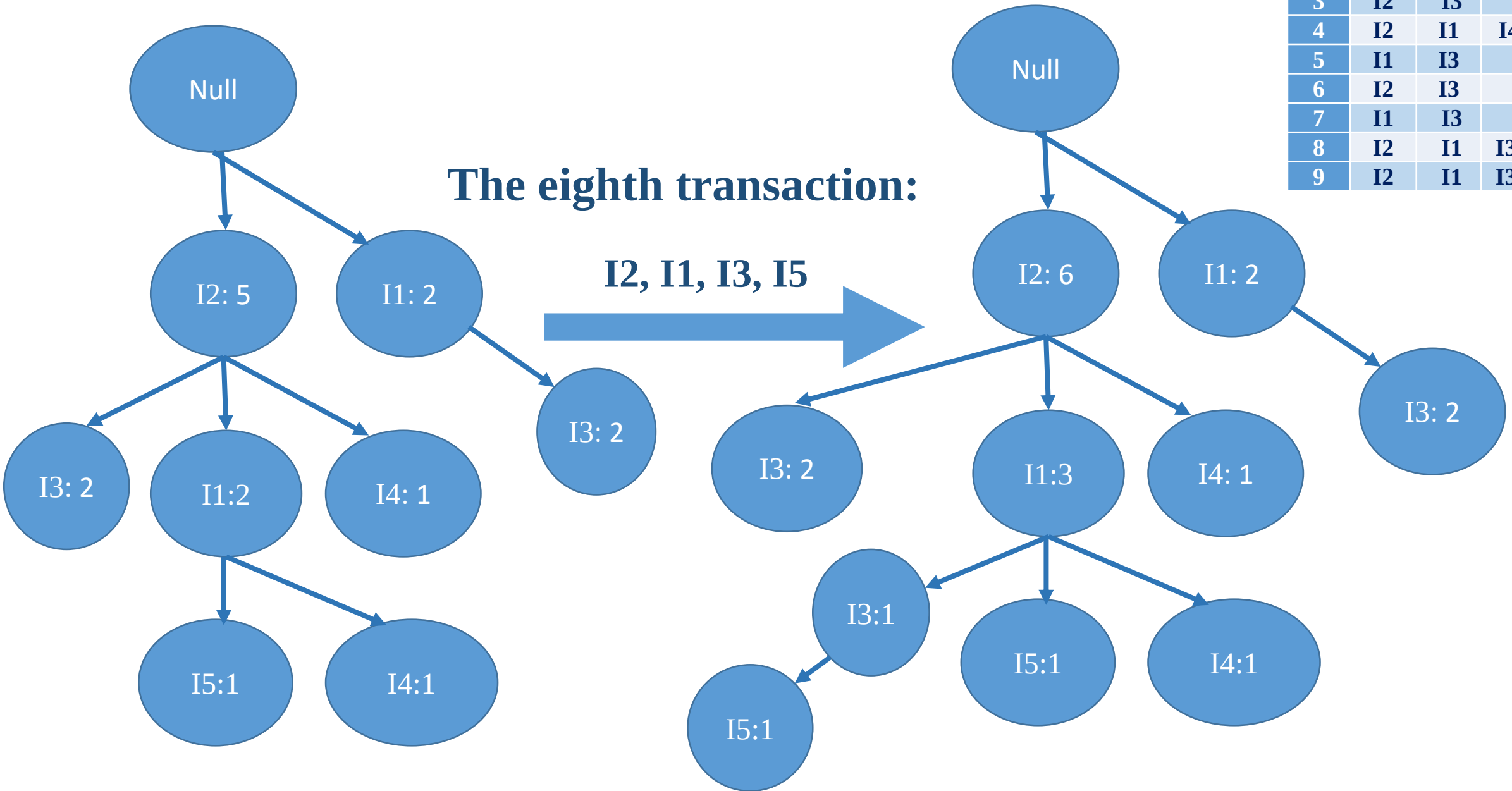
FP Growth algorithm example. 3 cont.

Tid	Items Bought			
1	I2	I1	I5	
2	I2	I4		
3	I2	I3		
4	I2	I1	I4	
5	I1	I3		
6	I2	I3		
7	I1	I3		
8	I2	I1	I3	I5
9	I2	I1	I3	



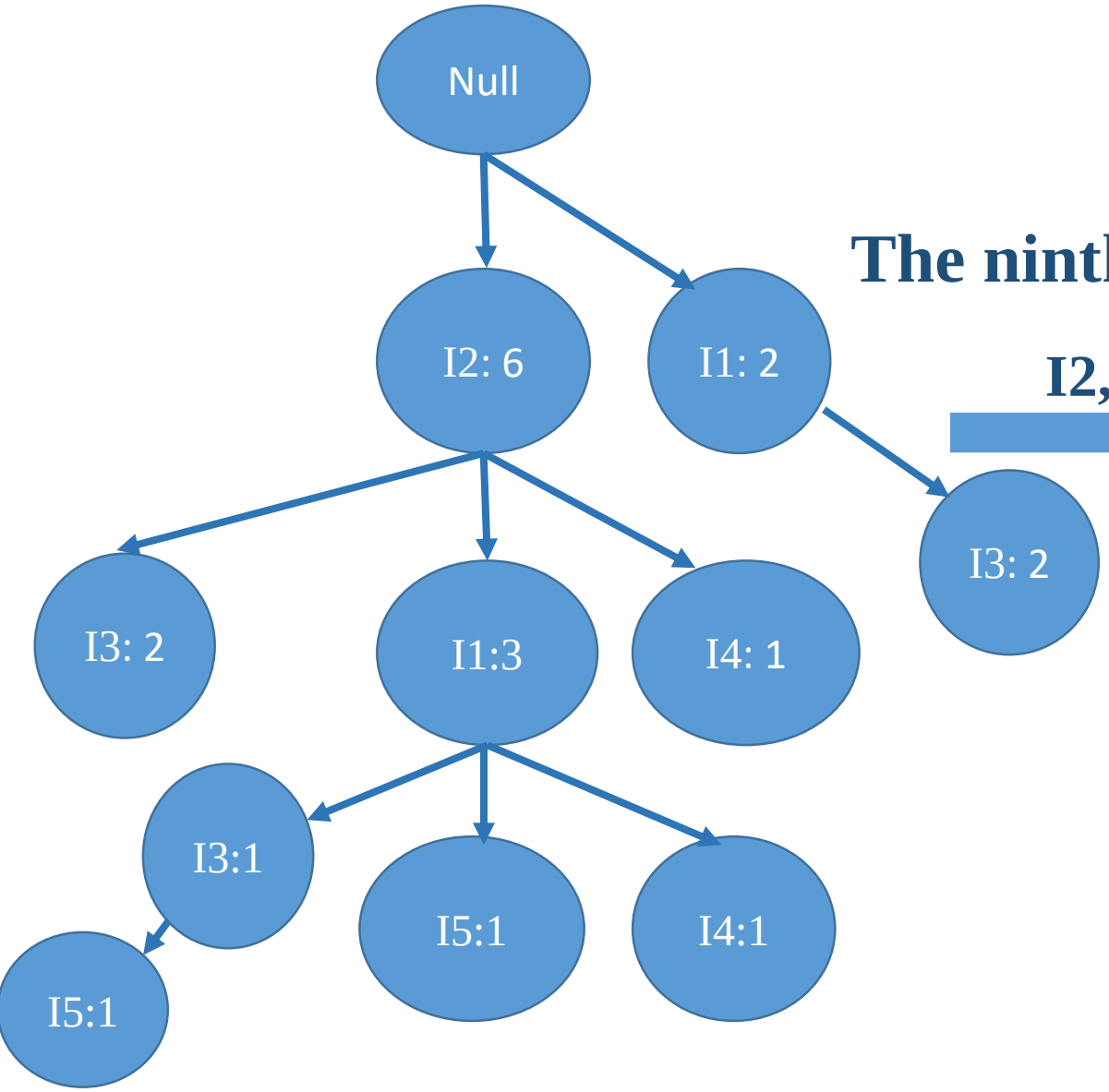
FP Growth algorithm example. 3 cont.

Tid	Items Bought			
1	I2	I1	I5	
2	I2	I4		
3	I2	I3		
4	I2	I1	I4	
5	I1	I3		
6	I2	I3		
7	I1	I3		
8	I2	I1	I3	I5
9	I2	I1	I3	



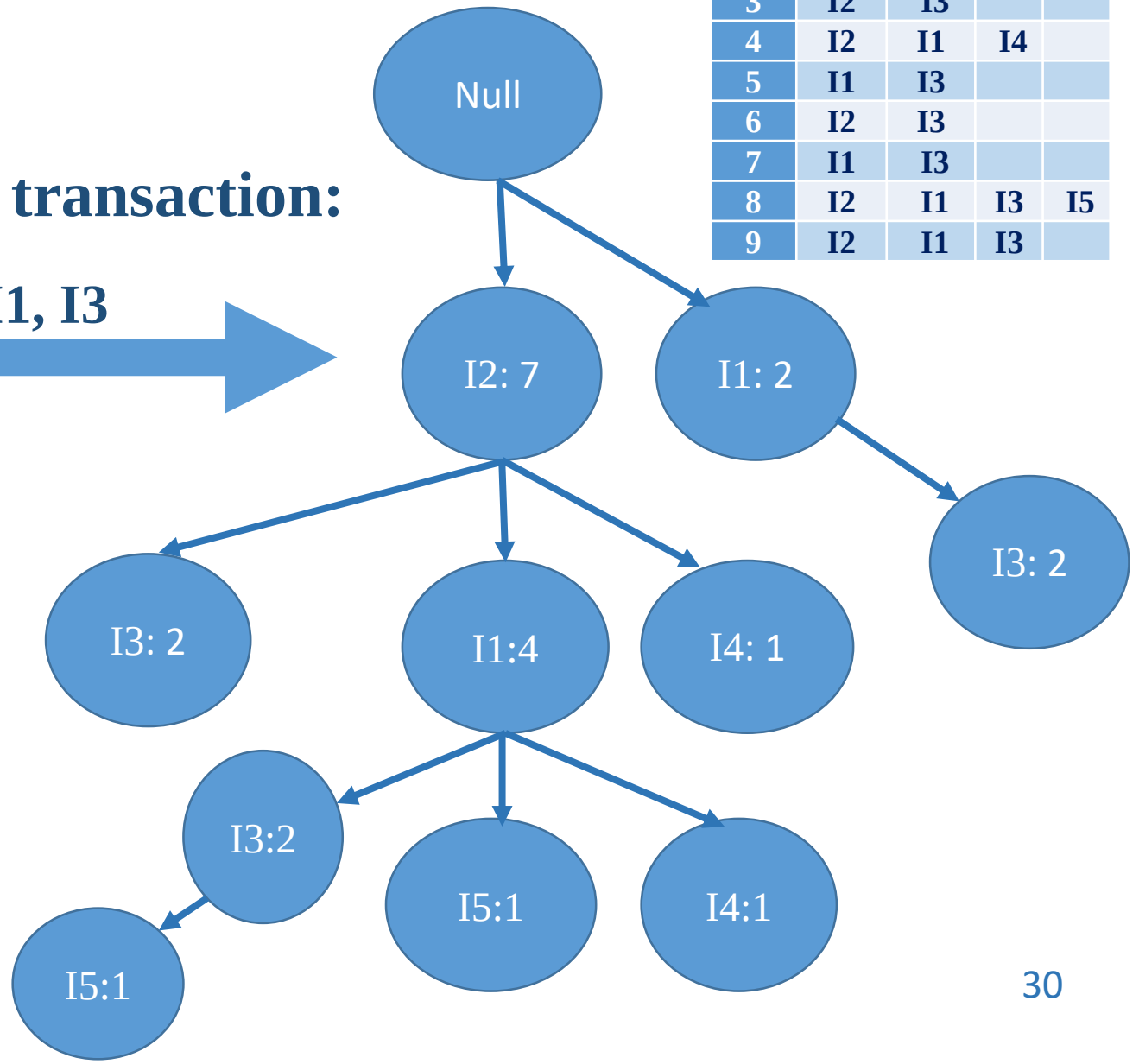
FP Growth algorithm example. 3 cont.

Tid	Items Bought			
1	I2	I1	I5	
2	I2	I4		
3	I2	I3		
4	I2	I1	I4	
5	I1	I3		
6	I2	I3		
7	I1	I3		
8	I2	I1	I3	I5
9	I2	I1	I3	



The ninth transaction:

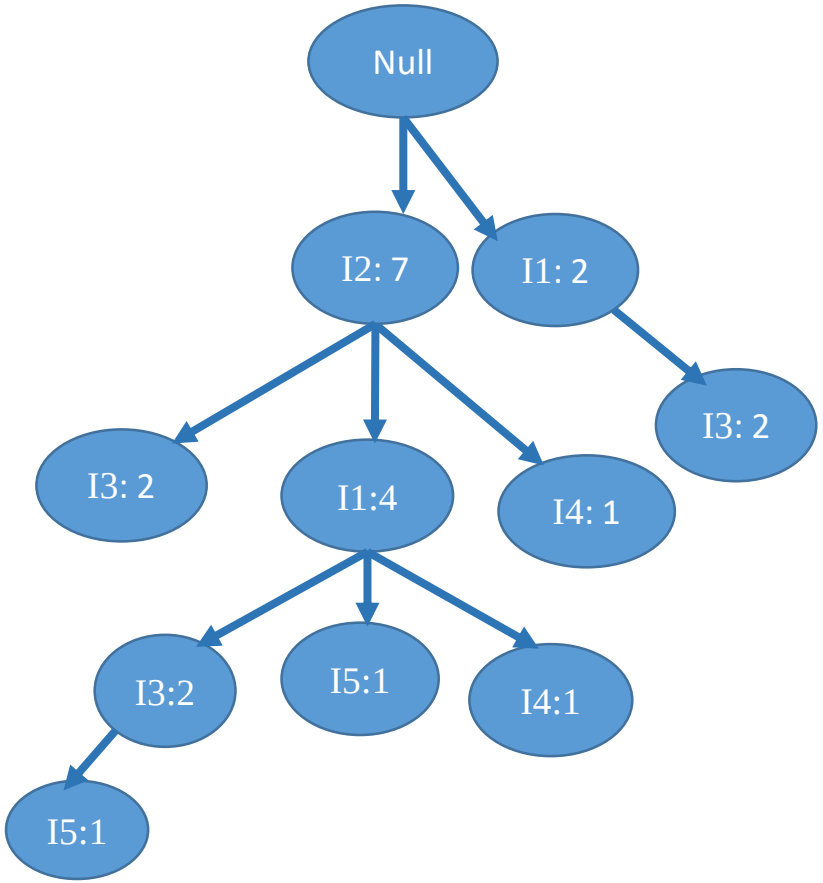
I2, I1, I3



Hint: look at the count for each item in the final constructed FP tree and the count support made in step 1

FP Growth algorithm example. 3 cont.

Tid	Items Bought			
1	I2	I1	I5	
2	I2	I4		
3	I2	I3		
4	I2	I1	I4	
5	I1	I3		
6	I2	I3		
7	I1	I3		
8	I2	I1	I3	I5
9	I2	I1	I3	



Item	Conditional pattern base	Conditional F P tree	Frequent pattern generation
I5	{{I2, I1, I3:1}, {I2, I1:1}}	<I1:2>, <I2:2>, <I3:1>	{I1, I5:2}, {I2, I5:2} , {I1, I2, I5:2}
I4	{I2:1}, {I2, I1:1}	<I2:2>, <I1:1>	{I2, I4: 2}
I3	{I1:2}, {I2, I1:2}, {I2:2}	<I1:4>, <I2:4>	{I1,I3 :4}, {I2,I3 :4} , {I1, I2, I3 :2}
I1	{I2:4},	<I2:4>	{I1,I2 :4}
I2	Null		

After that determine strongest association rules as in lecture #5.

FP-Growth advantages and disadvantages

Advantages of FP-Growth

FP-growth is faster than Apriori

- No candidate generation, no candidate test
- Eliminate repeated database scan
- Basic operation is counting and FP-tree building

Disadvantages of FP-Growth

- FP Tree is more cumbersome and difficult to build than Apriori.
- It may be expensive.
- When the database is large, the algorithm may not fit in the shared memory.

Python

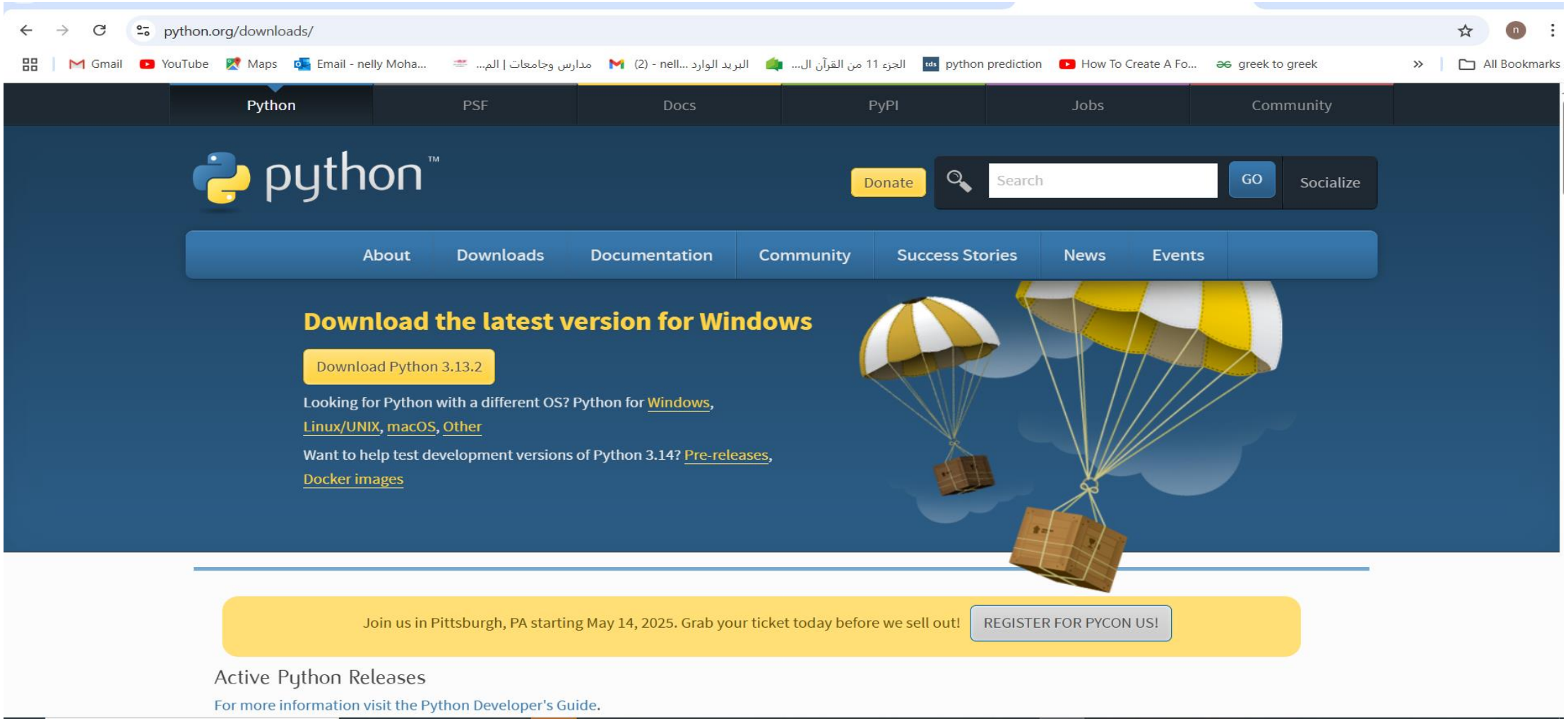
Why Python

- Python is a **multidisciplinary** language because it is used across various fields and industries. Its flexibility and extensive libraries make it suitable for a wide range of applications
- Strong Community and Libraries

Python

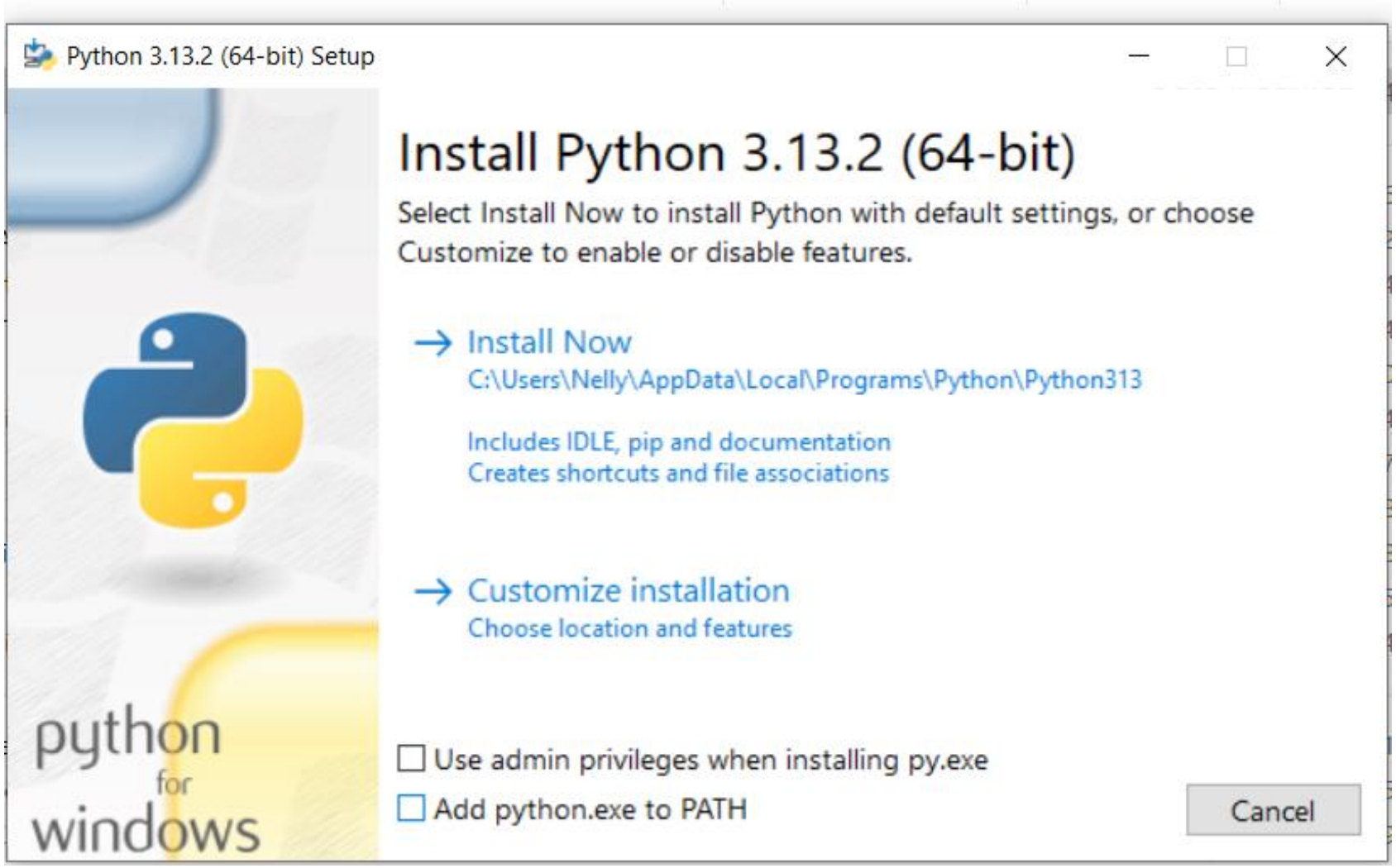
install python

https://www.python.org/downloads/



Python cont.

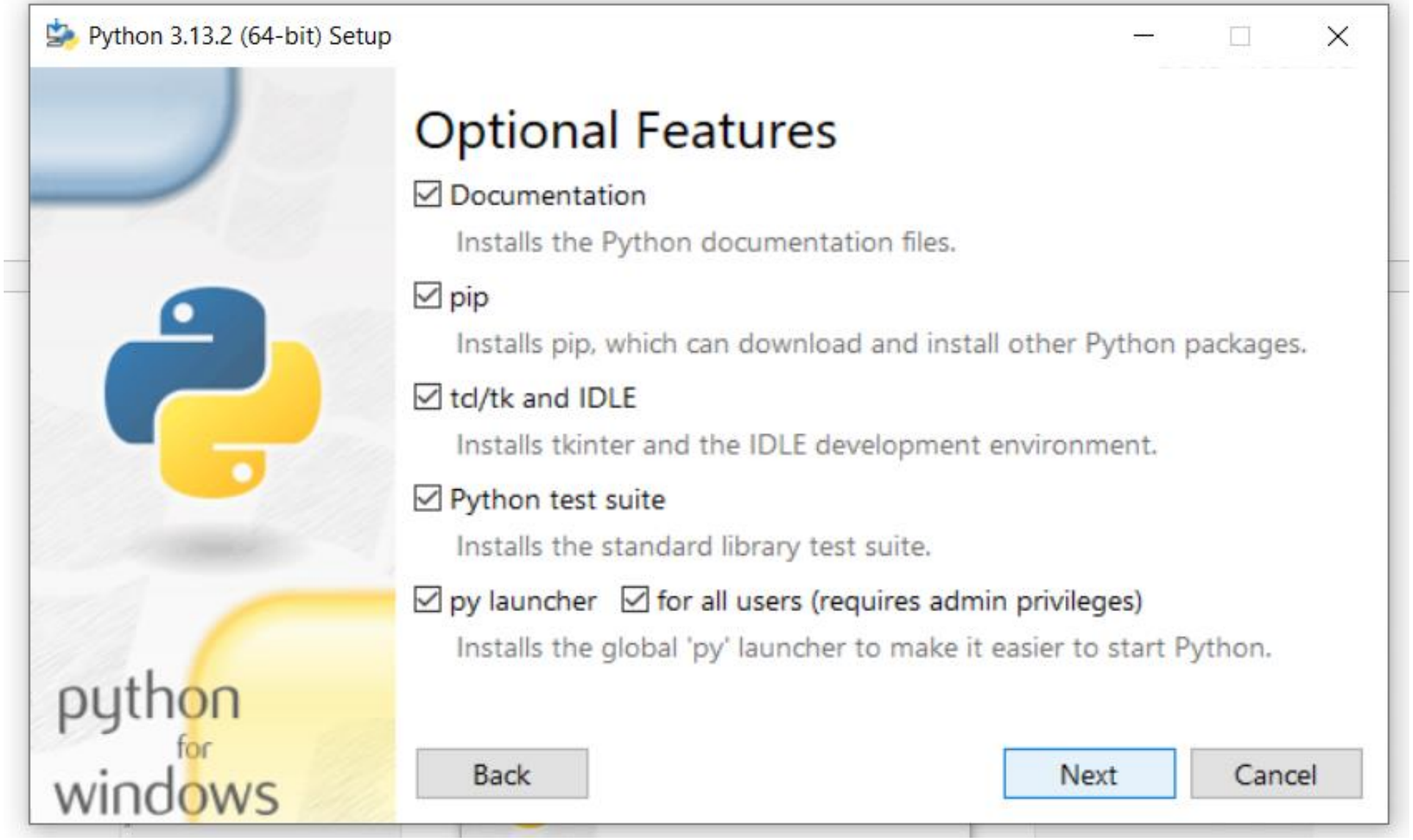
install python



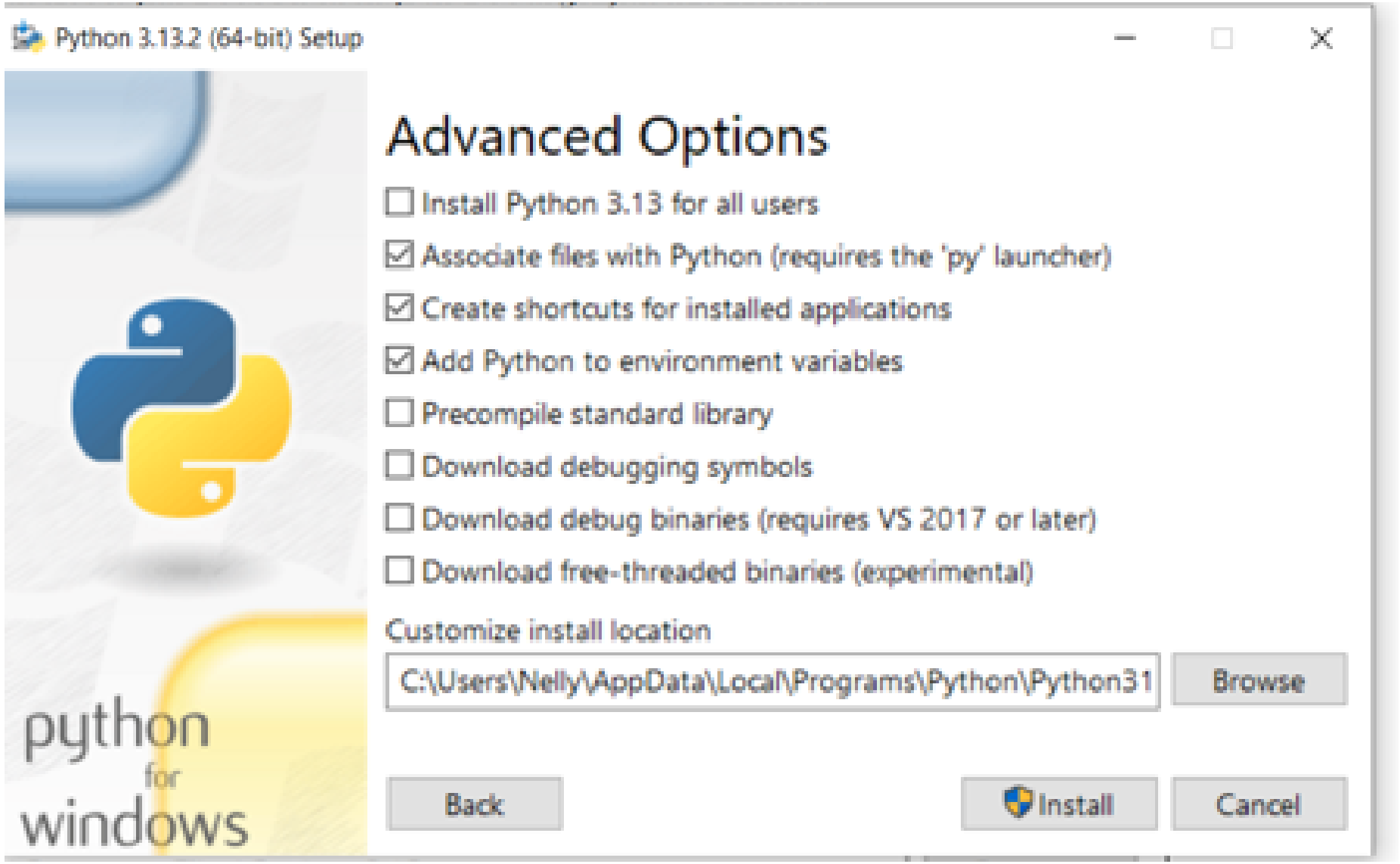
Python cont.

Customize installation

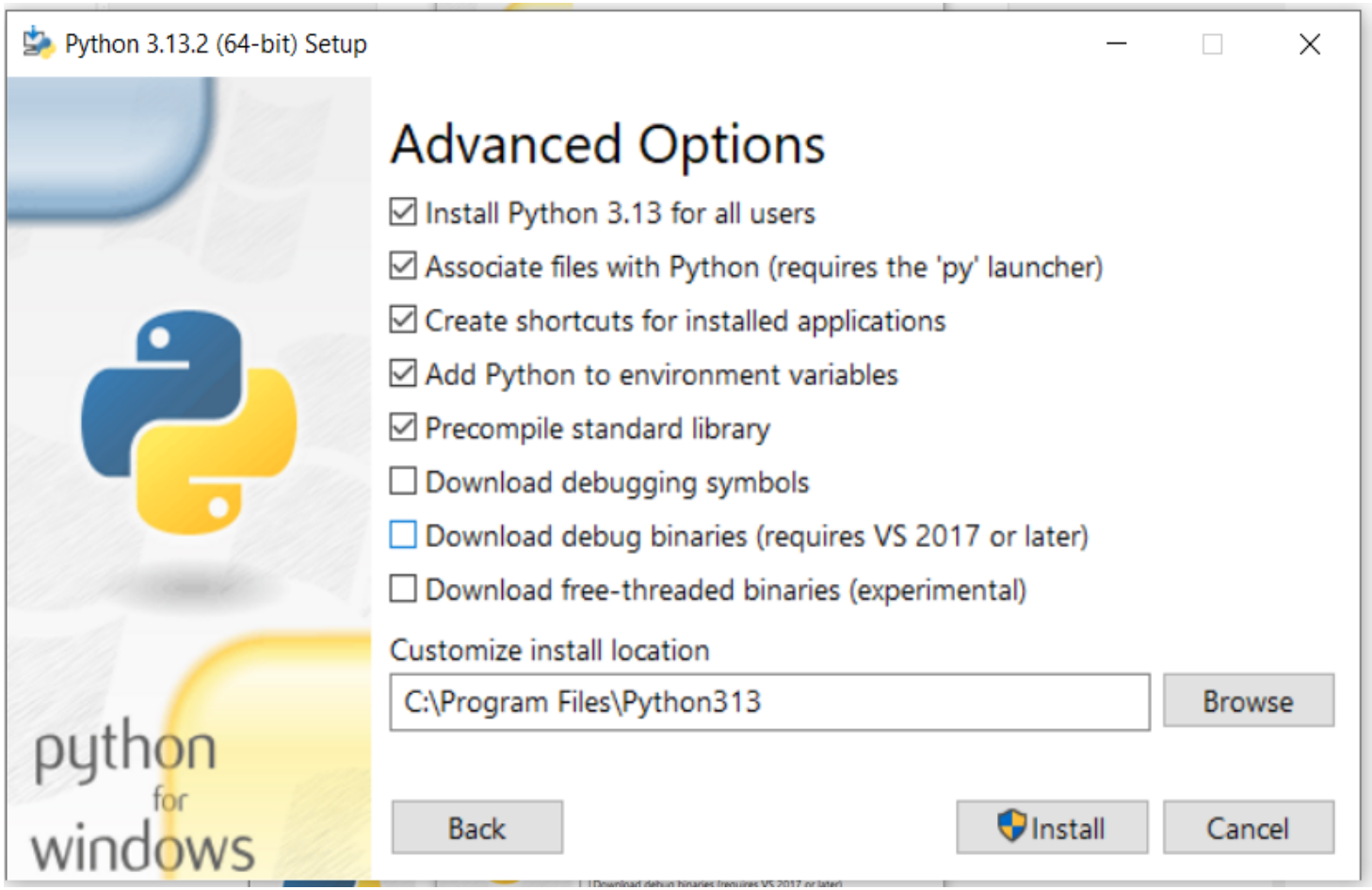
Python cont.


Python 3.13.2 (64-bit) Setup
Optional Features
Documentation
Installs the Python documentation files.
pip
Installs pip, which can download and install other Python packages.
tcl/tk and IDLE
Installs tkinter and the IDLE development environment.
Python test suite
Installs the standard library test suite.
py launcher
for all users (requires admin privileges)
Installs the global 'py' launcher to make it easier to start Python.
python
for
windows
Back
Next
Cancel

Python cont.


Python 3.13.2 (64-bit) Setup
Advanced Options
Install Python 3.13 for all users
Associate files with Python (requires the 'py' launcher)
Create shortcuts for installed applications
Add Python to environment variables
Precompile standard library
Download debugging symbols
Download debug binaries (requires VS 2017 or later)
Download free-threaded binaries (experimental)
Customize install location
C:\Users\Nelly\AppData\Local\Programs\Python\Python31
Browse
python
for
windows
Back
Install
Cancel

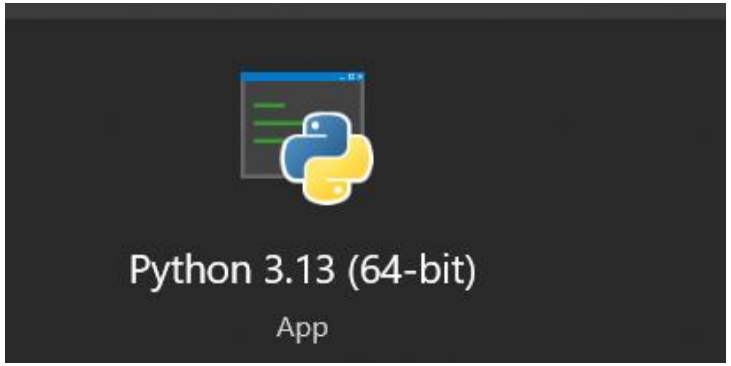
Python cont.



install

Python cont.

After finish setup, In start menu you have the following:



- This is python shell
- you run one line at a time, you can not save command
- You can access it according to its icon in start menu or by CMD (enter CMD and write python)

- python idle =Python's Integrated Development and Learning Environment)
- IDLE provides a simple text editor where you can write and save full Python programs (.py files).

This tool allows you to read Python's official documentation, including information about built-in modules, functions, and classes.

Libraries install

To install any library

From cmd you can install libraries, not from idle, not from shell(python 3.13(64-bite))

Type:
pip install library name

pip install pandas → Creating a DataFrame (Like an Excel Table)

pip install mlxtend → supports both Apriori & FP-Growth

Assignment #6

1- Given the following transactions, and the minimum count support=2, determine the frequent patterns (itemsets) using FP-Growth algorithm.

tid	Set of items
1	$\{Bread, Butter, Milk\}$
2	$\{Eggs, Milk, Yogurt\}$
3	$\{Bread, Cheese, Eggs, Milk\}$
4	$\{Eggs, Milk, Yogurt\}$
5	$\{Cheese, Milk, Yogurt\}$

2- By using Python , given the table in exercise 1, do the following:

A- determine the frequent patterns (itemsets) using Apriori and FP-Growth algorithms, with minimum support 40%

References.

[1] P. NING TAN, M. STEINBACH , A. KARPATNE , V. KUMAR, “INTRODUCTION TO DATA MINING”, (2nd Edition) (2nd. ed.), Pearson, 2018.

[2] Han, J., Pei, J., & Kamber, M, “ Data Mining: Concepts and Techniques” (3rd ed.). Morgan Kaufmann, 2011.

[3]Aguilar-Ruiz, J., Rodríguez -Baena, D., Alves, R. , “Frequent Pattern Mining.” In: Dubitzky, W., Wolkenhauer, O., Cho, KH., Yokota, H. (eds) Encyclopedia of Systems Biology. Springer, New York, 2013.

[4] C. C. Aggarwal, “Association Pattern Mining”, in: “Data Mining”, Springer, 2015.

[5] P.N Kavitha, “Comparative Analysis of Apriori and FP-Growth Algorithms For Frequent Item Sets ”, International Journal of Advanced in Management, Technology and Engineering Sciences, vol. XII, pp. 9-18, 2022

Advanced Topics in Information systems

SE204

Lecture 7

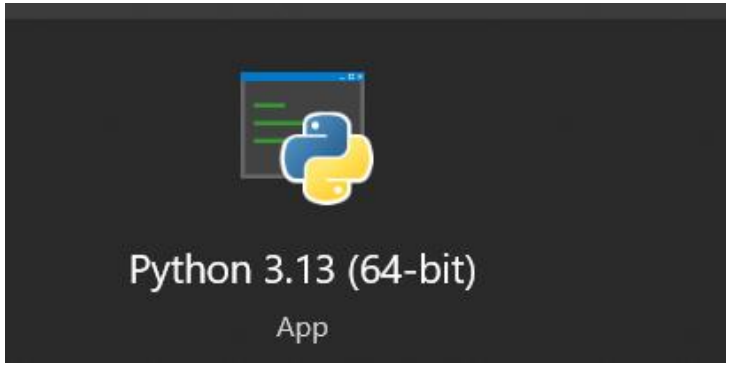
Dr. Nelly Amer

Agenda

- **Python for frequent pattern mining cont.**
- **What is machine learning.**
- **Machine learning and interpretability.**
- **Types of machine learning.**
- **Machine learning algorithms.**

Python cont.

After finish setup, In start menu you have the following:



- This is python shell
- you run one line at a time, you can not save command
- You can access it according to its icon in start menu or by CMD (enter CMD and write python)

- python idle =Python's Integrated Development and Learning Environment)
- IDLE provides a simple text editor where you can write and save full Python programs (.py files).

This tool allows you to read Python's official documentation, including information about built-in modules, functions, and classes.

Libraries install

To install any library

From cmd you can install libraries, not from idle, not from shell(python 3.13(64-bite))

Type:
pip install library name

Examples:

pip install pandas → Creating a DataFrame (Like an Excel Table)
pip install mlxtend → supports both Apriori & FP-Growth

How to Write and Run a Script in IDLE

1- From the Start Menu Open **IDLE (Python 3.13 64-bit)**.

2- Click **File → New File**.

3- Type your Python code, as:

```
print("Hello World")
x=5+3
print("x=",x)
```

4- Save the file from **File → save**(Ctrl + S) with a **.py extension** (e.g., test.py).

5- Run the script by clicking **Run → Run Module** (F5).

6- The output will appear in the **Python Shell (interactive window)**.

How to Write and Run a Script in IDLE cont.

Practical examples:

Open IDLE (Python 3.13 64-bit) and writ examples includes:

- Import libraries
- Input data
- Convert it to Boolean
- Use Apriori and fpgrowth functions, knowing the meaning of their parts
- Determine strongest association rules, according to minsupport and minconfidence
- How to handle NaN, and its reasons.
- Print rules
- Python case-sensitive.

What is machine learning.

Machine learning (ML)

Machine learning (ML) is a field of computer science that studies algorithms and techniques for automating solutions to complex problems that are hard to program using conventional programing methods.

Machine learning focuses on creating predictive models that learn from data and generalize unseen data

I.e., The larger the dataset, the more accurate they become

The conventional programming

The conventional programming

consists of two distinct steps:

The first step: is to create a detailed design for the program, i.e., a fixed set of steps or rules for solving the problem.

The Second step: is to implement the detailed design as a program in a computer language.

Machine learning vs conventional programming

The conventional programming

Design a program that sum 2 integers

input X , Y

Z= x+ y

Output z

You give the computer **Rules** , **Inputs** (X, Y)

The computer applies the rules and gives **output**

i.e., you **tell the computer what to do**.

Machine learning (ML)

Given the input data, and output data, design a model to predict the output for new input data.

Input data=[(1,3), (4,7), (8,12),……(7,20)]

output data=[(4), (11), (20),……(27)]

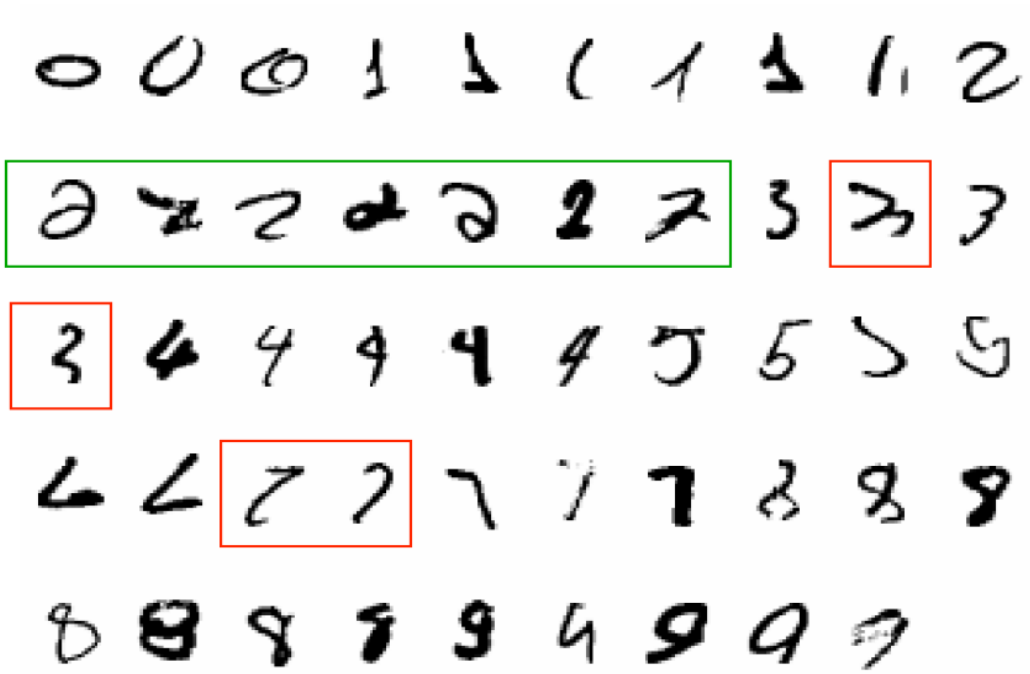
Machine learning algorithm find the relation between input and output

You give the computer **Input–Output pairs** (examples)

The computer finds the **rules** by itself

i.e., you **give it examples**, and the computer **figures out what to do**.

Machine learning cont.



Looking at the picture, decide which is a cat and which is a dog

given a picture, determine which symbol makes 2

This could be done :

By machine learning ✓

By conventional programming ✗

Machine learning cont.

ML research made slow and steady progress on solving complex problems until the mid-2000s, after which the progress in the field accelerated drastically. The reasons for this dramatic progress include:

- Availability of large amount of data due to the Internet, such as large datasets of images
- Availability of large amounts of compute power, supported by **large memory and storage space**
- Improved algorithms that are optimized for large datasets.

advancements in ML include the following:

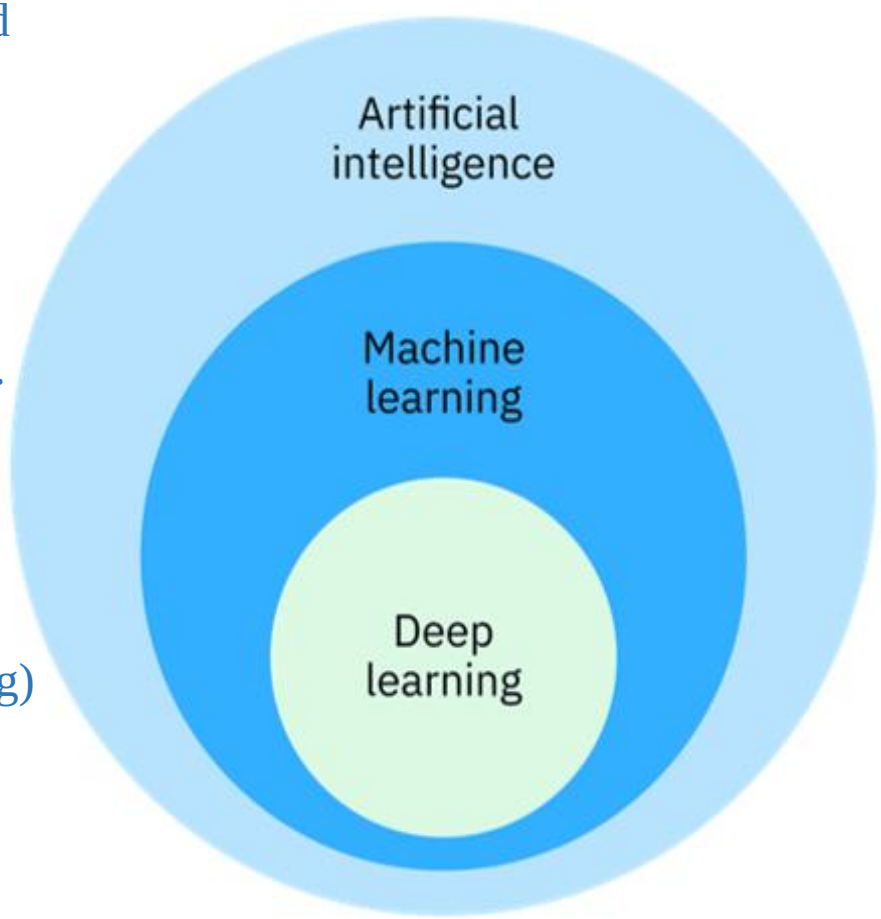
- ❑ Speech recognition – the ability to recognize speech and convert it to text.
- ❑ Language translation – understanding and forming language constructs without formal training in grammar, and a huge increase in the accuracy of translation compared to earlier methods.
- ❑ Driverless vehicles – the ability to navigate the vehicle without human intervention

Machine learning & Deep learning & Artificial intelligence (AI)

AI Goal: Create systems that can perform tasks that require human intelligence.
It enables computers to simulate human thinking, decision-making, learning, and problem-solving.

ML is a **subset of AI** that focuses on the idea that machines can **learn from data** and improve over time without being explicitly programmed for every task.

Goal: Build algorithms that can learn from data and make predictions or decisions.



Other techniques in AI are not based on ML:

- Expert Systems
- Traditional Computer Vision
- Game Playing
- Traditional NLP (natural language processing)

Deep learning is a subset of **machine learning (ML)** that uses **artificial neural networks with multiple layers** to model complex patterns in data.

Basic concepts

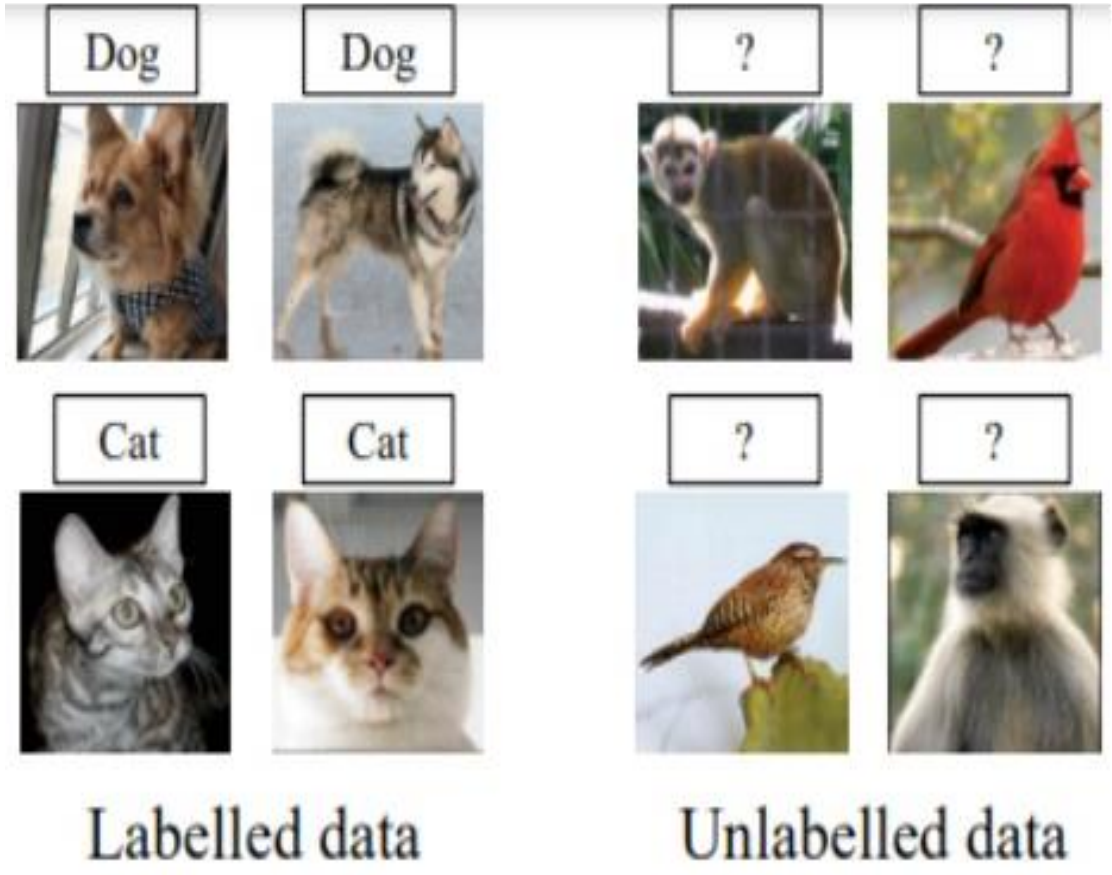
❑ Labeled data & unlabeled data

Labeled Data

Any data which has a characteristic, category, or attributes assigned to it can be referred to as labeled data. For example, a photo of a cat, a photo of dog.

Unlabeled data

Any data that does not have any labels specifying its characteristics, identity, classification, or properties can be considered unlabeled data. For example photos, videos, or text that do not have any category or classification assigned to it can be referred to as unlabeled data.



Basic concepts cont.

❑ Training set & testing set

Training data:

Training data is the data used for training the model.

Testing set:

Testing data is new data that the model **has never seen**. We use it to **evaluate how well** the model can generalize to new, real-world data.

❑ Overfitting & underfitting

Overfitting

Overfitting happens when a model learns the training data too well and doesn't generalize well to new, unseen data. It occurs when the model is very complex for the amount of training data given.

Underfitting

Underfitting happens when a model is too simple to learn the underlying pattern in the data. It performs poorly on both the training data and new, unseen data.

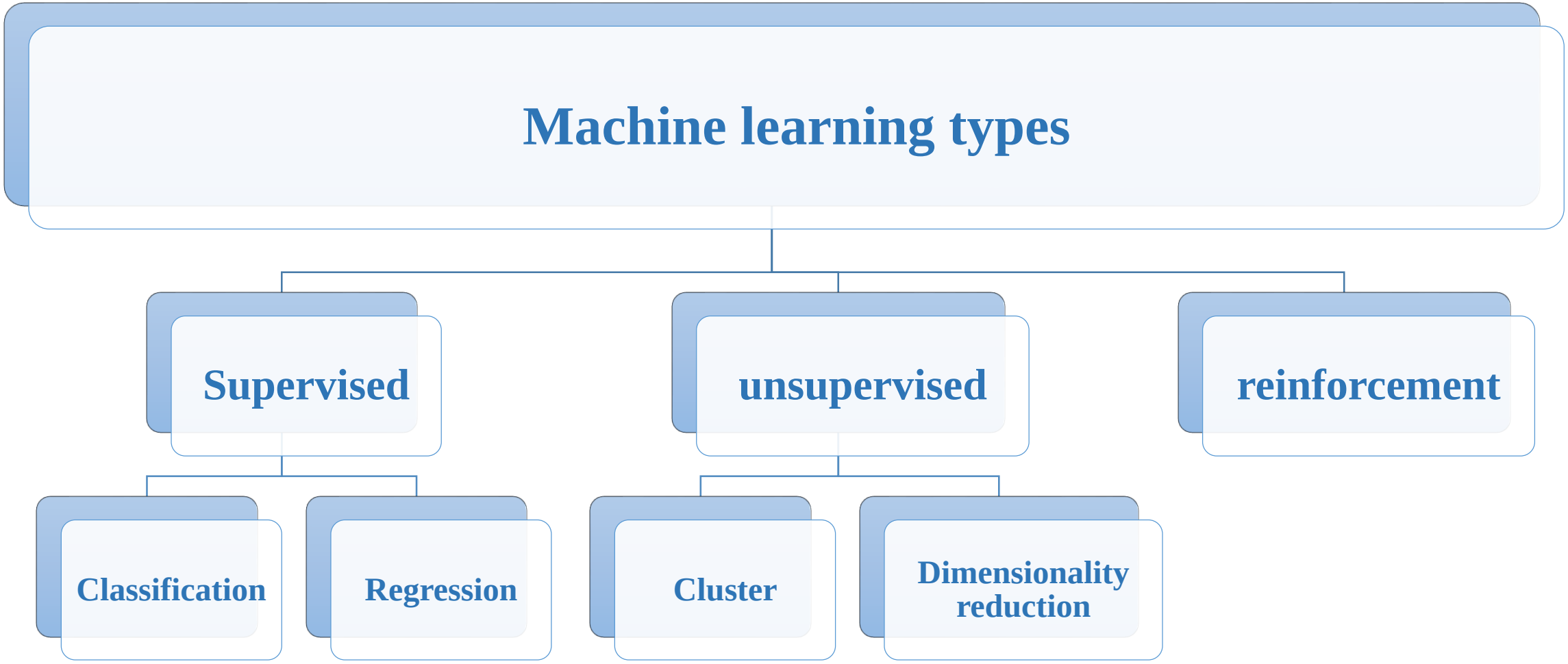
Machine learning types

Supervised learning → (classification, regression)

Unsupervised Learning → (cluster, Dimensionality reduction)

Reinforcement Learning

Machine learning types.



Supervised learning

Supervised machine learning is one of the most commonly used and successful types of machine learning.

It is applied when there is a need to predict a specific outcome based on input data, and we have examples of input and output pairs (labeled data).

In supervised learning, models are trained on labeled datasets, where each input is associated with a known output.

The goal is to learn a mapping function so the model can accurately predict outputs for new, never unseen inputs.

real-world applications of supervised learning :

Fraud detection
Spam filtering
Medical diagnosis
Email classification

Unsupervised learning.

Unsupervised machine learning is used when we do not have labeled data — that means we only have input data and no specific output.

The goal is to let the machine discover structures, or relationships in the data on its own.

Unsupervised learning is helpful when:
We don’t know the categories or groups in advance.
We want to group similar items.

The unsupervised learning algorithm looks at the input data and tries to group, cluster, or reduce dimensions based on similarities.

real-world applications of unsupervised learning :

Customer segmentation in marketing.
Organizing documents by topic
Reducing data for visualization

Reinforcement Learning

In the Reinforcement learning, an agent is interacting with an environment. The agent observes the state of the environment, and based on its observations, it can choose an action. Depending on the chosen action, it gets a reward. If the action was good, the agent will get a high reward and vice versa. **The goal of the agent is to find the best action for each state**.

How It Works

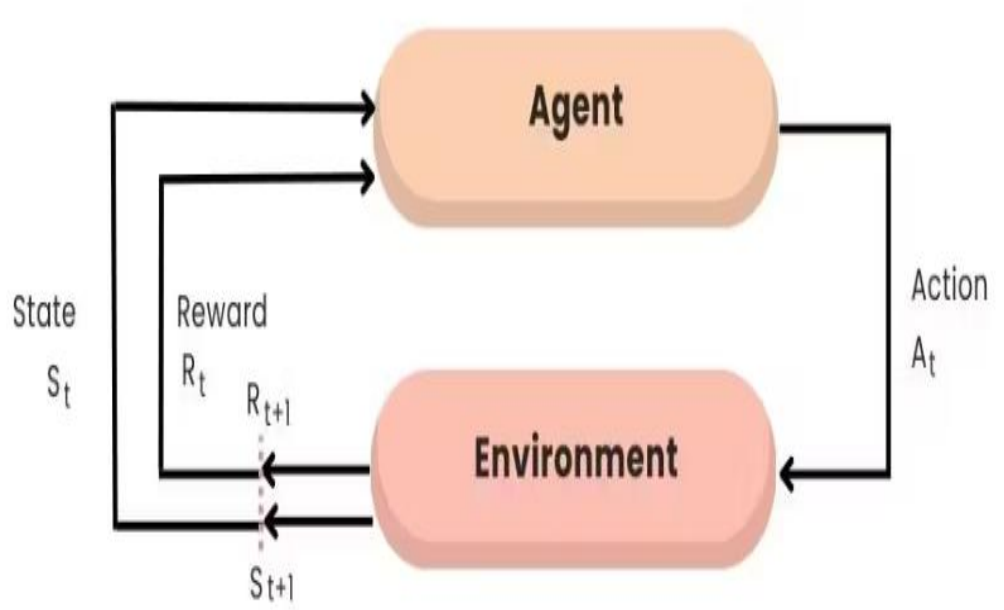
1- The agent observes the environment (where it is in a maze).

2- It takes an action (move left or right).

3- The environment gives a reward:
Positive: if it moves toward the goal
Negative: if it hits a wall

4- The agent learns: which actions lead to higher rewards.

5- Repeat: The agent keeps trying, improving its strategy over time.



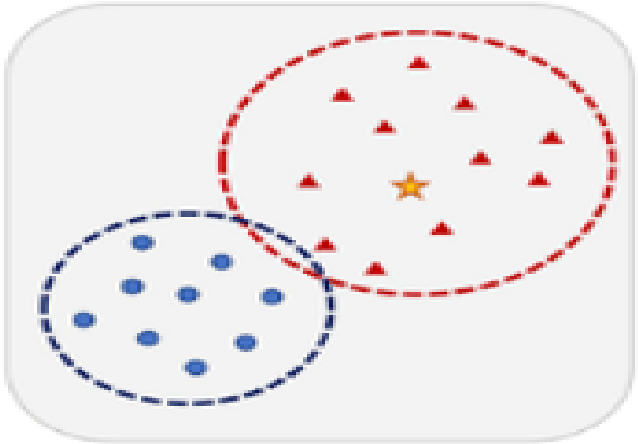
Example of reinforcement learning:

many robotics applications that learn how to walk

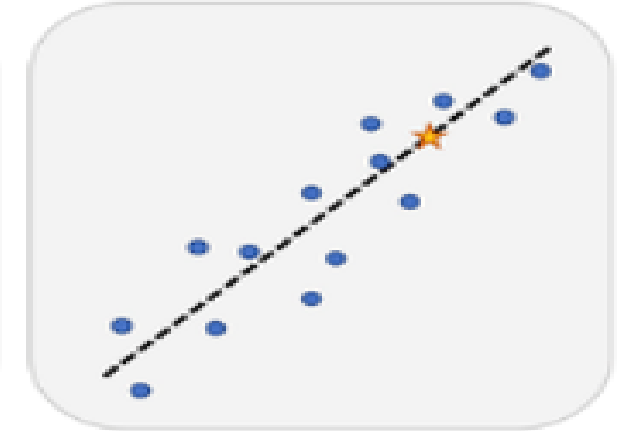
Supervised machine learning tasks

Supervised learning tasks

Classification



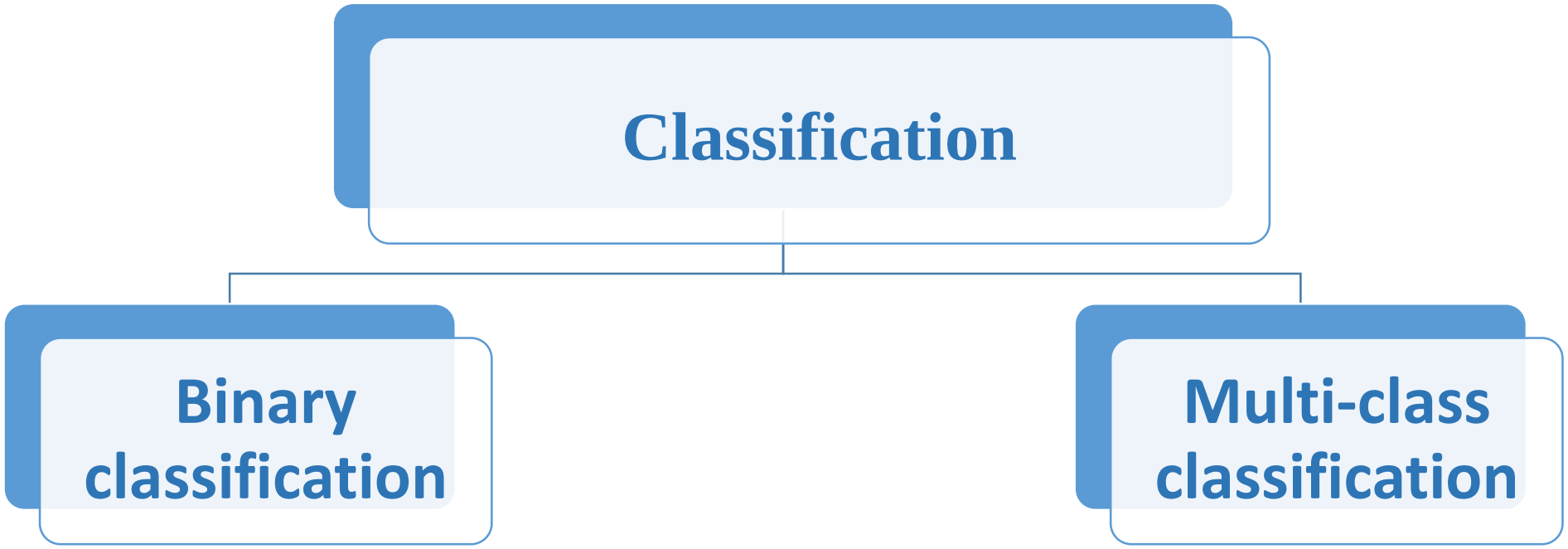
Regression



Classification machine learning.

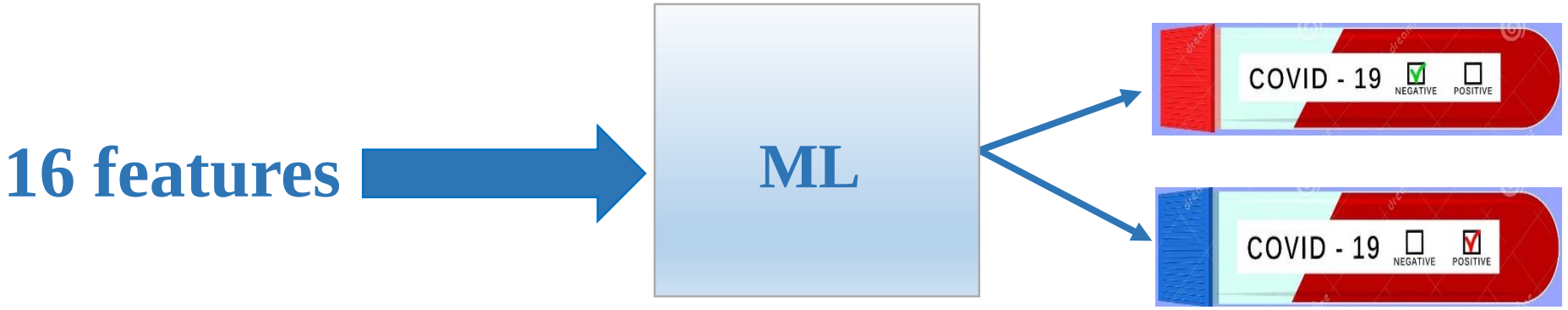
Classification is a supervised learning algorithm where a training set of labelled data is available. The model learned from training data to identify the category or class of the input feature or data is called classifier.

The classifier can be a binary classifier or a multi-class classifier.



Binary classification

A binary classifier identifies the input as belonging to one of the two output categories. For example, the mail received is a spam or not spam, Corona Virus Diagnosis classification.



Corona Virus Diagnosis classification.

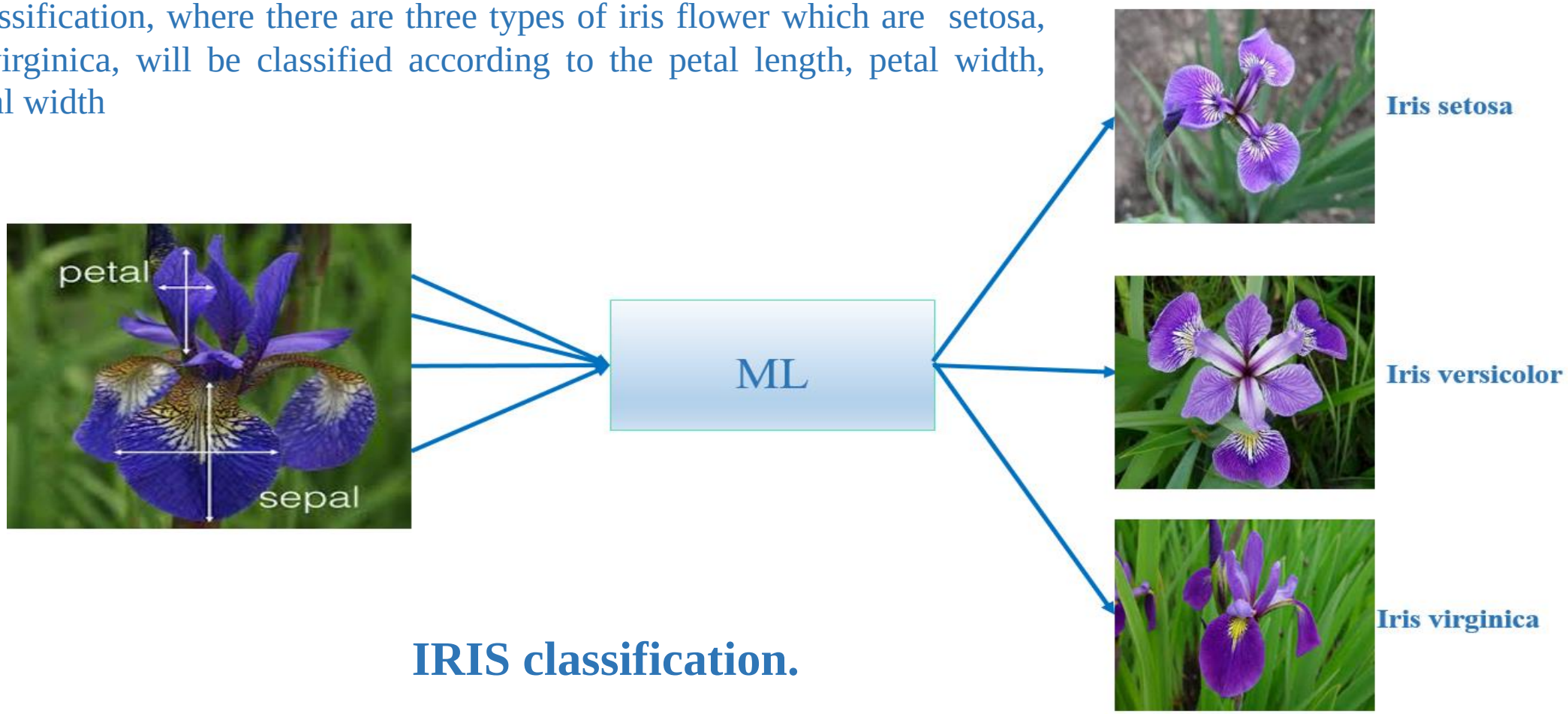
Multi-class classification

A multi-class classifier identifies the input vector as one of more than two categories.

For example:

1- the mail received is a promotional email that represents some kind of advertisement, personal email received from friends or associates, or a spam email.

2- Iris flower classification, where there are three types of iris flower which are setosa, versicolor, and virginica, will be classified according to the petal length, petal width, sepal length, sepal width



IRIS classification.

ROC Curve

The receiver operating characteristic (ROC) curve is a common tool used with **binary classifiers**.

The objective of the ROC curve is to evaluated the performance of a binary classification model.

How to Plot the ROC Curve:

The ROC curve is a graph with:

False Positive Rate (FPR) on the X-axis

True Positive Rate (TPR) (also known as Recall) on the Y-axis

Where:

FPR = FP / (FP + TN)= ratio of actual negative instances incorrectly classified as positive=

TPR = TP / (TP + FN)= ratio of actual positive instances correctly classified as positive

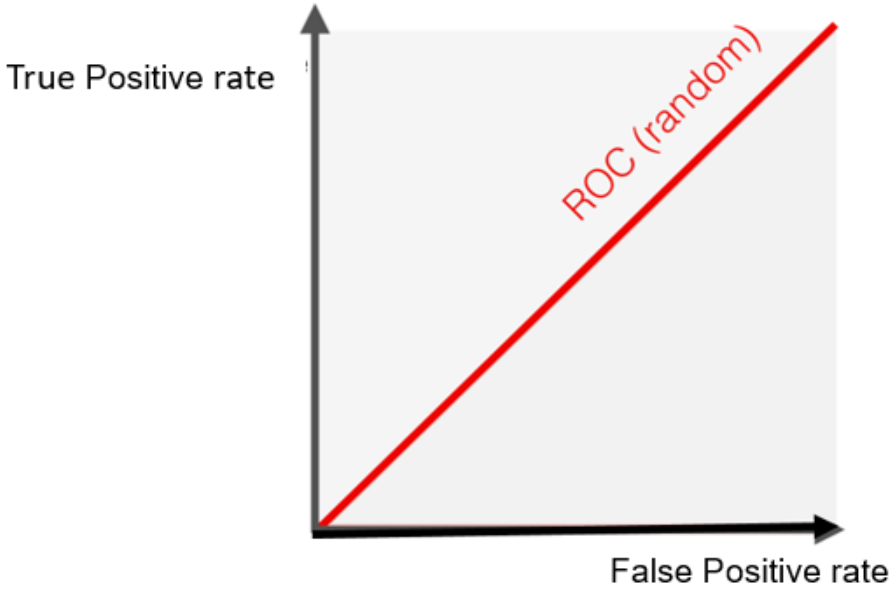
Types of ROC curve:

1- Diagonal ROC curve.

2- L-shaped ROC curve.

ROC Curve cont.

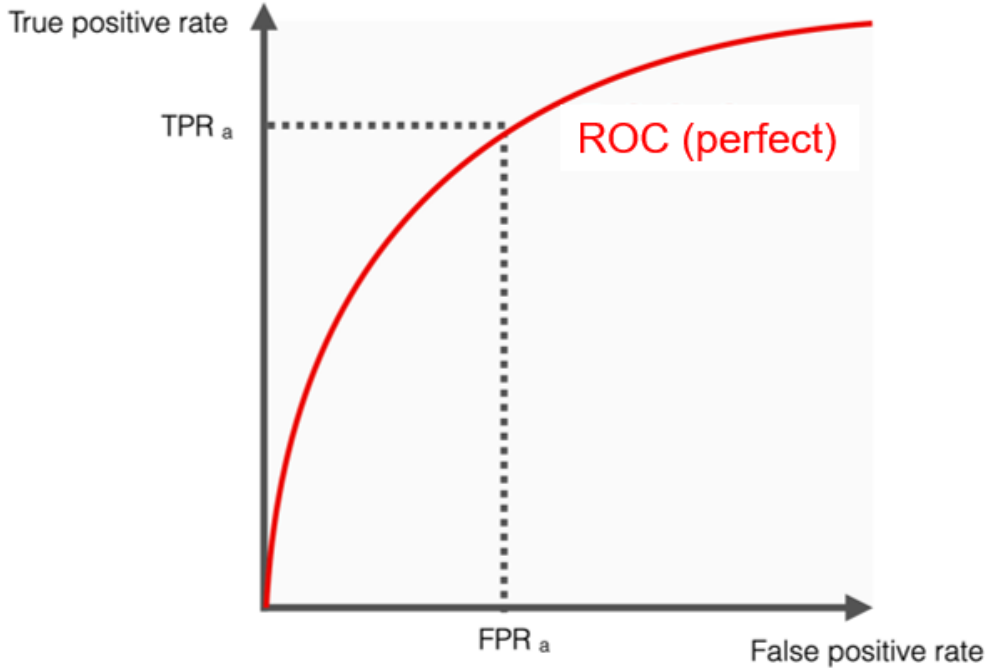
Diagonal ROC curve



Means the model performs like random guessing.
For every correct positive, there's a false one:
TP ≈ FP TN ≈ FN

It is bad model, since it cannot tell the difference between the two classes.

L-shaped ROC curve



Means the curve rises quickly toward the top-left corner.
Model makes many correct predictions:

- True Positives (TP) are high
- False Positives (FP) are low

It is great model — it separates the classes well.

Classification machine learning algorithms.

Classification algorithms

- **Logistic Regression**
- Decision Trees for Classification
- Artificial Neural Networks (ANNs) for Classification
- Support Vector Machines (SVM)
- Random Forest
- K-Nearest Neighbors (KNN)

We will focus on Logistic Regression

Logistic regression

Logistic Regression is a classification algorithm, which allows you to perform binary classification or multi-class classification.

For example:
Classify emails as spam or ordinary.
Classify iris flowers into Iris setosa, Iris virginica, or Iris versicolor.

In binary classification:
The sigmoid function is used to compute the estimated output $\hat{y}$:
$\hat{y} = \frac{1}{1+e^{-z}}$,
where $\hat{y}$ is the predicted output, and z is linear combination of features X (i.e., $Z=b_0+b_1x_1+b_2x_2+\ldots\ldots+b_n x_n$)

The performance of the classification model is evaluated during training using a cost function in the form of a loss function:
For binary classification, the loss function is:
function : $L(y, \hat{y}) = -[y * log(\hat{y}) + 1 - y(log(1 - \hat{y})]$.
where: y is the true label (either 0 or 1), $\hat{y}$ is the predicted probability (output of the sigmoid function).

In multi-class classification:
The softmax function is used to compute the probabilities for each class.

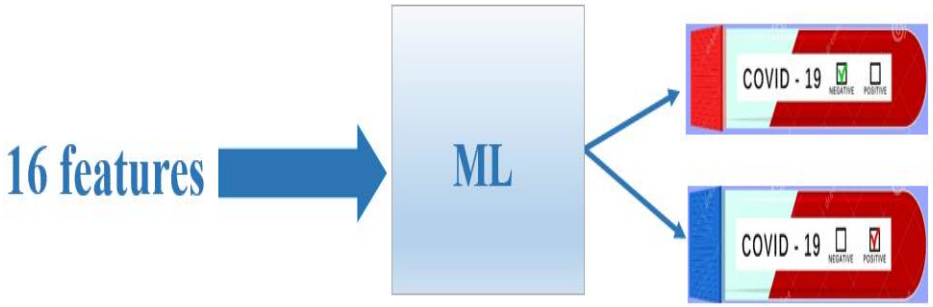
Logistic regression

We have input vector of X (features of blood analysis for 20 individuals)

and their Y vector output as +VE Corona virus or -VE Corona virus) :

$X=[(x_1,…..x_{16})_1, (x_1,…..x_{16})_2 ……,……..(x_1,…..x_{16})_{20}]$

Y=[-VE, +VE, ……...,+VE]



How the algorithm work:

1. Initialize the coefficients in the sigmoid function: $\hat{y} = \frac{1}{1+e^{-z}}$, where $\hat{y}$ is the predicted output, and z is linear combination of features X (i.e., $Z=b_0+b_1x_1+b_2x_2+……+b_n x_n$)
2. Enter The first instance.
3. Compute the predicted output $\hat{y} = \frac{1}{1+e^{-z}}$
4. Compute the error between the predicted output and the actual output using the cost function in the form of loss function : $\boldsymbol{L(y,\hat{y}) = -[y * \log(\hat{y}) + 1 - y(\log(1-\hat{y})]}$.
5. Update the coefficients of sigmoid function to minimize the error using **the gradient descent algorithm**.
6. Input the second instance and repeat the steps from 3 to 5 until end all instances as the first iteration,
7. Do more iterations until the model reaches convergence, hence reach the high accuracy.

Assignment #7

1- Given the following transactions, determine the strongest association rules with support 40% and confidence 60%., and print it, using python

tid	Set of items
1	$\{Bread, Butter, Milk\}$
2	$\{Eggs, Milk, Yogurt\}$
3	$\{Bread, Cheese, Eggs, Milk\}$
4	$\{Eggs, Milk, Yogurt\}$
5	$\{Cheese, Milk, Yogurt\}$

References.

[1] G. Rebala, A. Ravi, S.Churiwala “An Introduction to Machine Learning”, Springer, 2019

[2] A. Géron, “Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow”, O’Reilly, 2019