Module 2

Data Warehouse Implementation

Nithin Kumar
Asst. professor
Dept of CS&E
VVCE, Mysuru

Data Warehouse contain huge volume of data. OLAP Servers demand that decision support queries be answered in the order of seconds.
Therefore, it is crucial for data warehouse systems to support highly efficient cube computation techniques, access methods & query processing techniques.

* Efficient Data Cube Computation :-
Multidimensional data Analysis is the efficient computation of aggregations across many sets of dimensions.
In SQL, these aggregations are referred as "group-by". Where Each group-by can be represented by a "Cuboid", set of group-by's form a lattice of Cuboids defining a data cube.

* Compute Cube Operation :- The "Compute Cube" operators computes aggregates over all subsets of the dimensions specified in the operation.
This can require excessive storage space, especially for large number of dimensions.

→ Consider the example of "AllElectronics" sales data cube that contains the following "city", "item", "year" & "sales-in-dollars" you want to be able to analyze the data, with queries such as follows
  * "compute the sum of sales, grouping by City & Item"
  * "Compute the sum of sales, grouping by City"
  * "Compute the sum of sales, grouping by Item"

→ The total no of Cuboids, or Group-by's that can be computed for this data cube is $2^3 = 8$. (City, Item, year)

The possible group-by's are as : {(City, Item, year), (City, Item), (City, year), (Item, year), (City), (Item), (year), ()}. where "()" means that the group-by is Empty.

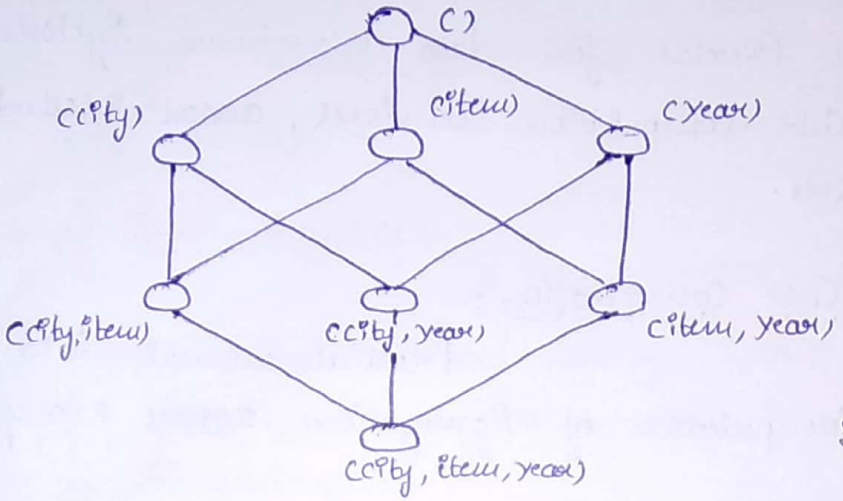


Fig:- Lattice of Cuboids

→ The "Base Cuboid" Contains all three dimensions "city", "Item" & "year". It can return the total Sales for any Combination of three dimensions.

→ The "Apex Cuboid" or "0-D" refers to the case where the group-by is Empty. It contains the total Sum of all Sales.

→ The Base Cuboid is the least generalized (Most Specific) of the Cuboids.
The Apex Cuboid is the Most generalized (Least Specific) of the Cuboids.

→ An Sql query containing no group by (E.g "Compute the Sum of total Sales") is a zero dimensional operation.

An Sql query containing one group by (E.g "Compute the Sum of Sales, grouping by City") is a one Dimensional operation.

A Cube operator on n dimensions is Equivalent to a Collection of group-by Statements.

→ The SQL Syntax could be defined as

"define Cube Sales_Cube [City, year, Item] : Sum (Sales_in_dollars)"

→ A Statement Such as

"Compute Cube Sales_Cube"

would Explicitly cued to Compute the Sales Aggregate Cuboids for all Eight Subsets of the set [City, year, Item], Including the Empty Subset.

* Curse of Dimensionality :-

A Major challenge related to precomp-utation is that the required Storage Space may Explode if all Cuboids in data cube are precomputed. Especially when the Cube has Many dimensions.

The Storage requirements are Even More Excessive when Many of dimensions have associated Concept hierarchies. This problem is referred as "Curse of Dimensionality".

→ For Example, time is usually Explored not at only one Concept-ual level (e.g. year)

But rather at Multiple Conceptual levels Such as in the hierarchy "day < Month < Quarter < Year".

→ For an n-dimensional Data Cube, the total no of Cuboids that can be generated is

$$\text{Total no of Cuboids} = \prod_{i=1}^{n} (L_i + 1)$$

where L_i is the number of levels associated with the Dimension i. one is added to L_i to include the Virtual top level, all.

* Partial Materialization : Selected Computation of Cuboids :- There are three choices for Data Cube Materialization

→ "No Materialization" : Do not precompute any of the "non-base" Cuboids.
This leads to computing Expensive Multidimensional Aggregates on-the-fly, which can be Extremely Slow.

→ "Full Materialization" : precompute all the Cuboids. The resulting lattice of Computed Cuboids is referred as "Full Cube".
This choice typically requires huge amounts of Memory Space in order to Store all precomputed Cuboids.

→ "partial Materialization" : Selectively compute a proper Subset of the whole Set of possible Cuboids.
We May compute a Subset of the Cube, which contains only those cells that satisfy Some User-Specified Criterion, Such as where the tuple count of Each Cell is above Some threshold.

→ The partial Materialization of Cuboids or Subcubes Should consider three factors
  * Identify the Subset of Cuboids or Subcubes to Materialize.
  * Exploit the Materialized Cuboids or Subcubes during query processing.
  * Efficiently update the Materialized Cuboids or Subcubes during Load & Refresh.

→ Selection of Subset of Cuboids or Subcubes to Materialize Should take into account the queries in workload, their frequencies, & their accessing costs. In addition to Storage requirements.

→ A popular Approach is to Materialize the Cuboids Set on which other frequently referenced Cuboids are based.
We can compute an "Iceberg Cube" which is a data cube that

Stores only those cube cells with an aggregate value.

→ Another common strategy is to Materialize a "Shell Cube", this involves precomputing the cuboids for only a small number of dimensions (e.g. three to five) of a Data Cube.

→ Once the Selected Cuboids have been Materialized, it is important to take advantage of them during query processing. Such as

* How to determine the relevant Cuboids from the Materialized Cuboids
* How to use available Index Structures on Materialized Cuboids.
* How to Transform the OLAP operations onto the Selected Cuboids.

→ Finally, during load & Refresh, the Materialized Cuboids should be updated Efficiently.

* Indexing OLAP Data :-

Indexing will be done on OLAP data to facilitate Efficient data accessing, Most data Warehouse should Support Index Structures & Materialized Views (Cuboids) for Efficient Data accessing.

There are two ways of Indexing OLAP data

* Bitmap Indexing
* Join Indexing

* Bitmap Indexing :- It a popular Indexing Method, because it allows quick Searching in data cubes.

→ In the bitmap index for a given attribute, there is a distinct bit vector (BV),

For Each value 'v' in the attribute's domain. If a given attribute's domain consists of n values, then 'n' bits are needed for Each Entry in the bitmap index.

→ If the attribute has the value V for a given row in the data table, then the bit representing that value is set to '1' in corresponding row of bitmap index.
All other bits for that row are set to '0'.

→ Consider an Example of "All Electronics" data warehouse, Suppose the dimension "Item" at the top level has four values such as "Home Entertainment", "Computer", "phone" & "Security".

Base table

RID	Item	City
R_1	H	V
R_2	C	V
R_3	P	V
R_4	S	V
R_5	H	T
R_6	C	T
R_7	P	T
R_8	S	T

Item bitmap table

RID	H	C	P	S
R_1	1	0	0	0
R_2	0	1	0	0
R_3	0	0	1	0
R_4	0	0	0	1
R_5	1	0	0	0
R_6	0	1	0	0
R_7	0	0	1	0
R_8	0	0	0	1

City bitmap table

RID	V	T
R_1	1	0
R_2	1	0
R_3	1	0
R_4	1	0
R_5	0	1
R_6	0	1
R_7	0	1
R_8	0	1

→ Bitmap Indexing is advantageous compared to "hash" or "Tree Indices"
Because comparison, Join & aggregation operations are reduced to bit Arithmetic which substantially reduces the processing Time.

→ Bitmap Indexing provides a significant reductions in Space & Input/output (I/O) since a String of characters can be represented by a single bit.

* Join Indexing :- Method gained popularity from its use in relational database query processing.

→ Traditional Indexing Maps the value in a given column to a list of rows having the value.

→ Join Indexing registers the joinable rows of two relations from a relational database.
For ex: If two relation R(RID, A) and S(B, SID) Join on

the attributes 'A' and 'B', then the Join Index record contains the pair (RID, SID), where RID & SID are record Identifiers from the R and S Relations.

→ Join Index records can identify joinable tuples without performing costly Join operations.

→ The Star Schema Model of data Warehouse Makes Indexing attractive for cross-table search.
Because the linkage between a fact table & its corresponding dimension table comprises the fact table's foreign-key & the Dimension table's primary key.

→ Join Indexing maintains relationships between attribute values of a dimension (within dimension table) & the corresponding rows in the Fact table.

→ Consider an Example of "All Electronics" Star Schema of a Join Index relationship between the "Sales" fact table & the "location" & "Item" Dimension tables

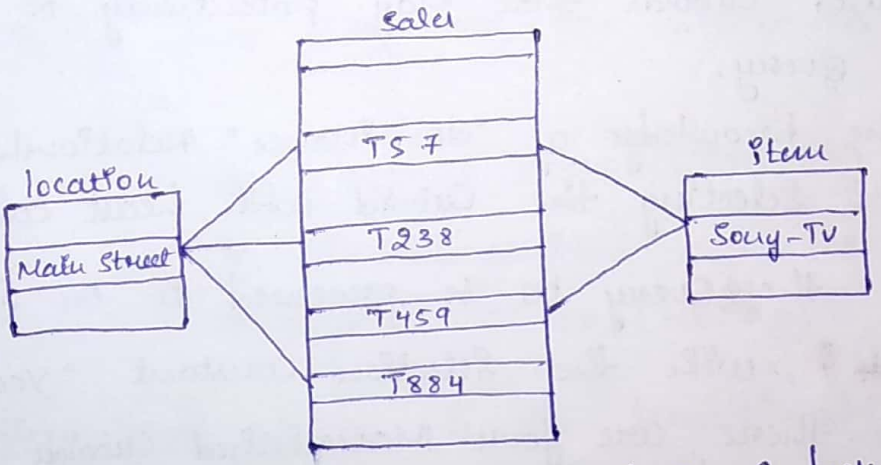


Fig: Linkage between Sales fact table & location & Item Dimension table

location	Sales-key
...	...
Main Street	T57
Main Street	T238
Main Street	T884
...	...

Item	Sales-key
...	...
Sony-TV	T57
Sony-TV	T459
...	...

location	Item	Sales-key
...	...	...
Mainstreet	SonyTV	T57
...	...	...

→ To Speed-up query processing, the Join Indexing & bitmap Indexing Methods can be Integrated to form "bitmapped Join Indices".

* Efficient processing of OLAP queries :-
The purpose of Materialization Cuboids & constructing OLAP Index Structures is to Speed up query processing in data Cubes.

The query processing in Data cube should proceed as follows.

* Determine which operation Should be performed on available Cuboids :- This involves transformation of any selection, projection roll-up or drill-down operations Specified in the query into Corresponding SQL & OLAP operation.

* Determine to which Materialized Cuboid(s) the relevant operation Should be performed :- This involves Identifying all of the Materialized Cuboids that may potentially be used to answer the query.
By using the knowledge of "dominance" relationships among the Cuboids & Selecting the Cuboid with least cost.

→ Suppose that the query to be processed is on {Brand, province_or_State}, with the selection constant "year = 2010" Also Suppose that there are four Materialized Cuboids available, as follows

* Cuboid 1: {year, Item_name, city}
* Cuboid 2: {year, brand, country}
* Cuboid 3: {year, brand, province_or_State}
* Cuboid 4: {Item_name, province_or_state} where year = 2010

"which of these four Cuboids should be Selected?"

→ Finer-granularity data cannot be generated from Coarser-granularity data.
Therefore, "Cuboid 2" cannot be used because "Country" is a More general concept than "province_or_State".
"Cuboids 1, 3 & 4" can be used to process the query.

→ "How would the costs of Each Cuboid compare if used to process the query?"

* Cuboid 1 would cost the most because both "Item_name" & "City" are at lower level than "province_or_State" & "Brand".
* Cuboid 3 is Smaller than Cuboid 4 & thus Cuboid 3 Should be choosen to process the query.
If Efficient indices are available for Cuboid 4, then Cuboid-4 May be a better choice.

* OLAP Server Architectures ROLAP versus MOLAP versus HOLAP

Cubes in a data Warehouse are Stored in three different Modes.

* Relational OLAP (ROLAP)
* Multidimensional OLAP (MOLAP)
* Hybrid OLAP (HOLAP)

* Multidimensional OLAP :- It is the traditional Mode in OLAP analysis. In MOLAP data is Stored in form of Multidimensional Cubes & not in Relational databases.

→ The advantages of this Mode is that it provides Excellent query performance & the Cubes are built for fast data retrieval.
All Calculations are pre-generated when the Cube is Created & can be Easily applied while querying data.

→ The disadvantages of this Model are that it can handle only a limited amount of data.
Since all calculations have been pre-built when the Cube was created, the Cube cannot be derived from a large volume of data.

* Relational OLAP :- Data in this Model is stored in Relational databases. Since the data is stored in relational databases, this Model gives the appearance of Traditional OLAP's Sliding & dicing functionality.
→ The advantage of this Model is it can handle a large amount of data & can leverage all the functionalities of the relational database
→ The disadvantages are that the performance is slow & Each ROLAP report.
It is also limited by SQL functionalities.

* Hybrid OLAP :- Approach combines ROLAP & MOLAP Technology. Benefiting from the greater Scalability of ROLAP & Faster Computation of MOLAP.
→ HOLAP leverages Cube technology & for drilling down into detail it uses the ROLAP Model.
The Microsoft SQL Server 2000 Supports a hybrid OLAP Server.

* Specialized SQL Servers :- To Meet the Growing demand of OLAP processing in relational databases, Some database System Vendors implement specialized SQL Servers to implement
  * Advanced Query language
  * Advanced Query processing

Over "Star" & "Snowflake" Schemas in ready-only Environment

* Data mining :-

Data Mining is a technology that leads traditional Data analysis Method with Sophisticated Algorithms for processing large volume of data.

→ It has also opened up Exciting opportunities for Exploring & Analyzing new types of data & for analyzing diff types of data in new ways.

"Data Mining is a process of automatically discovering useful Information in Large Data Repositories".

* Data Mining & Knowledge Discovery :- Data Mining is an integral part of Knowledge Discovery in Database (KDD). which is a overall process of converting raw data into useful info.

→ This process consists of a Series of Transition Steps from "Data preprocessing" to "post processing of Data Mining Results".

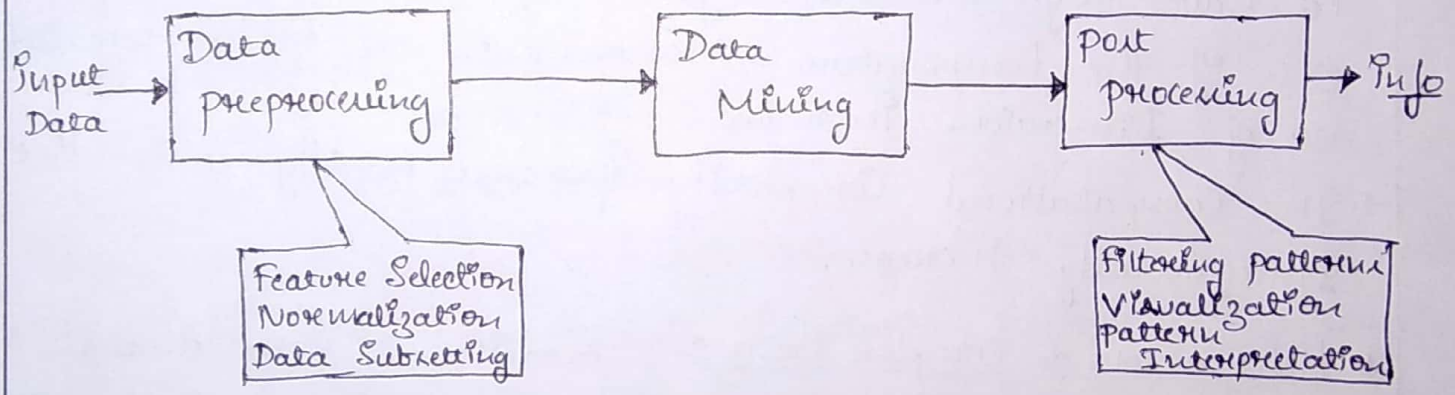


Fig :-

The Input data can be Stored in a Variety of files formats Such as Flat files, Spread Sheets or relational tables.

→ The purpose of preprocessing is to Transform the raw Input data into an appropriate Format for Subsequent Analysis.

The Steps Involved in Data preprocessing are

* Fusing Data from Multiple Sources.

* Cleaning Data to remove noise.
* Selecting records & features to data Mining, this is the most time consuming step in knowledge discovery process.

→ "post processing" steps that ensures that only valid & useful results are incorporated into the Decision Support System.

* Challenges :-
The Motivating challenges of Data Mining are.

* Scalability :- Advances in Data generation & collection of data sets with sizes of Giga bytes, Terabytes or Even Petabytes are become common.
→ If the data Mining Algorithms are to handle these Massive data sets, then they Must be Scalable.

* High Dimensionality :- Its now common to Encounter data sets with hundreads or thousands of Attributes
→ Data sets with temporal or Spatial components also tend to have High Dimensionality.
Ex :- If the temperature Measurements are taken Repeatedly no of Dimensions Increases.
→ The Computational Complexity Increases Rapidly with the Dimensionality Increase.

* Heterogenous & Complex Data :- Traditional Data Analysis Methods often Deal with data sets Containing Attributes of the Same type, Either Continous or Categorical.
→ The role of Data Mining in Business, Science, Medicine & other fields have grown, so has the need for Techniques that handle heterogenous Attributes.

* Data Ownership & Distribution :- Sometimes the Data Model for Analysis is not Stored in one location or owned by one org Instead data is Geographically Distributed.

→ The key challenges faced by distributed data Mining Algorithms include

* How to reduce the Amount of communication required to perform Distributed computation.
* How to Efficiently consolidate the Data Mining results obtained from Multiple Sources.
* How to Address Data Security Issues.

* Non-Traditional Analysis :- Traditional Statistical Approach is based on a "hypothesize-and-Test" paradigm.

→ Current Data Analysis tasks often requires the Generation & Evolution of thousands of hypothese & Consequently the development of some data Mining techniques has been Motivated. It is a non-traditional Approach.

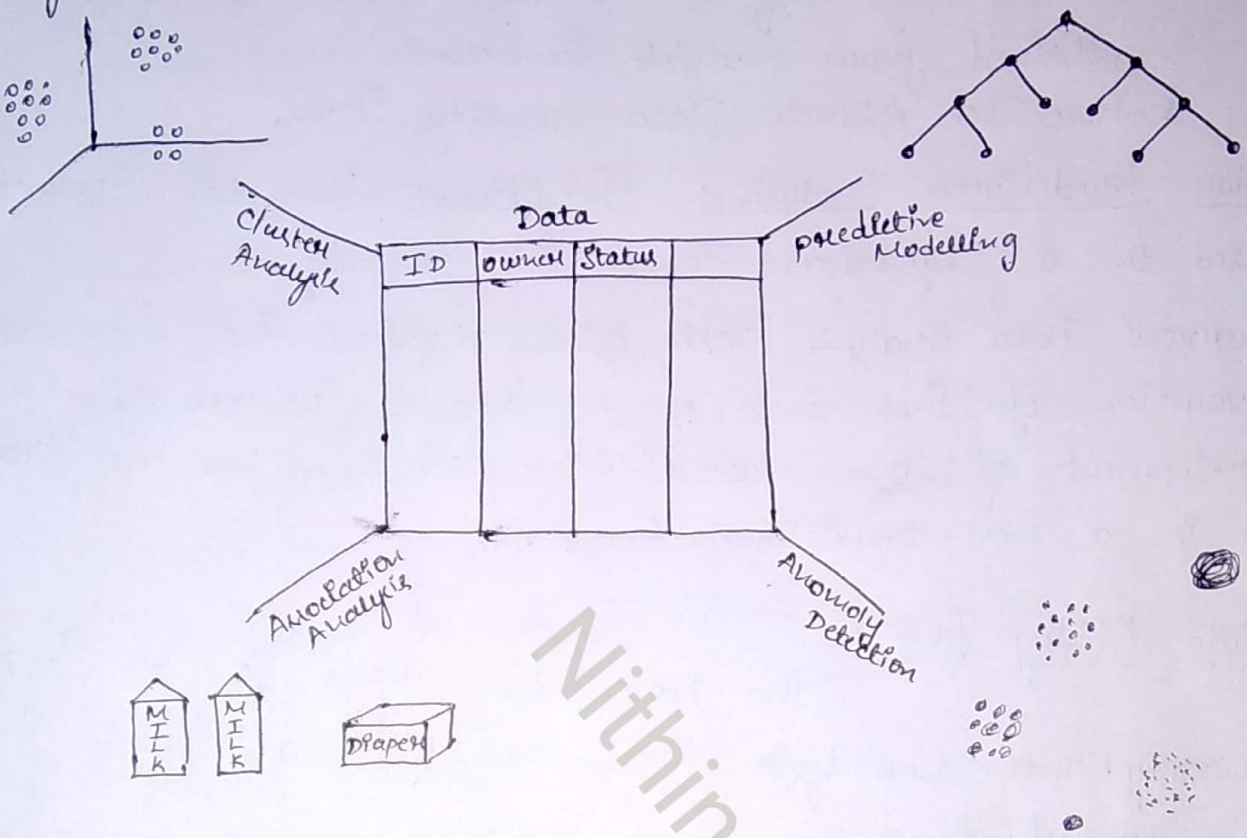
* Data Mining Tasks :- The Data Mining tasks can be classified generally into two types based on what a Specific task tries to achieve.
  * predictive Tasks
  * Descriptive Tasks.

There are a number of Data Mining task such as Classification, prediction, Time-Series Analysis, Association, Clustering, Summerization, Pon Etc. All these tasks are Either predictive tasks or Descriptive tasks.

* predictive Tasks → Usually from the available data Set that is helpful in predicting unknown or future values of another data Set of interest.
A Medical practioner trying to diagnose a disease based on the Medical test results of a patient can be considered as a predictive Data Mining task.

* Descriptive Tasks → Usually finds data describing patterns & comes up with new, Significant Information from the Avail -able Data Set.

A retailer trying to Identify products that are purchased together can be considered as a descriptive data Mining Task.



* predictive Modeling :- Is a process that uses Data Mining & probability to Forecast outcomes.

→ There are two types of predictive Modeling

* Classification → used for Discrete Variables
* Regression → used for continous Variables.

The Goal of both tasks is to learn a Model, in which Error between the predicted & True values is low.

→ For Example, a Model can predict the Income of an Employe based on Education, Experience & other demographic factors like place of Stay, gender Etc.

* Anomaly Detection :- is a task of identifying observations whose characteristics are significantly different from the rest of Data.
It is also called as "outlier Detection".

→ A good Anomaly detector must have a high detection rate & a low fake alarm rate.
The good anomaly detection Algo is to discover real Anomalies & Avoid Fake labeling.

→ Typically the anomalous items will translate to some kind of problem such as bank fraud, structural defect, Medical problems etc

* Association Analysis :- Association discovers the association or connection among a set of items. Association identifies the relationships among objects.

→ Association Analysis is used for Commodity Management, Advertising, direct Marketing Etc.

→ For Example, A retailers can identify the products that normally Customers purchase together or Even find the Customers who respond to the promotion of same kind of products.

* Clustering Analysis :- is used to identify data objects that are similar to one another.

→ The similarity can be decided based on a no of factors like purchase behaviour, responsiveness, geographic location & so on.

→ For Example, An Insurance Company can cluster its Customers based on age, residence, Income Etc.

* Types of Data :-
Data set can be often be viewed as a Collection of data objects. Other names for a data object are "record", "point", "vector", "pattern", "Event", "case", "Sample" or "observation".

Data can be classified into two broad types such as

* Qualitative
* Quantitative (numeric)

* Qualitative Data :- arise when the observations fall into separate distinct categories.

→ for Ex:- Colors of Eyes – blue, green brown
Exams result – Fail, pass

Such data are inherently "discrete", in that there are finite number of possible categories into which each observation may fall.

* Quantitative Data :- or numerical data arise when the observations are "Counts" or "Measurements".

→ The data are said to be "discrete" if the Measurements are integers (Ex No of people in a House) &
"continuous" if the Measurements can take on any value, usually within some range (Ex weight).

* Attribute & Measurement of Data :- An "Attribute" is a property or characteristic of an object that may vary, either from one object to another or from one time to another.

→ For Ex: Eye color varies from person to person, while the temperature of an object varies over time.

→ "Measurement Scale" is a rule (Function) that associates a numerical or symbolic value with an attribute of an object.

For Ex:- we count the no of chairs in a room to see if there will be enough to seat all the people coming to a Meeting.

* Different Types of Attributes :- A useful way to specify the type of an attribute is to identify the properties of numbers that correspond to underlying properties of the attribute.

→ Distinctness = & ≠ (Nominal) : The values of a nominal attribute are just different names
Nominal values provide only enough info to distinguish one object from another (=, ≠)
For Ex:- Zip codes, Employee ID

→ Order < ≤ > & ≥ (Ordinal) : The values of an ordinal attribute provide enough info to order objects
For Ex:- Grades, Street Numbers

→ Interval + & - (Addition) : For Interval attributes, the differences between values are meaningful. i.e a unit of measurement exists (+, -)
For Ex:- Calender dates.

→ Multiplication * & / (Ratio) : For ratio variables both differences & ratios are meaningful (*, /)
For Ex:- Counts, Age, Mass Etc.

* General Characteristics of Data Sets :- There are 3 characteristics that apply to many data sets & have a significant impact on the data Mining techniques.

* Dimensionality
* Sparsity
* Resolution

→ Dimensionality : refers to How many attributes a dataset has.
For Ex, Healthcare data is notorious for having vast amounts of variables (Blood pressure, Weight, Cholesterol level)
This data could be represented in Spreadsheet, with one column representing each dimension.

→ Sparsity : It is one in which a relatively high percentage of the Variables cells do not contain actual data.
For ex: a new variable dimensioned by MONTH for which you do not have data for part Months.

→ Resolution : It is frequently possible to obtain data at different levels of resolution, & often the properties of the data are different at different resolutions.
For ex: the Surface of Earth seems very uneven at a resolution of few Meters, but is relatively smooth at a resolution of tens of kilometers.

* Types of Data Sets :- There are Many types of data sets & as the field of Data Mining develops & Matures, a greater Variety of Data Sets become available for Analysis.

→ Record Data : is a collection of records (data objects), Each of which consists of a fixed set of data fields (Attributes).
There is no Explicit relationship among records or data fields. & Every record has the Same set of attributes.
→ Record data is usually Stored Either in flat files or in rel-tional databases.
→ Different types of Record data are
  * Transaction Data → where Each record (Transaction) involves a set of items.
  * Data Matrix → collection of Data with Same fixed set of Numeric attributes.
  * Document - Term Matrix → where Each term is a Comp-onent (Attribute) of the Vector.
→ Graph-Based Data : A graph can Sometimes be a Convenient & powerful Representation of Data.

→ There are two diff types of Graph-Based Data

* Data with Relationship among objects → Relationship among objects frequently convey important information.
  Ex: Web page in World Wide Web

* Data with objects that are Graphs → If objects contain subobjects that have relationships, then such objects frequently represented as Graphs.

→ Ordered Data: In one in which the attributes have relationship that involve order in time or space.

→ Different types of Ordered Data are

* Sequential Data → an Extension of Record data, where each record has a time associated with it.
  Ex: retail transaction data with time of transaction.

* Time Series Data → It a special type of Sequential Data in which each record is a time series.
  a series of Measurements taken over time.

* Spatial Data → Some objects have spatial attributes, such as positions or areas, as well as other types of attributes.
  Ex: Weather Data.

* Data Quality :-
Today's real-world databases are highly susceptible to noisy, Mining & Inconsistent Data due to their typically huge size & their likely origin from Multiple Heterogenous Source -s.

→ The Data Mining Applications focuses on

* the detection & Correction of Data quality problems
* Use of Algorithms that can tolerate poor Data Quality

"Detection" & "Correction" is often called "Data cleaning".

* Measurement & Data Collection Errors :- refers to any problem resulting from the measurement process.
A common problem is that the value recorded differs from the true value to some extent.

→ The term "data collection error" refers to errors such as omitting data objects or attribute values or inappropriately including data object.

* Noise and Artifacts :- Noise is the random component of a measurement error. It may involve the distortion of a value or addition of spurious objects.

fig: True Series

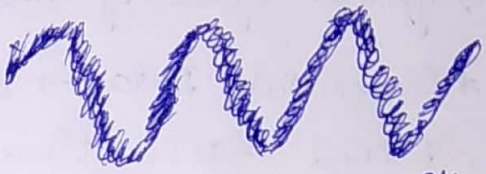


fig: True Series with Noise

→ Data errors may be the result of a more deterministic phenomenon. Such deterministic distortion of the data are often referred to as "Artifacts"
Ex: A streak in same place on a set of photographs.

* precision, Bias & Accuracy :- "precision" is a closeness of repeated measurements to one another.
"Bias" is a systematic variation of measurements from the quantity being measured.

→ precision is often measured by the standard deviation of a set of values, while bias is measured by taking the difference between the mean of the set of values & quantity being measured.

→ It is common to use general term, "Accuracy" to refer the degree of measurement error in data.
Accuracy depends on precision & Bias.

* Outliers :- An outlier is an observation point that is distant from other observations (Data).
An outlier may be due to variability in the measurement or it may indicate Experimental Error.
→ The latter are sometimes, outliers will be excluded from the data set.
An outlier can cause serious problems in Analysis.

* Missing Values :- Imagine you need to analyze an Enterprise data, you note that many tuples have no recorded value for several attributes in such case, fill the Missing values by fall Methods
1. Ignore the tuple → Usually done when class label is Missing, this Method is very ineffective.
2. fill the Missing value Manually → Its very time Consuming & not feasible.
3. Use a global Constant to fill Missing value → Replace all Missing values by some Constant such as "Unknown" or "∞" this Method is simple, Its not fool proof.
4. Use a Measure of Central tendancy for the attribute (Mean or Median) to fill Missing values.
5. Use the Most probable Values to fill the Missing values → this may be determined with regression, Bayesian or Decision tree.

* Inconsistent Values :- Data can contain inconsistent values.
Some types of inconsistent values are Easy to detect.
For Ex :- A person's Height should not be Negative
Once An inconsistency has been detected, it is sometimes possible

to Correct the data. The Correction of an Inconsistency requires additional or redundant Information.

* Duplicate Data :- A data Set May Include data objects that are duplicates, or almost duplicates of one another.

→ To detect & Eliminate Such duplicates, two Main Issues Must be addressed

* If there are two objects that actually represent a Single object with different values, & these Inconsistent values Must be resolved
* Care to be taken to avoid accidentally Combining data - objects that are Similar, but not duplicate

The "Deduplication" is often used to deal with duplicate data.

* Data preprocessing :-

Is a Data Mining technique that Involves transforming raw data Into an Understandable format.

Real-world data is often Incomplete, Inconsistent, and/or lacking In certain behaviors or trends, & is likely to Contain Many Errors.

Data preprocessing is a proven Method of resolving Such Issues.

→ Some of the Most Important Approaches for Data preprocessing are

* Aggregation
* Sampling
* Dimensionality reduction
* Feature Subset Selection
* Feature Creation
* Discretization & Binarization
* Variable Transformation

These Approaches fall Into two categories, Selecting data objects & Attributes for Analysis or Creating/changing the attributes.

* Aggregation :- Aggregation refers to combining two or more attributes (or objects) into a single attribute (or object).
For Ex:- Merging daily sales figures to obtain monthly sales figures.

→ Data Aggregation helps in Data reduction, Allows use of more Expensive Algorithms.
If done properly, Aggregation can act as Scope or Scale, providing a high level view of data instead of low level view.

→ The Aggregate quantities have "less variability" than the individual objects.

* Sampling :- Is the process of understanding characteristics of Data or Models Based on a subset of the original Data.
It is used Extensively in all aspects of Data Exploration & Data Mining.

→ Obtaining the "Data of Interest" on Entire Set is too Expensive & time Consuming
So we can use Sampling, where Entire Set of data may not be necessary.

→ a "Representative Sample" is a representative for a particular operation, if it results in approximately the same outcome as if the Entire data Set was used.

→ There are two probability of Selecting any particular Item
  * Sampling without replacement → Once an Item is selected it is removed from the population for obtaining Future Samples.
  * Sampling with replacement → Selected Item is not removed from the population for obtaining the Future Samples.

→ Even if proper Sampling technique is known, it is important to choose proper Sample Size.
  * larger Sample Size increase the probability that a Sample will be "representative".
  * Smaller Sample Size may result in Missed or Erroneous patterns Detected.

* Dimensionality Reduction:- "Curse of Dimensionality", the Data Analysis becomes Significantly Harder as the Dimensionality of the Data increases.

→ Dimensionality Reduction is a process of Determining the Dimensions that are important for Modeling.
Many of Data Mining Algorithms work better if the Dimensionality of Data is lower.

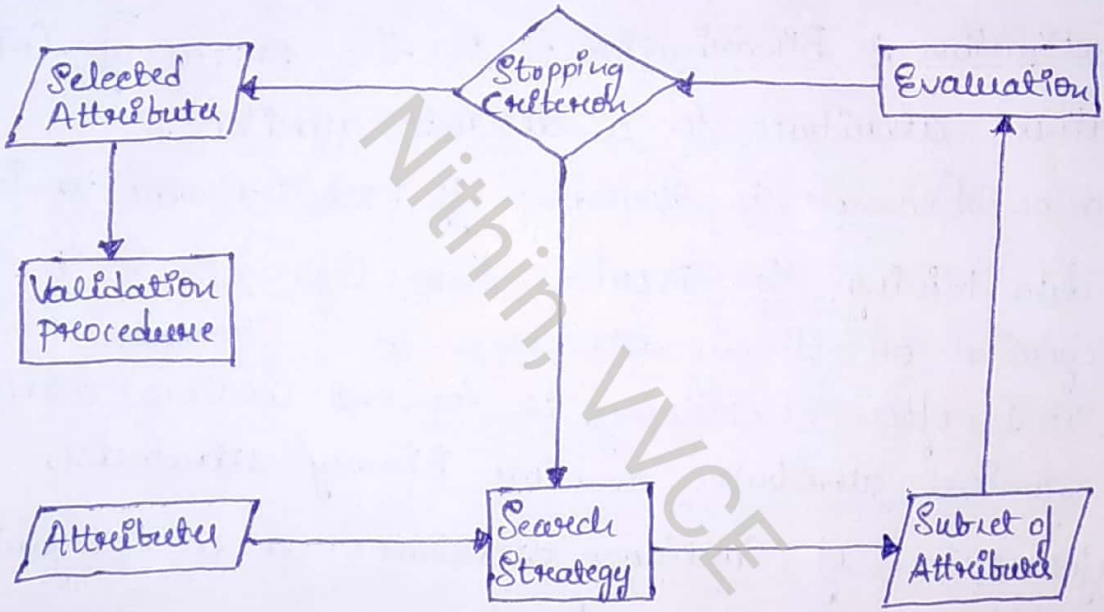
→ If Dimensionality Reduction Eliminates Irrelevant features or Reduces Noise, then Quality of results May Improve.
Can lead to More Understandable Model.

→ The Dimensionality Reduction can be done in two ways
  * Redundant features → Duplicate Much or all of the info contained in one or More attributes.
    Ex:- purchase price of product & Sales tax paid contain the Same info.
  * Irrelevant features → Contain no information that is useful for Data Mining task.
    Ex:- Student ID is Irrelevant to the task of predicting their Marks.

→ Dimensionality Reduction can be done using "principal Component Analysis (PCA)" is a linear Algebra technique for continous attributes that finds new Attributes.

* Feature Subset Selection :- There are three Standard Approaches to Feature Selection.
  * Embedded Approach → Feature Selection occurs naturally as part of the Data Mining Algorithm.
  * Filter Approaches → Features are Selected before the Data Mining Algorithm is run.
  * Wrapper Approaches → Use the target data Mining Algo as a black box to find the best Subset of attributes.

→ The Architecture for Feature Subset Selection is



The feature selection process is viewed as consisting of four parts such as, a Measure for Evaluating a Subset, a Search Strategy that controls the generation of new Subset of features, a Stopping Criterion & a Validation procedure.

* Feature Creation :- Sometimes, a Small number of new attributes can Capture the Important Information in a Data Set Much More Efficiently than the original attribute.

Such Attributes can be generated by three Methodologies
  * Feature Extraction → is a Creation of a new, Smaller Set of features from the original Set of features.
    Ex: In a Set of photographs, where Each photograph

is to be classified whether its human face or not.

* Mapping the Data to New Space → Sometimes a totally different View of the data can reveal important & interesting features.

  Ex:- Applying fourier Transform to data to detect time series patterns.

* Feature Construction → one or More new features Constructed out of the original features.

  Ex:- there are two attributes that record volume & Mass of a set of objects. i.e density = mass/volume.

* Discretization & Binarization :- is the process of converting a continous attribute to a discrete attribute.

A common Example is rounding off real numbers to integers.

→ Some data Mining Algo requires that the data be in the form of Categorical or binary attributes.

Thus, it is often necessary to convert continous attributes in to categorical attributes and/or Binary attributes.

→ Transformation of continous attributes to a categorical attributes / Binary attributes involves

  * Deciding How Many categories to have
  * How to Map the values of the continous Attributes to categorical Attribute.

* Attribute Transformation :- Refers to a transformation that is applied to all values of an attribute.

i.e for Each object, the transformation is applied to the value of the attribute for that object.

→ There are two important types of Attribute Transformation

  * Simple function Transformation → x^k, $\log x$, e^x, $\sqrt{x}$ etc
  * Standardization or Normalization

* Measures of Similarity & Dissimilarity (Similarity Measures)

Similarity & Dissimilarity are important because they are used by a number of Data Mining Techniques, such as Clustering, nearest neighbour Classification & anomaly Detection.

→ The term "proximity" is used to refer to either Similarity or Dissimilarity

* "Similarity" between two objects is a numerical Measure of the degree to which the two objects are alike. Consequently, Similarities are higher for pairs of objects that are More alike.
Similarities are usually non-negative & are often between "0" (no Similarity) & "1" (complete Similarity).

* "Dissimilarity" between two objects is a numerical Measure of the degree to which the two objects are different.
Dissimilarity is lower for More Similar pairs of objects.

→ The term "distance" is used as a Synonym for Dissimilarity. Dissimilarities sometimes fall in the interval [0, 1], but it is also common for them to range from "0" to "∞".

* proximity Measures :- Especially Similarities are defined to have values in the interval [0, 1]. If the Similarity between objects can range from "1" (Not Similar) to "10" (completely Similar).
We can Make them fall into the range [0, 1] by formula.

$$s' = (s-1)/9$$

where s & s' are original & the new Similarity values respectively.

→ The More general case s' is calculated as

$$s' = (s - min\_s) / (max\_s - min\_s)$$

Where min_s & max_s are the Minimum & Maximum Similarity values respectively.

→ Likewise, Dissimilarity Measures with a finite range can be Mapped to the Interval [0,1] using the formula

$$d' = (d - min\_d) / (max\_d - min\_d)$$

* Similarity & Dissimilarity between Simple attributes :- The proximity of objects with a number of attributes is defined by Combining the proximities of individual attributes.

Consider "p" & "q" are the attribute values for two data objects.

Attribute Type	Dissimilarity	Similarity
Nominal	$d = \begin{cases} 0 & \text{if } p = q \\ 1 & \text{if } p \neq q \end{cases}$	$s = \begin{cases} 1 & \text{if } p = q \\ 0 & \text{if } p \neq q \end{cases}$
Ordinal	$d = \frac{\lvert p - q \rvert}{n-1}$ (Value Mapped to Integer 0 to $n-1$)	$s = 1 - d$
Interval or Ratio	$d = \lvert p - q \rvert$	$s = -d$, $s = \frac{1}{1+d}$ or $s = e^{-d}$

* Distances :- Distances are dissimilarities with certain properties the "Euclidean Distance" d, between two points x & y in one, two or High dimensional Space is given by

$$d(x,y) = \sqrt{\sum_{k=1}^{n} (x_k - y_k)^2}$$

Where n is the number of Dimensions & x_k & y_k are respectively, the k^{th} attribute of x and y.

* Similarities between Data objects :- $S(x,y)$ is the Similarity between points x & y, then typically we will have
  * $S(x,y) = 1$ only if $x = y$ $(0 \le S \le 1)$
  * $S(x,y) = S(y,x)$ for all x & y (Symmetry)

* Similarity Measures for Binary Data :- Similarity Measures between objects that contain only Binary attributes are called "Similarity Co-Efficients".

→ "Simple Matching Co-Efficient (SMC)" one commonly used Similarity Co-efficient is defined as

$$SMC = \frac{\text{No of Matching attribute value}}{\text{No of attributes}} = \frac{f_{11}+f_{00}}{f_{11}+f_{00}+f_{01}+f_{10}}$$

this Measures counts both presence & absence Equally.

→ "Jaccard Co-Efficient" Measures only the presence of an Item is relevant

$$J = \frac{\text{No of Matching presences}}{\text{No of attributes not Involved in 00 Matches}} = \frac{f_{11}}{f_{11}+f_{10}+f_{01}}$$