Advanced Topics in Data Mining

Instructor:
Dr. Hamid Turab Mirza

Department of Computer Science
CUI, Lahore

DATA

Outline

- Introduction
- Attributes, Data Sets, and Data Storage
- Values, Features, and Objects
- Data Sets
- Data Storage: Databases and Data Warehouses
  - Data storage and data mining
- Issues Concerning the Amount and Quality of Data
  - High dimensionality
  - Dynamic data
  - Imprecise, Incomplete, and Redundant data
  - Missing values and Noise

Introduction

Data mining and KDP results depend on the quality and quantity of data

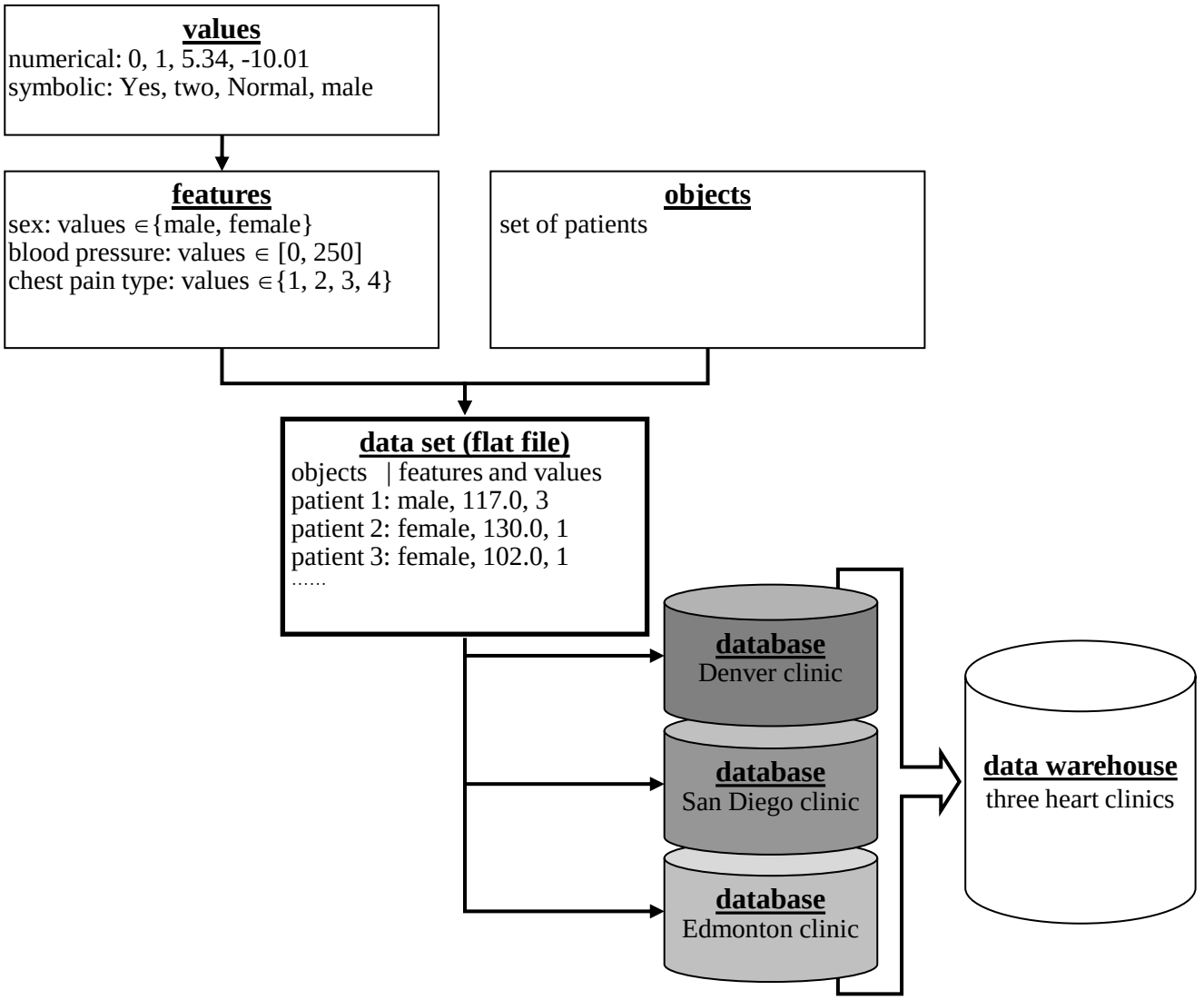
- **we focus on three issues: data types, data storage techniques, and amount and quality of the data**
- **the above constitutes necessary background for the KDP steps**

Attributes, Data Sets, and Data Storage

Data have diverse formats and are stored using a variety of different storage modes

- **a single unit of information is a value of a feature/attribute, where each feature can take on a number of different values**
- **objects, described by features, are combined to form data sets, which are stored as flat files or in other formats in databases and data warehouses**

Attributes, Data Sets, and Data Storage



Values, Features, and Objects

The key types of values are:

- **numerical values are expressed by numbers**
  - **for instance real numbers (-1.09, 123.5), integers (1, 44, 125), prime numbers (1, 3, 5), etc.**
- **symbolic values usually describe qualitative concepts**
  - **colors (white, red) or sizes (small, medium, big)**

Values, Features, and Objects

Features described by numerical and symbolic values can be either discrete (categorical) or continuous

- **Discrete features concern a situation in which the total number of values is relatively small (finite)**
  - **a special case of a discrete feature is binary feature, with only two distinct values**
- **Continuous features concern a situation in which the total number of values is very large (infinite) and covers a specific interval/range**
- **Nominal feature implies that there is no natural ordering between its values, while an ordinal feature implies some ordering**
- **values for a given feature can be organized as sets, vectors, or arrays**

Values, Features, and Objects

Objects (records, examples, units, cases, individuals, data points)

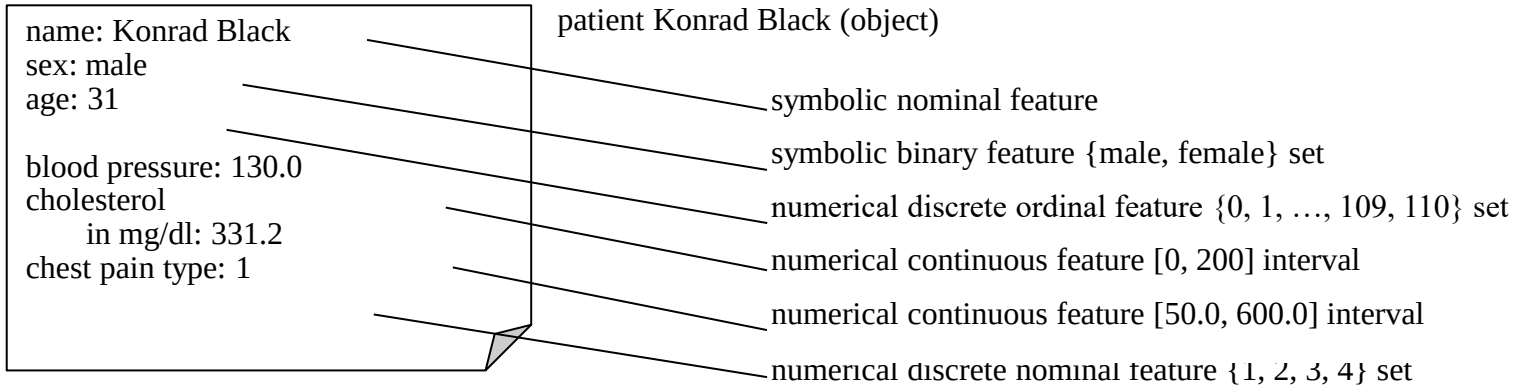
represent entities that are described by one or more features

- **multivariate data - where objects are described by many features**
- **univariate data – where objects are described by a single feature**

Values, Features, and Objects

Example: patients at a heart disease clinic

- **Object "patient" is described by name, sex, age, diagnostic test results such as blood pressure, cholesterol level, and diagnostic evaluation like chest pain type (severity)**



Data Sets

Objects described by the same features form data sets

- **many DM tools assume that data sets are organized as flat files, stored in a 2D array comprised of rows and columns, where**
  - **rows represent objects**
  - **columns represent features**
- **flat files store data in a text file format, and are often generated from spreadsheets or databases**

Data Sets

Example of patient data at a heart disease clinic

Name	Age	Sex	Blood pressure	Blood pressure test date	Cholesterol in mg/dl	Cholesterol test date	Chest pain type	Defect type	Diagnosis
Konrad Black	31	male	130.0	05/05/2005	NULL	NULL	NULL	NULL	NULL
Konrad Black	31	male	130.0	05/05/2005	331.2	05/21/2005	1	normal	absent
Magda Doe	26	female	115.0	01/03/2002	NULL	NULL	4	fixed	present
Magda Doe	26	female	115.0	01/03/2002	407.5	06/22/2005	NULL	NULL	NULL
Anna White	56	female	120.0	12/30/1999	45.0	12/30/1999	2	normal	absent
…	…	…	…	…	…	…	…	…	…

- **notice new feature type: date (numeric/symbolic)**
- **NULL value indicates that the corresponding feature value is unknown (not measured, or missing)**
- **several objects relate to the same patient (Konrad Black, Magda Doe)**
- **for Anna White, the cholesterol value is 45.0, which is outside of the interval defined as acceptable for this feature (so it is incorrect)**

Data Sets

There are three popular flat file data repositories:

- **http://www.ics.uci.edu/~mlearn/ (Machine Learning repository)**
- **http://kdd.ics.uci.edu/ (Knowledge Discovery in Databases archive)**
- **http://lib.stat.cmu.edu/ (StatLib repository)**

All provide free access to numerous data sets often used for benchmarking purposes

- **data are posted with results of their analysis**

Data Storage: Databases and Data Warehouses

DM tools can be used on a variety of other than flat files data formats such as

- **databases**
- **data warehouses**
- **advanced database systems**
  - **object-oriented and object-relational database**
  - **data-specific databases, such as transactional, spatial, temporal, text, or multimedia databases**
- **World Wide Web (WWW)**

Data Storage: Databases and Data Warehouses

Why to use database systems?

- **The flat file may not fit into the memory of a computer**
- **DM methods need to work on many different subsets of data and therefore a data management system is required to efficiently retrieve the required pieces of data**
- **Data may need to be dynamically added and updated, often by different people in different locations, and at different times**
- **Flat file may include redundant information, which can be avoided if data are stored in multiple tables**

Data Storage: Databases and Data Warehouses

Data Base Management System (DBMS)

- **consists of a database that stores the data and a set of programs for management and fast access to the data**

- **It provides services like**
  - **ability to define the structure (schema) of the database**
  - **ability to store the data, to access the data concurrently, and have data distributed/stored in different locations**
  - **ensures security (to protect against an unauthorized access or the system crash)**

Databases

The most common DB type is a relational database, which consists of a set of tables

- **each table is rectangular and can be perceived as a single flat file**
- **tables consist of attributes (columns, fields) and tuples (records, rows)**
- **each table has a unique name, and each record is assigned a special attribute (known as a key) that defines unique identifiers**
- **It includes the Entity-Relational Data (ER) model, which defines a set of entities (tables, records, etc.) and their relationships**

Databases

Relational database

– using multiple tables results in removal of redundant information present in the flat file

– the data are divided into smaller blocks that are easier to manipulate and that fit into memory

patient

Patient ID	Name	Age	Sex	Chest pain type	Defect type	Diagnosis
P1	Konrad Black	31	male	1	normal	absent
P2	Magda Doe	26	female	4	fixed	present
P3	Anna White	56	female	2	normal	absent
…	…	…	…	…	…	…

blood_pressure_test

Blood pressure test ID	Patient ID	Blood pressure	Blood pressure test date
BPT1	P1	130.0	05/05/2005
BPT2	P2	115.0	01/03/2002
BPT3	P3	120.0	12/30/1999
…	…	…	…

cholesterol_test

Cholesterol test ID	Patient ID	Cholesterol in mg/dl	Cholesterol test date
SCT1	P1	331.2	05/21/2005
SCT2	P2	407.5	06/22/2005
SCT3	P3	45.0	12/30/1999
…	…	…	…

performed_tests

Patient ID	Blood pressure test	Cholesterol test
P1	BPT1	SCT1
P2	BPT2	SCT2
P3	BPT3	SCT3
…	…	…

Databases

DBMS uses a specialized language: SQL (Structured Query Language)

- **SQL provides fast access to portions of the entire database**
  - **For example we may want to extract information about tests performed between specific dates**
    - **this is simple with SQL while with a flat file the user has to manipulate the data himself to extract the desired portion of data**

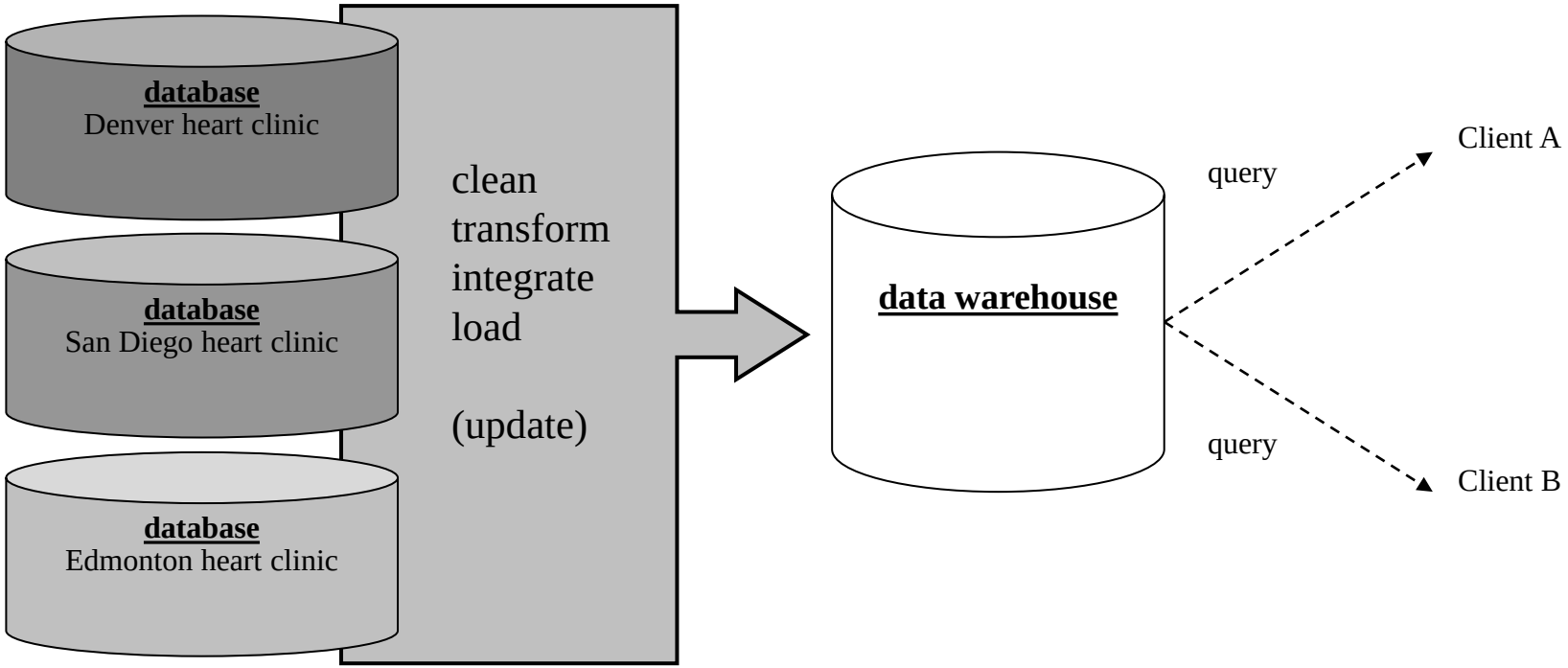
Data Warehouses

The main purpose of a DB is data storage, while the main purpose of a data warehouse is data analysis

- **data warehouse is organized as a set of subjects of interest**
  - **such as patients, test types, or diagnoses**
- **analysis is done to provide information from a historical perspective**
  - **for instance we ask for a breakdown of the most often performed clinical tests over the last five years**
  - **such requests/queries require availability of summarized information**
    - **For instance a DW may store the number of tests performed by each clinic, during a month, for a specific patient age intervals**

Data Warehouses

Data warehouse for the heart disease clinics



Data Warehouses

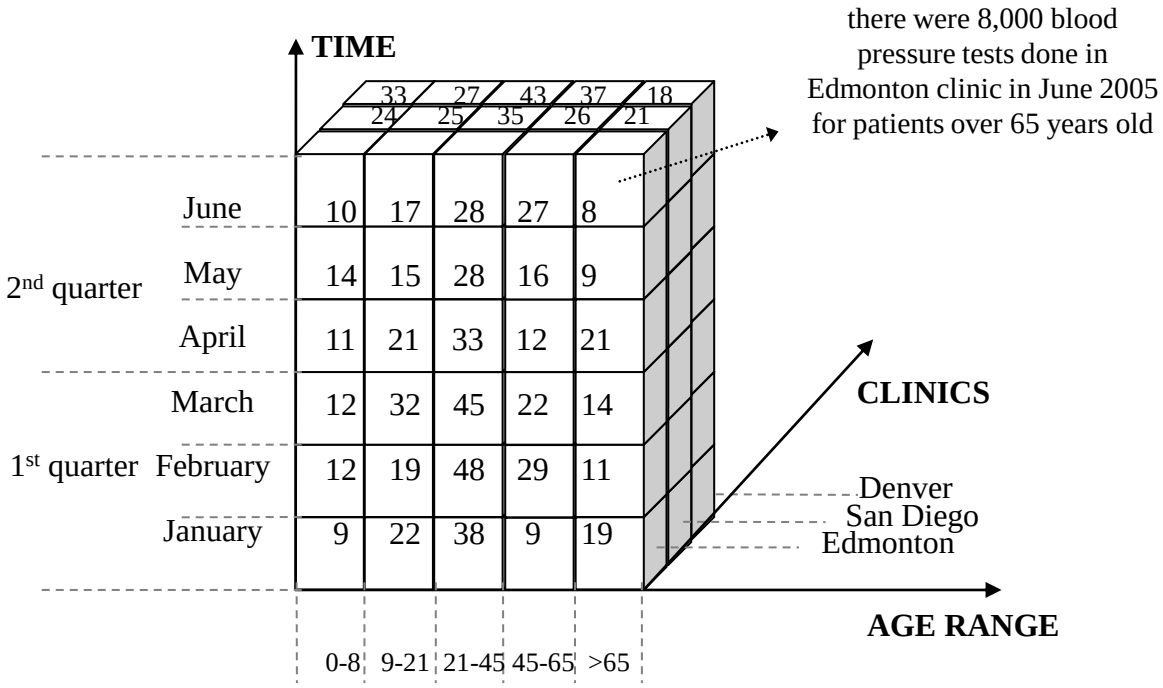
DW usually uses a multidimensional database structure

- **each dimension corresponds to an attribute or a set of attributes**
- **each cell (value) in the database corresponds to some summarized (aggregated) measure, such as average, etc.**
- **the implementation of a DW can be as a relational database or a multidimensional data cube**
  - **the cube is a 3D view of the data and allows for fast access**

Data Warehouses

Data cube for the heart disease clinics

- 3D: clinic (Denver, San Diego, Edmonton), time (in months), and age range (0-8, 9-21, 21-45, 45-65, over 65)
  - each dimension can be summarized, e.g., we can collapse months into quarters
- values are in thousands and show how many blood pressure tests were performed



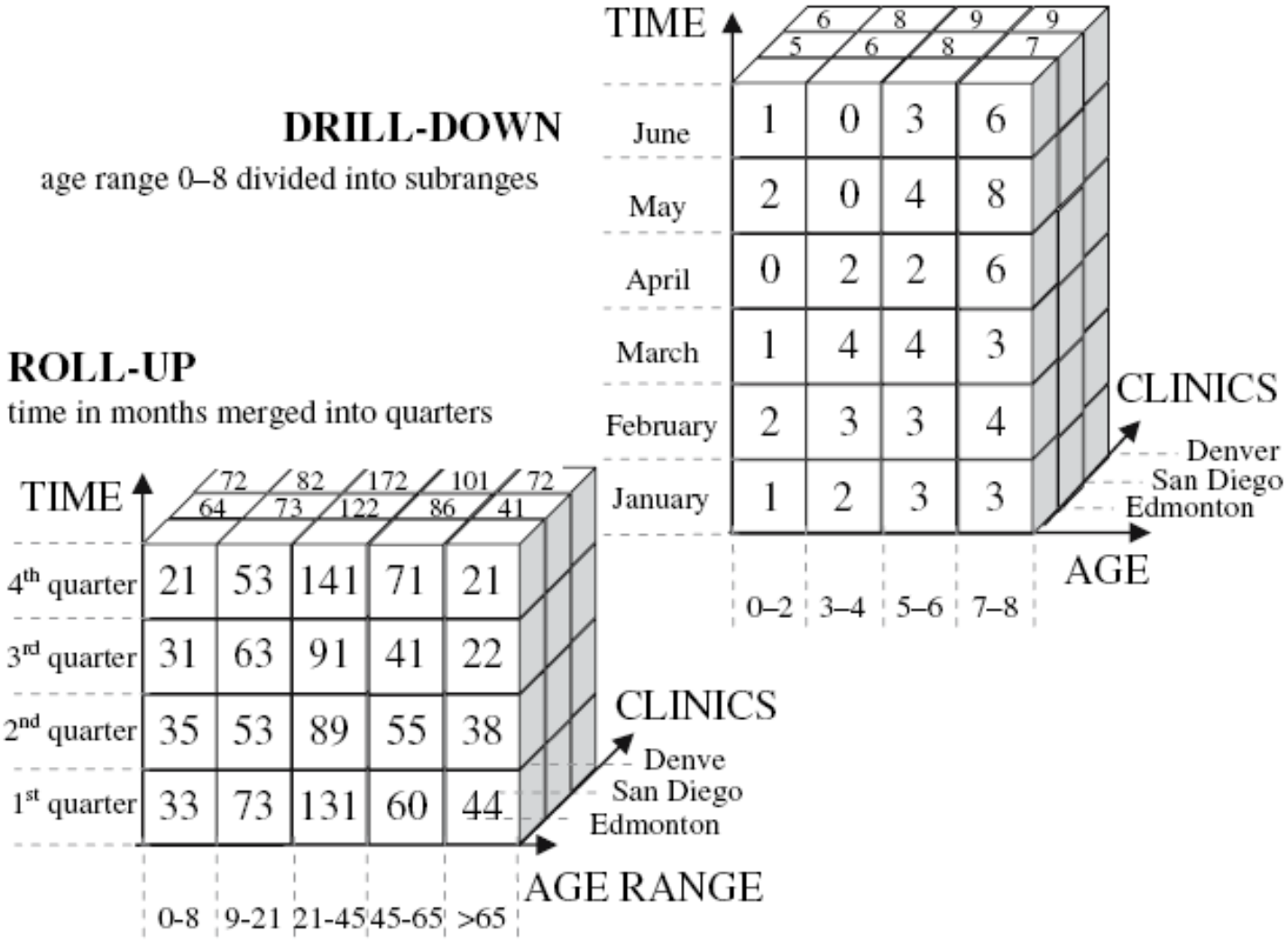


Figure 3.7. Results of the drill-down and roll-up operations for the data cube shown in Figure 3.6. The values are in thousands; for drill-down, we subdivide the age range dimension; for roll-up, we merge the time dimension. For readability, only some of the cube cells are shown.

Advanced Data Storage

Relational databases/warehouses are often used by retail stores/banks.

More advanced DBMS have emerged to satisfy users who need to handle more complex data; they store and manipulate data such as:

- **transactional**
- **spatial data**
- **hypertext**
- **multimedia data**
- **temporal data**
- **the WWW content**

They utilize efficient data structures and methods for handling operations on the more complex data.

Advanced Data Storage

Object-oriented databases are based on the object-oriented programming paradigm, which treats each stored entity as an object

- **object encapsulates a set of variables (its description), a set of messages that the object uses to communicate with other objects, and a set of methods that contain code that implement the messages**

- **similar (the same) objects are grouped into classes, and are organized in hierarchies**
  - **For instance, the patient class would have variables name, address, and sex, while its instances would be particular patients**
    - **patient class could have a subclass of "retired patient", which inherits all variables of the patient class but also have new variables such as date of the home release**

Advanced Data Storage

Transactional databases consist of records that represent transactions

- a transaction includes unique identifier and a set of items that make up the transaction
  - an example is a record of a purchase in a store

sales

Transation ID	Set of item IDs
TR000001	Item1, Item32, Item52, Item71
TR000002	Item2, Item3, Item4, Item57, Item 92, Item93
…	…

  - the difference between the relational and the transactional database is that the latter stores a set of items, rather than a set of values of the related features
    - » they are used to identify sets of items that frequently co-exist in transactions

Advanced Data Storage

Spatial databases handle spatially-related data, such as geographical maps, satellite images, or medical images

- **spatial data can be represented in two ways**
  - **in a raster format**
  - **in a vector format**

Raster format uses n dimensional pixel maps, while the vector format requires representing all objects as simple geometrical objects, such as lines, and uses vectors to compute relations between the objects

Advanced Data Storage

Temporal (time-series) databases extend the relational database to handle time-related features

- **attributes may be defined using timestamps like days and months, or hours and minutes, etc.**
- **the database keeps time-related features by storing sequences of their values that change with time**
  - **a relational database usually stores the most recent values only**

Advanced Data Storage

Text databases include features (attributes) that use word descriptions of objects

- **sentences or paragraphs of text**
  - **unstructured like sentences written in plain language (English, Polish, Spanish, etc.)**
  - **semistructured where some words or parts of the sentence are annotated (like drug's name and dose)**
  - **structured where all the words are annotated (physician's diagnosis may use a fixed form that lists specific drugs and doses)**
- **they require special tools and integration with text data hierarchies, such as dictionaries, thesauruses, and specialized term-classification systems (such as those used in medicine)**

Advanced Data Storage

Multimedia databases allow storage, retrieval and manipulation of image, video, and audio data

- **the main concern for such data sources is their very large size**
  - **both video and audio data are recorded in real-time, and thus the database must include mechanisms that assure a steady and predefined rate of acquisition to avoid gaps, system buffer overflows, etc.**

Advanced Data Storage

WWW is an enormous distributed repository of data linked together via hyperlinks

- **hyperlinks link individual data objects of different types together allowing for an interactive access**
- **most specific characteristic of the Web is that the users seek information by traversing between objects via links**
- **WWW uses specialized query engines such as Google and Yahoo!**

Advanced Data Storage

Heterogeneous databases consist of a set of interconnected databases that communicate between themselves to exchange the data and provide answers to user queries

- **the biggest challenge is that the objects in the component databases differ substantially, which results in difficulties in developing common semantics facilitating communication between them**

Data Storage and Data Mining

Data Mining vs. Utilization of a Data Storage

- DW and DB users often understand data mining as an execution of a set of the OLAP commands
  - data or information retrieval should not be confused with data mining
- data mining provides more complex techniques for understanding data and generating new knowledge
  - it allows for semi-automated discovery of patterns and trends
    - for instance, learning that increased blood pressure over a period of time leads to certain heart defects

Issues of the Amount and Quality of Data

Several issues related to the data have significant impact on the quality of the outcome of a KDP:

- **huge and highly dimensional volume of data, and the problem of DM methods scalability**
- **dynamic nature of the data – data are constantly being updated/changed**
- **problems related to data quality, such as imprecision, incompleteness, noise, missing values, and redundancy**

High Dimensionality

Among many DM tools available only some are "truly" being able to mine high-dimensional data

- **to handle massive amount of data that requires the algorithms to be scalable**

- **scalability is not related to efficient storage and retrieval (these belong to DBMS) but to the algorithm design**

  - **systems that are not capable of handling large quantities of data are known as machine learning or statistical data analysis systems**

High Dimensionality

Is it a real problem?

- **analysis of data from large retail stores goes into hundreds of millions of objects, while in bioinformatics data are described by thousands of features (e.g., microarray data)**

- **large commercial databases now average about one PB of objects**

High Dimensionality

Three "dimensions" of high-dimensionality need to be addressed

- **The number of objects, which may range from a few hundred to a few billion**
- **The number of features, which may range from a few to several thousand**
- **The number of values a feature assumes, which may range from 1-2 to a few million**

High Dimensionality

The ability of a particular DM algorithm to cope with highly dimensional data is described by its asymptotic complexity

- **it estimates the total number of operations, which translates into the specific amount of run time**
- **it describes the growth rate of the algorithm's run time as the size of each dimension increases**
- **the most commonly used complexity analysis describes scalability with respect to the number of objects**

High Dimensionality

To illustrate:

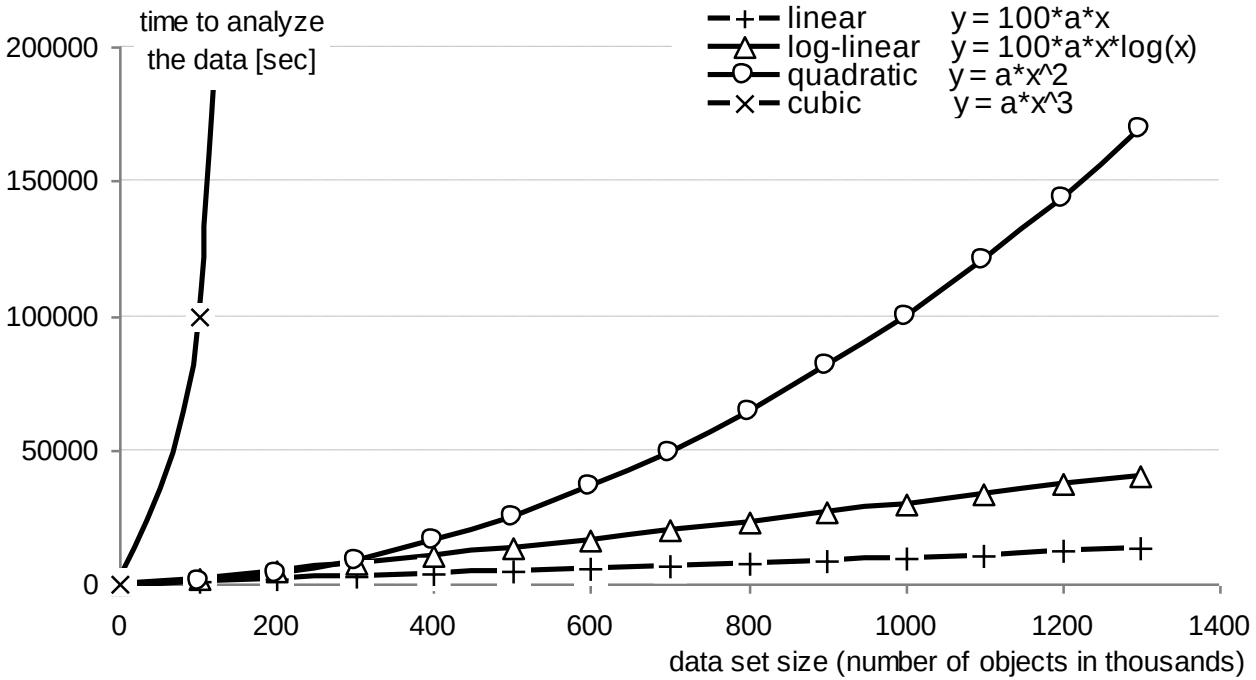
- **Assume the user wants to generate knowledge in terms of production rules using either a decision tree or a rule algorithm:**
  - **Proprietary See5 alg. has log-linear complexity, i.e., O(n*log(n)), n is the number of objects**
  - **DataSqueezer alg. has log-linear complexity, i.e., O(n*log(n))**
  - **CLIP4 alg. has quadratic complexity, i.e., O(n^2)**
  - **C4.5 rules alg. has cubic complexity, i.e. ,O(n^3)**

Table 3.2. Comparison of running time with respect to the "number of objects" for four rule-learning algorithms.

Number of objects in thousands	Running time [sec]			
	DataSqueezer $y = 100^*a^*x$	See5 $y = 100^*a^*x^*\log(x)$	CLIP4 $y = a^*x^2$	C4.5 rules $y = a^*x^3$
100	1000.0	2000.0	1000.0	100000.0
200	2000.0	4602.1	4000.0	800000.0
300	3000.0	7431.4	9000.0	2700000.0
400	4000.0	10408.2	16000.0	6400000.0
500	5000.0	13494.9	25000.0	12500000.0
600	6000.0	16668.9	36000.0	21600000.0
700	7000.0	19915.7	49000.0	34300000.0
800	8000.0	23224.7	64000.0	51200000.0
900	9000.0	26588.2	81000.0	72900000.0
1000	10000.0	30000.0	100000.0	100000000.0
1100	11000.0	33455.3	121000.0	133100000.0
...	...	...	...	...

High Dimensionality

This example illustrates the importance of asymptotic complexity



- **for 100 objects the linear algorithm computes the results in 1,000 seconds, while the cubic algorithm in 100,000 seconds (using previous formulas)**
- **when the number of objects increases 10x (to 1000) the time to compute the results for the linear algorithm increases to 10,000 seconds, and to 100,000,000 seconds for the cubic algorithm**

High Dimensionality

Two techniques can be used to improve scalability:

- **speeding up the algorithm**
  - **achieved through the use of heuristics, optimization, and parallelization**
    - **Heuristics – for instance: generate only rules of a certain maximal length**
    - **Optimization: use efficient data structures such as bit vectors, hash tables, or binary search trees to store and manipulate the data**
    - **Parallelization: distribute processing of the data into several processors**

High Dimensionality

Two techniques can be used to improve scalability:

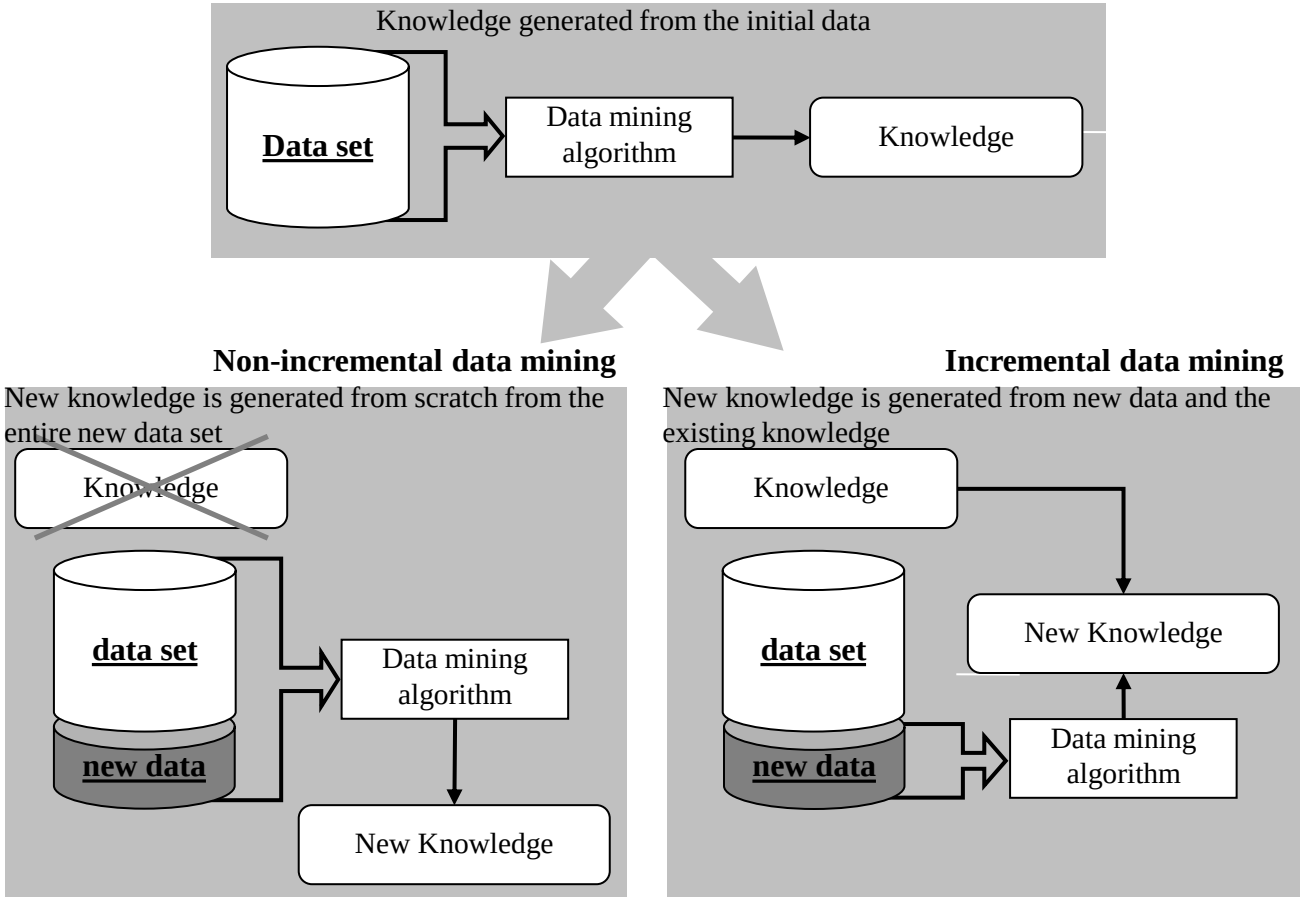
- **partitioning of the data set**
  - **reduce dimensionality of the data and use sequential/parallel processing of subsets of data**
    - **in dimensionality reduction the data is sampled and only a representative subset of objects and/or features is used**
      - **Sometimes it is necessary to reduce the number of values of an attribute by discretization**
    - **division of data into subsets is used when a DM algorithm's complexity is worse than linear**

Dynamic Data

Data are often dynamic in nature

- new objects and/or features may be added, and some objects and/or features may be removed or replaced by new ones

- DM algorithms should evolve with time
  i.e.
  the knowledge derived so far should be incrementally updated



Imprecise Data

Real data often include *imprecise* data objects

- **for instance, we may not know the exact value of a given test, but know *whether the value is high, average, or low***
- **in such cases fuzzy and/or rough sets can be used to process such information: INFORMATION GRANULARITY**

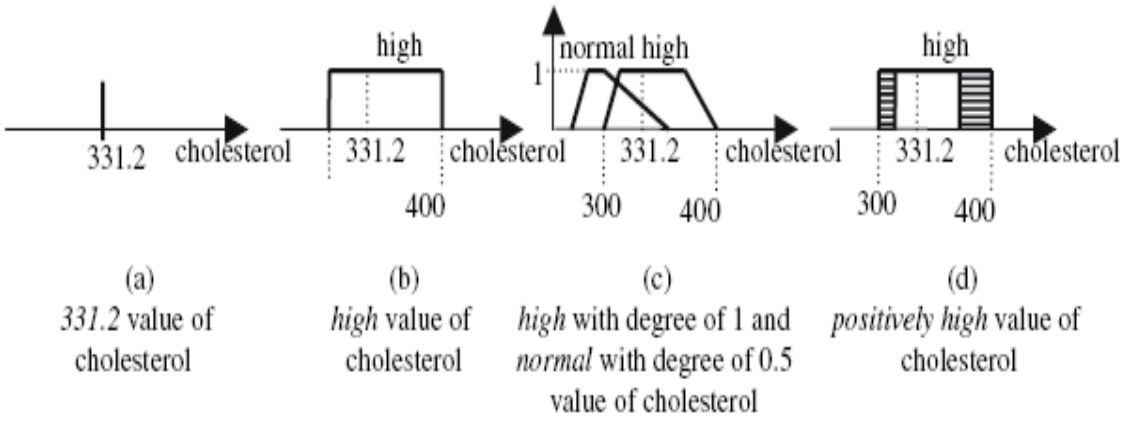


Figure 3.3. Information granularization methods. (a) Numerical; (b) interval based; (c) fuzzy set based; (d) rough set based.

Incomplete Data

Def.: Data that does not contain enough information to discover (potentially) new knowledge

- **for instance, when analyzing heart patient's data if one wanted to distinguish between sick and healthy patients**

  but only demographic information was given it would be impossible to do it

Incomplete Data

In case of incomplete data we first need to identify the problem and take some corrective measures

- **to detect incompleteness the user must analyze the existing data and assess whether the features and objects give sufficiently rich representation of the problem**
  - **if not we collect additional data (new features and/or objects)**

Redundant Data

Def.: Data containing two or more identical objects, or when features are strongly correlated

- **redundant data is often removed, but in some cases the redundant data contains useful information, i.e., frequency of the same objects provides useful information about the domain**
- **a special case is the irrelevant data, where some objects and/or features are insignificant with respect to data analysis**
  - **e.g., we can expect that patient's name is irrelevant with respect to heart condition**

- ***redundant data are identified by feature selection and extraction algorithms and removed***

Missing Values

Many data sets are plagued by the problem of missing values

- **missing values can be a result of manual data entry, incorrect measurements, equipment errors, etc.**
- **they are usually denoted by special characters such as:**

  NULL

  *****

  ?

Missing Values

Are handled in two ways:

- **removal of missing data**

  - **the objects and/or features with missing values are discarded: can be done only when the removed objects features are not crucial for analysis (e.g., in case of a huge data set)**
  - **practical only when the data contain small amount of missing values**
    - **for example, when distinguishing between sick and healthy heart patients, removing blood pressure feature will result in biasing the discovered knowledge towards other features, although it is known that blood pressure is important**

Original data set with missing values

Object ID	Name	Age	Sex	Blood pressure	Cholesterol in mg/dl	Chest pain type	Defect type	Diagnosis
1	Konrad Black	31	male	130.0	NULL	NULL	NULL	NULL
2	Konrad Black	31	male	130.0	331.2	1	normal	absent
3	Magda Doe	26	female	115.0	NULL	4	fixed	present
4	Magda Doe	26	female	115.0	407.5	NULL	NULL	NULL
5	Anna White	56	female	120.0	45.0	2	normal	absent

Data set with removed objects with missing values

Object ID	Name	Age	Sex	Blood pressure	Cholesterol in mg/dl	Chest pain type	Defect type	Diagnosis
2	Konrad Black	31	male	130.0	331.2	1	normal	absent
5	Anna White	56	female	120.0	45.0	2	normal	absent

Missing Values

Are handled in two ways:

- **imputation (filling-in) of missing data**
  - **performed using**
    - **single imputation methods, where a missing value is imputed by a single value**
    - **multiple imputations methods, where several likelihood-ordered choices for imputing the missing value are computed and one "best" value is selected**

Missing Values

- **single imputation**
  - **mean imputation method uses the mean of values of a feature that contains missing data**
    - **in case of a symbolic/categorical feature, *a mode* (the most frequent value) is used**
    - **the algorithm imputes missing values for each attribute separately**
    - **it can be conditional or unconditional**
      - **the conditional mean method imputes a mean value that depends on the values of the complete features for the incomplete object**

Data set with removed features with missing values

Object ID	Name	Age	Sex	Blood pressure
1	Konrad Black	31	male	130.0
2	Konrad Black	31	male	130.0
3	Magda Doe	26	female	115.0
4	Magda Doe	26	female	115.0
5	Anna White	56	female	120.0

Data set with missing values imputed using the mean imputation method

Object ID	Name	Age	Sex	Blood pressure	Cholesterol in mg/dl	Chest pain type	Defect type	Diagnosis
1	Konrad Black	31	male	130.0	261.2	2	normal	absent
2	Konrad Black	31	male	130.0	331.2	1	normal	absent
3	Magda Doe	26	female	115.0	261.2	4	fixed	present
4	Magda Doe	26	female	115.0	407.5	2	normal	absent
5	Anna White	56	female	120.0	45.0	2	normal	absent

The imputed values for the "cholesterol" feature equal (331.2+407.5+45)/3.

The imputed values for the "chest pain type" equal (1+4+2)/3 rounded to the nearest integer.

The imputed values for the "defect type" equal the most frequent value "normal".

Missing Values

- **single imputation**
  - **hot deck imputation: for each object that contains missing values the most similar object (according to some distance function) is found, and the missing values are imputed from that object**
    - **if the most similar record also contains missing values for the same feature then it is discarded and another closest object is found**
      - **the procedure is repeated until all the missing values are imputed**
      - **when no similar object is found, the closest object with the minimum number of missing values is chosen to impute the missing values**

Data set with missing values imputed using the hot deck imputation

Object ID	Name	Age	Sex	Blood pressure	Cholesterol in mg/dl	Chest pain type	Defect type	Diagnosis
1	Konrad Black	31	male	130.0	331.2	1	normal	absent
2	Konrad Black	31	male	130.0	331.2	1	normal	absent
3	Magda Doe	26	female	115.0	407.5	4	fixed	present
4	Magda Doe	26	female	115.0	407.5	4	fixed	present
5	Anna White	56	female	120.0	45.0	2	normal	absent

The distance between objects 1 and 2 (object ID is not considered as a feature)

equals 0+0+0+0+1+1+1+1=4.

The distance between objects 1 and 3 equals 1+1+1+1+1+1+1+1=8.

The object most similar to object 3 that has a complete value for "cholesterol" is object 4 (distance 4).

Figure 3.11. Results of using missing-data handling methods for a data set describing heart clinic patients.

Noise

Def.: Noise in the data is defined as a value that is a random error or variance in a measured feature

- **the amount of noise in the data can jeopardize the entire KDP results**
- **the influence of noise on the data can be prevented by imposing constraints on features to detect anomalies when the data is entered**
  - **for instance, DBMS usually provides facility to define constrains for individual attributes**

Noise

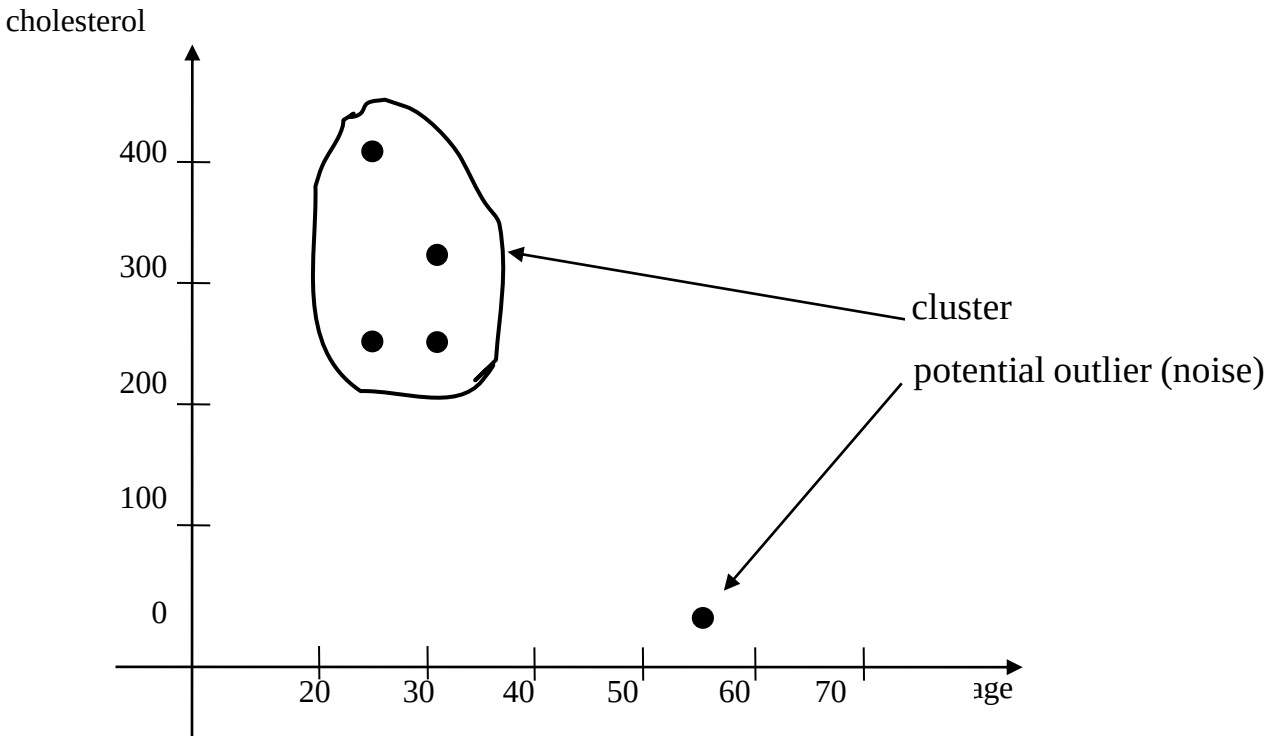
Noise can be removed using

- **Manual inspection with the use of predefined constrains on feature values**
  - **and manual removal (or change into missing values) of all values that do not satisfy them**
- **Binning**
  - **Requires ordering values of the noisy feature and then substituting the values with a mean or median value for predefined bins**
- **Clustering**
  - **it finds groups of similar objects and simply removes (or changes into missing values) all values that fall outside clusters**

Noise

Noise can be removed using

- **clustering**



References

Holsheimer, M. and Siebes, A. 1994. *Data Mining: The Search for Knowledge in Databases*, Report CS-R9406, ISSN 0169-118X, CWI: Dutch National Research Center, Amersterdam, Netherlands

Ganti, V., Gehrke, J. and Ramakrishnan, R. Aug. 1999. Mining Very Large Databases, *IEEE Computer*, 32(8):38-45

Klosgen, W. and Zytkow, J. (Eds) 2002. *Handbook of Data Mining and Knowledge Discovery*, Oxford University Press

Shafer, J.L. 1997. *Analysis of Incomplete Multivariate Data*, Chapman and Hall