Data Mining & Warehousing

Chapter Two

Data Preprocessing

Data Preprocessing

- Why preprocess the data?
- Data cleaning
- Data integration
- Data reduction
- Data transformation Discretization

  Summary

Why Data Preprocessing?

- **Data in the real world is dirty**
  - ✓ **incomplete**: lacking *attribute values*, lacking certain *attributes of interest*, or containing only aggregate data
  - ✓ **noisy:** containing errors or outliers
  - ✓ **inconsistent:** containing discrepancies in codes or names
- **No quality data, no quality mining results!**
  - ✓ Quality decisions must be based on quality data
  - ✓ Data warehouse needs consistent integration of quality data
  - ✓ Data quality required for both OLAP and Data Mining!

Why can Data be Incomplete?

- Attributes of interest are not available (e.g., customer information for sales transaction data)
- Data were not considered important at the time of transactions, so they were not recorded!
- Data not recorder because of misunderstanding or malfunctions
- Data may have been recorded and later deleted!
- Missing/unknown values for some data

Why can Data be Noisy/Inconsistent?

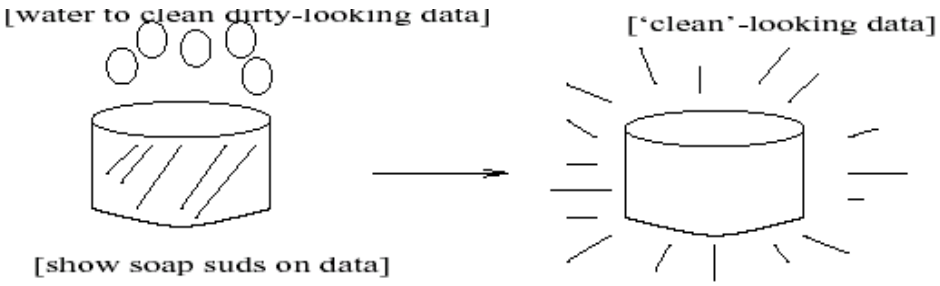
- Faulty instruments for data collection
- Human or computer errors
- Errors in data transmission
- Technology limitations (e.g., sensor data come at a faster rate than they can be processed)
- Inconsistencies in naming conventions or data codes (e.g., 2/5/2002 could be 2 May 2002 or 5 Feb 2002)
- Duplicate tuples, which were received twice should also be removed

Major Tasks in Data Preprocessing

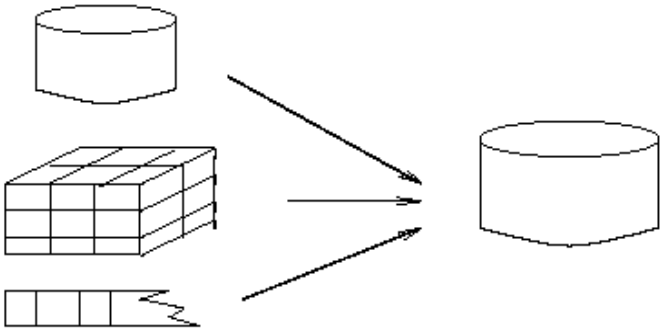
- Data cleaning
  - Fill in missing values, smooth noisy data, identify or remove outliers (exceptions), and resolve inconsistencies
- Data integration
  - Integration of multiple databases, data cubes, or files
- Data transformation
  - Normalization and aggregation
- Data reduction
  - Obtains reduced representation in volume but produces the same or similar analytical results
- Data discretization
  - Part of data reduction but with particular importance, especially for numerical data

Forms of data preprocessing

Data Cleaning



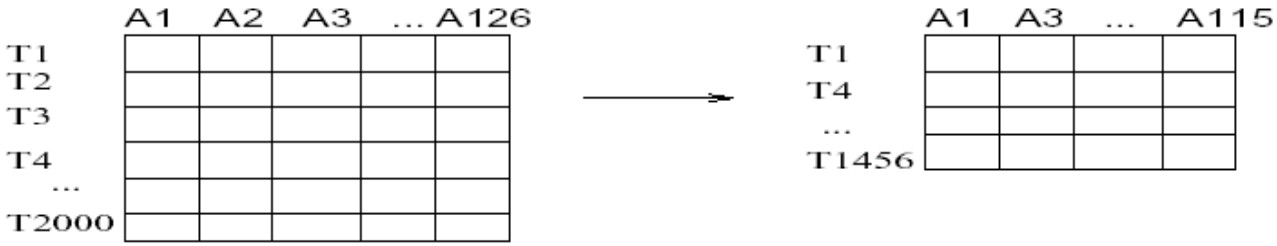
Data Integration



Data Transformation

-2, 32, 100, 59, 48 → -0.02, 0.32, 1.00, 0.59, 0.48

Data Reduction



Data Preprocessing

- Why preprocess the data?
- Data cleaning
- Data integration
- Data reduction
- Data transformation Discretization

  Summary

Data Cleaning

- Data cleaning tasks
  - Fill in missing values
  - Identify outliers and smooth out noisy data
  - Correct inconsistent data

How to Handle Missing Data?

- Ignore the tuple: usually done when class label is missing (assuming the tasks in classification) - not effective when the percentage of missing values per attribute varies considerably.
- Fill in the missing value manually: tedious + infeasible?
- Use a global constant to fill in the missing value: e.g., "unknown", a new class?!
- Use the attribute mean to fill in the missing value
- Use the attribute mean for all samples belonging to the same class to fill in the missing value: smarter
- Use the most probable value to fill in the missing value: inference-based such as Bayesian formula or decision tree

How to Handle Missing Data?

Age	Income	Religion	Gender
23	24,200	Muslim	M
39	?	Christian	F
45	45,390	?	F

Fill missing values using aggregate functions (e.g., average) or probabilistic estimates on global value distribution

E.g., put the most frequent religion here

E.g., put the average income here, or put the most probable income based on the fact that the person is 39 years old

Noisy Data

- Noise: random error or variance in a measured variable
- Incorrect attribute values may exist due to
  - faulty data collection instruments
  - data entry problems
  - data transmission problems
  - technology limitation
  - inconsistency in naming convention
- Other data problems which requires data cleaning
  - duplicate records
  - incomplete data
  - inconsistent data

How to Handle Noisy Data? Smoothing techniques

- Binning method:
  - first sort data and partition into (equi-depth) bins
  - then one can smooth by bin means, smooth by bin median, smooth by bin boundaries, etc.
- Clustering
  - detect and remove outliers
- Combined computer and human inspection
  - computer detects suspicious values, which are then checked by humans
- Regression
  - smooth by fitting the data into regression functions
- Use Concept hierarchies
  - use concept hierarchies, e.g., price value -> "expensive"

Simple Discretization Methods: Binning

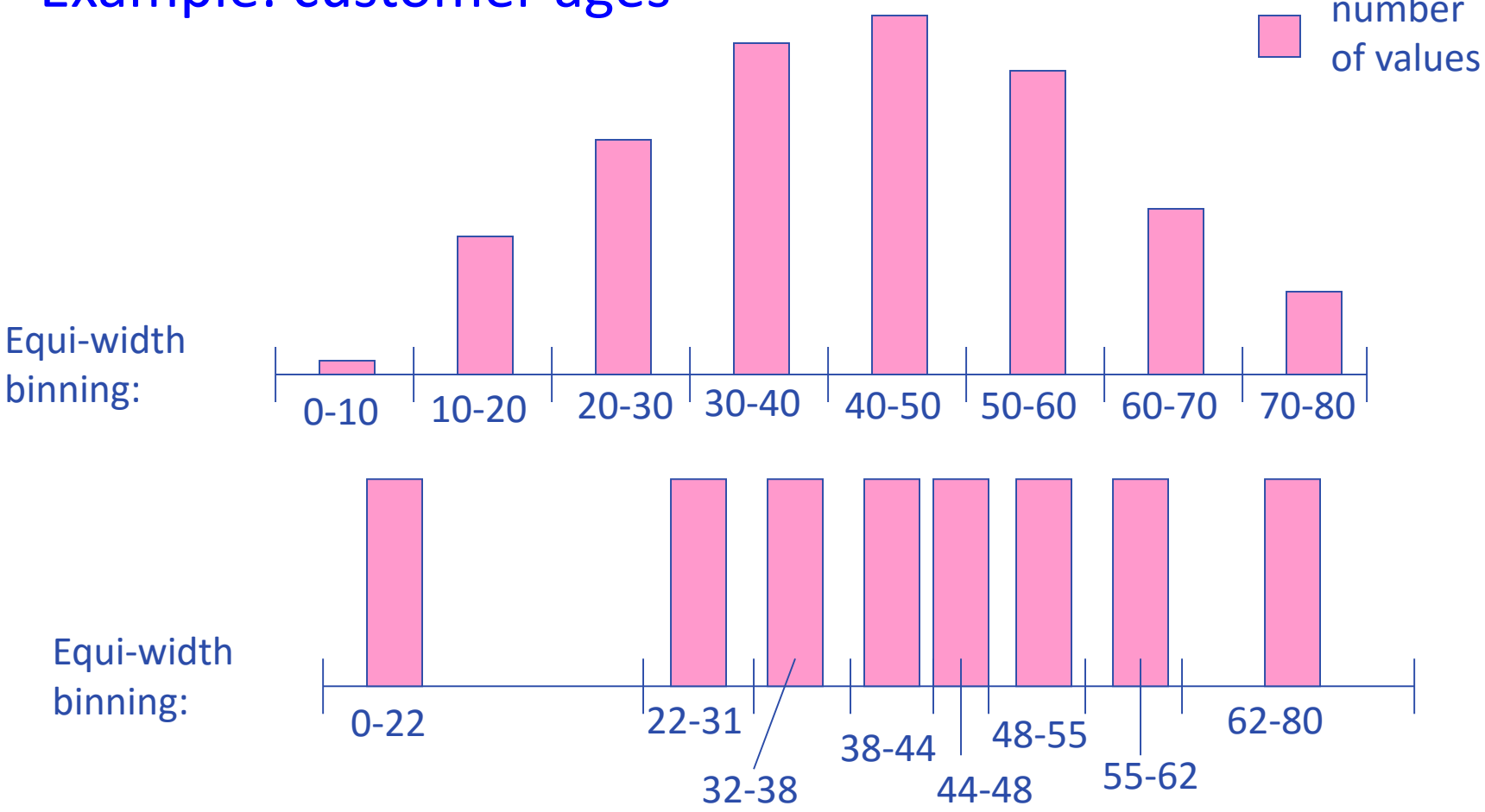
- Binning methods smooth a sorted data value by consulting its "neighborhood," that is, the values around it.
- The sorted values are distributed into a number of "buckets," or *bins*.
- **Equal-depth (frequency) partitioning**: it divides the range into *N* intervals, each containing approximately same number of samples.
- **Smoothing by bin means**, each value in a bin is replaced by the mean value of the bin.
- **Smoothing by bin medians** can be employed, in which each bin value is replaced by the bin median.
- **Smoothing by bin boundaries**, the minimum and maximum values in a given bin are identified as the *bin boundaries*. Each bin value is then replaced by the closest boundary value.

Smoothing using Binning Methods

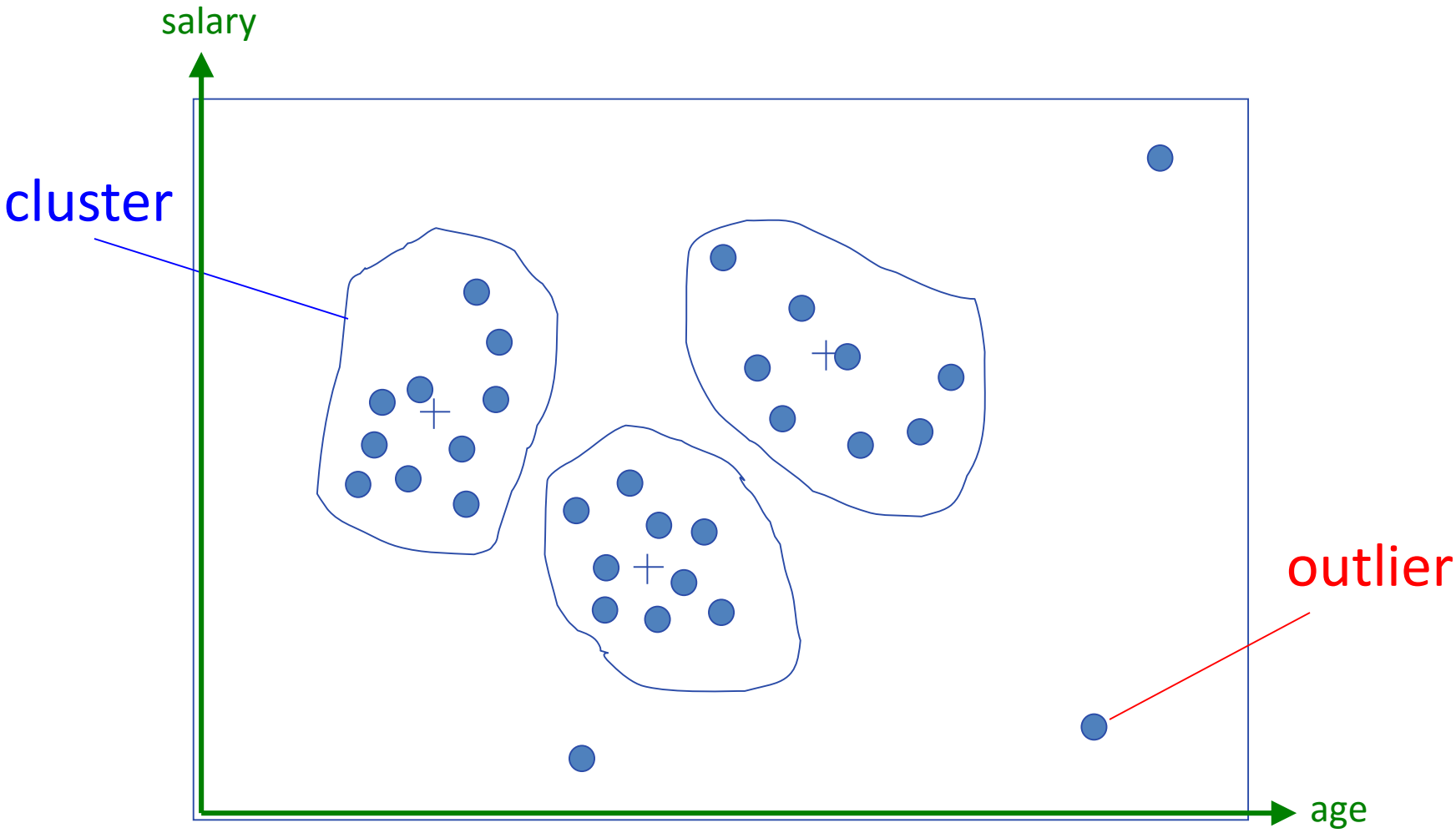
* Sorted data for price (in dollars): 4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34
* Partition into (equi-depth) bins:
  - Bin 1: 4, 8, 9, 15
  - Bin 2: 21, 21, 24, 25
  - Bin 3: 26, 28, 29, 34
* Smoothing by bin means:
  - Bin 1: 9, 9, 9, 9
  - Bin 2: 23, 23, 23, 23
  - Bin 3: 29, 29, 29, 29
* Smoothing by bin boundaries: [4,15],[21,25],[26,34]
  - Bin 1: 4, 4, 4, 15
  - Bin 2: 21, 21, 25, 25
  - Bin 3: 26, 26, 26, 34

Simple Discretization Methods: Binning

Example: customer ages

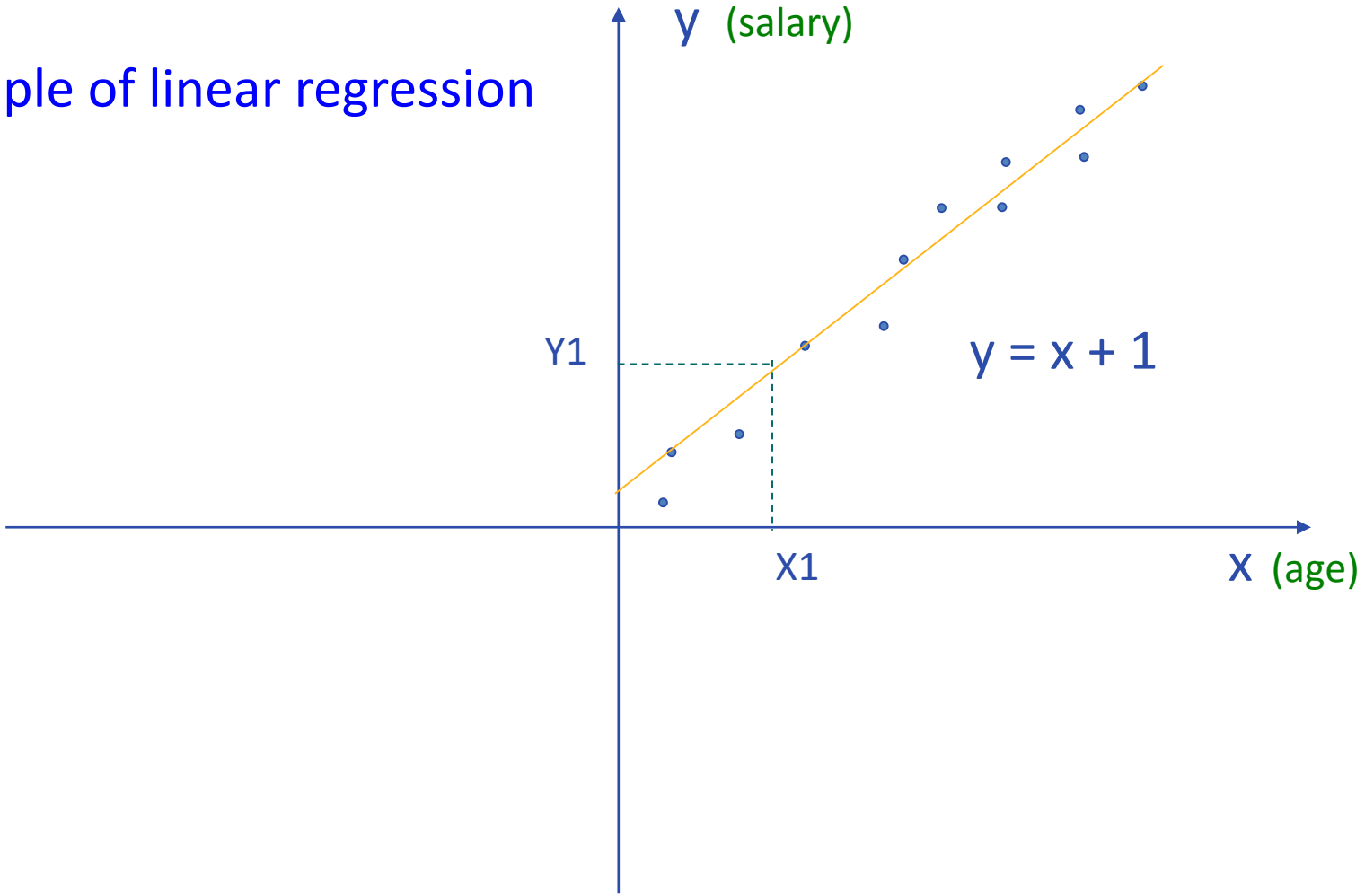


Cluster Analysis



Regression

Example of linear regression



Inconsistent Data

- Inconsistent data are handled by:
  - Manual correction (expensive and tedious)
  - Use routines designed to detect inconsistencies and manually correct them. E.g., the routine may use the check global constraints (age>10) or functional dependencies
  - Other inconsistencies (e.g., between names of the same attribute) can be corrected during the data integration process

Data Preprocessing

- Why preprocess the data?
- Data cleaning
- Data integration
- Data reduction
- Data transformation Discretization

Summary

Data Integration

- ❖Data integration: combines data from multiple sources into a coherent store.
- Careful integration can help reduce and avoid redundancies and inconsistencies in the resulting data set.
- This can help improve the accuracy and speed of the subsequent data mining process.
- The semantic heterogeneity and structure of data pose great challenges in data integration.
- How can we match schema and objects from different sources? Are any attributes correlated?
- Detection and resolution of data value conflicts.

Data Integration

❖ **Attribute naming (in schema integration)**

- Problem: Entity identification problem: identify real world entities from multiple data sources.
- Attributes are named differently across different data sources, e.g., A.cust-id ≡ B.cust-number (integrate metadata from different sources).

❖ **Data Encoding**

- Problem: Same attribute has same values denoted in different ways. for example ***Gender*** attribute value denoted as Male and Female in one system and M and F in another.

❖ **Measurement Basis (data value conflicts)**

- Problem: for the same real world entity, attribute values from different sources are different possible reasons: different representations, different scales.
- e.g., metric vs. British units, kg vs kg

Handling Redundant Data in Data Integration

- Redundant data occur often when integration of multiple databases
  - The same attribute may have different names in different databases
  - One attribute may be a "derived" attribute in another table, e.g., annual revenue
- Careful integration of the data from multiple sources may help reduce/avoid redundancies and inconsistencies and improve mining speed and quality
- Redundancy can be checked using correlation Analysis.

Data Preprocessing

- Why preprocess the data?
- Data cleaning
- Data integration
- Data reduction
- Data transformation Discretization
- Summary

Data Reduction

- Warehouse may store terabytes of data: data analysis/mining may take a very long time to run on the complete data set
- Data reduction
  - Obtains a reduced representation of the data set that is much smaller in volume but yet produces the same (or almost the same) analytical results
- Data reduction strategies
  - Dimensionality reduction
  - Data compression
  - Numerosity reduction

Data Reduction: Dimensionality Reduction

- **Dimensionality reduction** is the process of reducing the number of random variables or attributes under consideration.
- ***Attribute subset selection*** is a method of dimensionality reduction in which irrelevant, weakly relevant, or redundant attributes or dimensions are detected and removed.
- For example, if the task is to classify customers based on whether or not they are likely to purchase a popular new CD at *AllElectronics* when notified of a sale, attributes such as the customer's telephone number are likely to be irrelevant, unlike attributes such as *age*.
- The goal of attribute subset selection is to find a minimum set of attributes such that the resulting probability distribution of the data classes is as close as possible to the original distribution obtained using all attributes.

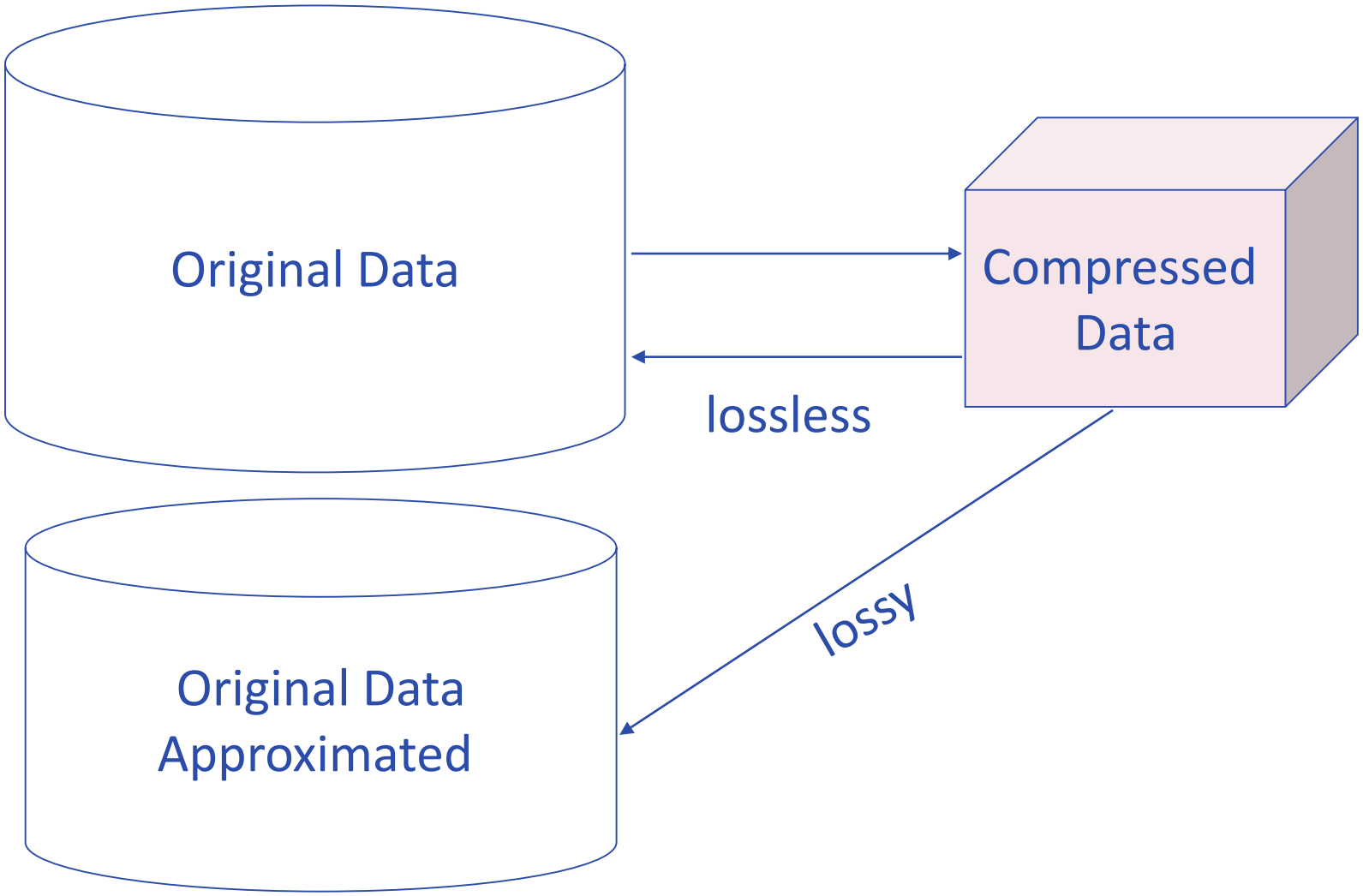
Data Reduction: Numerosity Reduction

- **Numerosity reduction** techniques replace the original data volume by alternative, smaller forms of data representation.
- Histograms use binning to approximate data distributions. A **histogram** for an attribute, *A*, partitions the data distribution of *A* into disjoint subsets, referred to as *buckets* or *bins*.
- Clustering techniques consider data tuples as objects. They partition the objects into groups, or *clusters*, so that objects within a cluster are "similar" to one another and "dissimilar" to objects in other clusters.
- Sampling can be used as a data reduction technique because it allows a large data set to be represented by a much smaller random data sample (or subset).
- Data Cube Aggregation: Imagine that you have collected data from *AllElectronics* sales per quarter for your analysis, however, you will be interested in the annual sales (total per year), rather than the total per quarter.

Data Reduction: Data Compression

- In **data compression**, transformations are applied so as to obtain a reduced or "compressed" representation of the original data.
- If the original data can be *reconstructed* from the compressed data without any information loss, the data reduction is called **lossless**.
- If, instead, we can reconstruct only an approximation of the original data, then the data reduction is called **lossy**.
- There are several lossless algorithms for string compression; however, they typically allow only limited data manipulation.
- Dimensionality reduction and numerosity reduction techniques can also be considered forms of data compression.

Data Reduction: Data Compression



Data Preprocessing

- Why preprocess the data?
- Data cleaning
- Data integration
- Data reduction
- Data transformation Discretization

  Summary

Data Transformation

- **Data transformation:** the data are transformed or consolidated into forms appropriate for mining processing.
- **Different strategies includes**
  - Smoothing – which works to remove noise from the data. binning, regression, and clustering
  - Attribute construction – new attributes are constructed from the given set of attributes.
  - Aggregation – summary or aggregation operations are applied e.g. construction of data cube
  - Normalization – data are scaled so as to fall within a smaller rage e.g. -1.0 to 1.0 or 0.0 to 1.0
  - Discretization - Values of numeric attribute are replaced by interval labels or conceptual labels. (concept hierarchy for numeric attribute)
  - Concept hierarchy generation for nominal data – nominal attribute values are generalized to higher-level concepts e.g. street is generalized to block, city or country.

Data Transformation by Normalization

- Normalization helps to prevent attributes with large ranges outweigh ones with small ranges.
- A database can contain n numbers of continuous type attributes. Where a larger range continuous type attribute or noise can shift the objects distance.
- For example: 'Income' attribute can dominate the distance as compared to 'Weight' and 'Age' attributes.
- The objective of normalization is convert all integer type attributes, so that there values fall within a small specified range, such as 0 to 1.0.
- There are many methods for data normalization. We study *min-max normalization*, *z-score normalization*, and *normalization by decimal scaling*.

Data Transformation: Normalization

- min-max normalization: performs a linear transformation on the original data.
- Suppose that min_A and max_A are the minimum and maximum values of an attribute, A. Min-max normalization maps a value, v_i, of A to v'_i in the range $[new\ min_A, new\ max_A]$ by computing

$$v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$$

- Example:- **Min-max normalization.** Suppose that the minimum and maximum values for the attribute *income* are \$12,000 and \$98,000, respectively. We would like to map *income* to the range [0.0, 1.0]. By min-max normalization, a value of \$73,600 for *income* is transformed to $\frac{(73600-12000)}{98000-12000}$ (1.0-0) + 0 = 0.716

Data Transformation: Normalization

- min-max normalization: (or *zero-mean normalization*), the values for an attribute, *A*, are normalized based on the mean (i.e., average) and standard deviation of *A*. A value, v_i , of *A* is normalized to $v'i$ by computing

$$vi' = \frac{vi - mean_A}{stand\_dev_A}$$

- Example:- **z-score normalization.** Suppose that the mean and standard deviation of the values for the attribute *income* are $54,000 and $16,000, respectively. With z-score normalization, a value of $73,600 for *income* is transformed to

$$\frac{(73600-54000)}{16000} = 1.225$$

Discretization

- Three types of attributes:
  - Nominal — values from an unordered set
  - Ordinal — values from an ordered set
  - Continuous — real numbers
- Discretization:
  - divide the range of a continuous attribute into intervals
  - why?
    - Some classification algorithms only accept categorical attributes.
    - Reduce data size by discretization
    - Prepare for further analysis

Discretization and Concept hierachy

- Discretization
  - reduce the number of values for a given continuous attribute by dividing the range of the attribute into intervals.
  - Interval labels can then be used to replace actual data values.
- Concept hierarchies
  - reduce the data by collecting and replacing low level concepts (such as numeric values for the attribute age) by higher level concepts (such as young, middle-aged, or senior).

Segmentation by natural partitioning

- Users often like to see numerical ranges partitioned into relatively uniform, easy-to-read intervals that appear intuitive or "natural". E.g., [50-60] better than [51.223-60.812]

The 3-4-5 rule can be used to segment numerical data into

relatively uniform, "natural" intervals.

* If an interval covers 3, 6, 7 or 9 distinct values at the most significant digit, partition the range into 3 equi-width intervals for 3,6,9 or 2-3-2 for 7
* If it covers 2, 4, or 8 distinct values at the most significant digit, partition the range into 4 equi-width intervals
* If it covers 1, 5, or 10 distinct values at the most significant digit, partition the range into 5 equi-width intervals

The rule can be recursively applied for the resulting intervals

Concept hierarchy generation for categorical data

- Categorical attributes: finite, possibly large domain, with no ordering among the values
  - Example: item type
- Specification of a partial ordering of attributes explicitly at the schema level by users or experts
  - Example: location is split by domain experts to street<city<state<country
- Specification of a portion of a hierarchy by explicit data grouping
- Specification of a set of attributes, but not of their partial ordering
- Specification of only a partial set of attributes

Specification of a set of attributes

- Concept hierarchy can be automatically generated based on the number of distinct values per attribute in the given attribute set.
- The attribute with the most distinct values is placed at the lowest level of the hierarchy.