Big Pre-Processing

CSCI316 Big Data Mining Techniques and Implementation

Lecture 3

Content

- Data
- Pre-processing:
  - Motivation, context, and definition.
  - Domain and Problem understanding
  - Data exploration
  - Data quality
  - Data Integration
  - Data selection
  - Data transformation
  - Imbalanced data
  - Feature creation

What is Data?

Data:
A collection of data objects and their attributes

- An attribute is a property or characteristic of an object.
- A collection of attributes describe an object.
  - An object can also be referred to as a: record, point, case, entity, instance, or sample.

Attributes

Objects

ID	Subject	Assgnmt	Labs	Exam	Grade
3204395	CSCI316	16	17	46	D
3194284	CSCI424	18	19	53	HD
3483509	CSCI316	12	13	39	C
3204395	CSCI319	15	17	51	D
3483509	INFO411	17	16	45	D
3204395	CSCI424	17	18	52	HD
3483509	INFO411	13	15	13	F

Class

What is Data?

- The number of attribute values in an object define the dimension of the object.
- Dimensions can be:
  - Fixed (all objects have the same number of attributes i.e. day-by-day weather observations)
  - Variable (i.e. shopping basket)
  - Continuous (i.e. sensor data as time series)
- Attributes can be distinguished by type:
  - Numeric (i.e. Discrete, binary, continuous, Fractional, Ratio, complex, …)
  - Symbolic (i.e. categorical, textual, symbols,…)
  - Single valued
  - Multi valued, compound
  - …

Attribute conversion

- It is often desirable to convert dimension and type of attributes without loss of information. Example:

ID	Subjects
3204395	CSCI319, CSCI316, CSCI424
3194284	CSCI424
3483509	CSCI316, INFO411, CSCI319

ID	CSCI316	CSCI319	CSCI424	INFO411
3204395	1	1	1	0
3194284	0	0	1	0
3483509	1	1	0	1

One-dimensional multi-valued symbolic attribute

Four-dimensional binary attribute

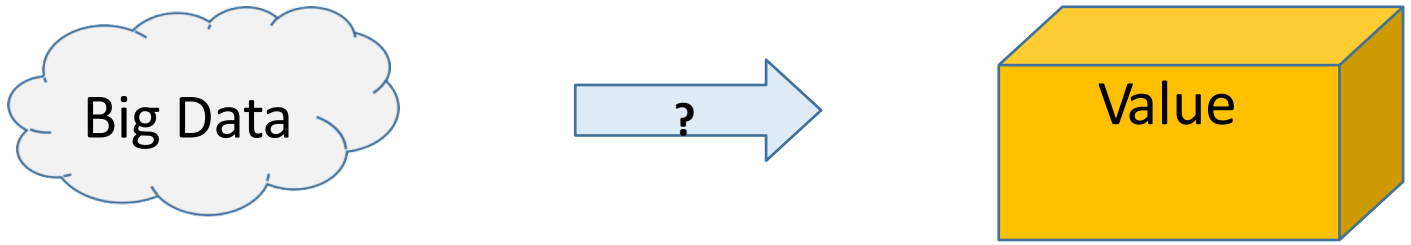
Sparsity

- Attributes can be sparse

ID	CSCI321	CSCI316	CSCI323	CSCI314	ISIT200	CSCI319	ITCS206	CSCI424	CSCI312	CSCI317	INFO411
3204395	0	1	0	0	0	1	0	1	0	0	0
3194284	0	0	0	0	0	0	0	1	0	0	0
3483509	0	1	0	0	0	1	0	0	0	0	1

- There is a tradeoff between number of unique records, dimensionality, and sparsity -> course-of-dimensionality.

Why Pre-processing?



- Objective of mining Big Data is to obtain insights that are of value.
- The performance and quality of knowledge extracted by mining techniques in any framework depends on:
  - Design, suitability, and performance of mining methods
    - Most techniques in Big Data are highly specialized and require data to be presented in a specific form and with specific properties. The performance of these methods is also often influenced by the property of data.
  - Quality and suitability of data.
    - Problem: Quality and suitability of big data cannot be guaranteed (due to the 4 Vs).

What is Pre-processing?

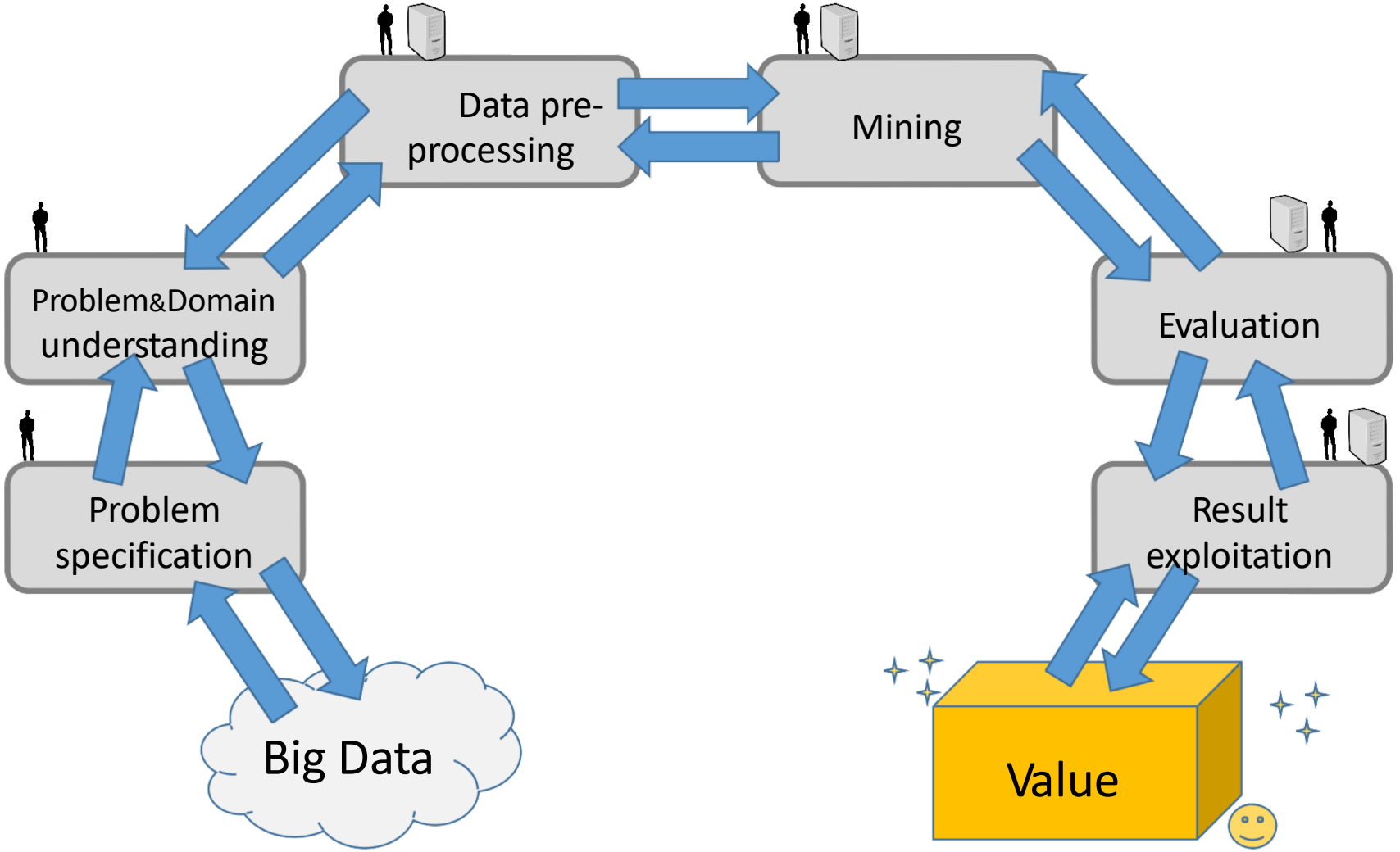
Pre-processing:

A set of machine implementable techniques used prior to mining big data.

Pre-processing addresses issues with:

- problem and domain understanding
- data quality (i.e. missing values, inconsistency, redundancy, noise,…)
- data format (i.e. unstructured data, unsuitable data types, …)
- size of the data (i.e. sheer volume,…)
- data from varying sources (i.e., data integration)
- Imbalanced data (i.e. imbalanced categorical data,…)
- …

Why Pre-processing?



Domain Understanding

Domain understanding:

Knowledge about the environment of the data and the understanding of the environment to which results are to be applied to.

- Domain understanding helps to understand the meaning and property of the data.
  - Was data collected manually or automatic?
  - How reliable or accurate were the processes that collect the data?
  - What is the meaning of each of the attributes?
- Domain understanding is essential to the formulation of a data mining problem.
- Domain understanding is usually acquired from
  - Domain experts, research, and data exploration / pre-processing
- *See Appendix for an example (at the end of slides)*

Data exploration

Data Exploration:
Preliminary investigation of the data to better understand its specific characteristics

- It can help to answer some of the big data questions
- To help in selecting the right pre-processing tools
- To help in selecting appropriate mining algorithms
- Things to look at:
  - Attribute balance
  - Dispersion of data attribute values
  - Skewness, outliers, noise, missing values
  - Attributes that vary together
  - …
- Visualization tools are important
  - Histograms, box plots, scatter plots, …

Data Imbalance

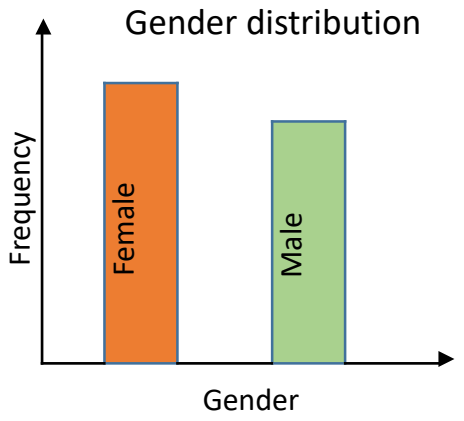
- For discrete attributes
  - What is the frequency of each attribute value?
  - Is there a considerable less frequent attribute value?
- For continues attrbiutes
  - Is there a “long tail”?
- Some mining algorithms are affected by value imbalance
  - Identify the problem in the data exploration phase

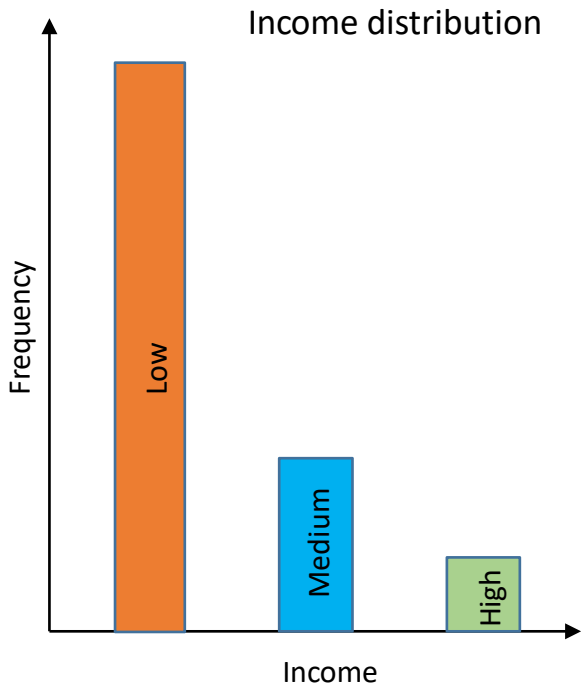
Data Imbalance

- If the affected attribute is a target variable: Class imbalance
- If the affected attribute is an input variable: Feature imbalance

Instance	Gender	Married	Years experience	Income
1	M	Y	13	High
2	F	N	18	Medium
3	M	N	3	Low
…	…	…	…	…
1999999	F	Y	9	Low

Class





Data Imbalance: Some useful statistics

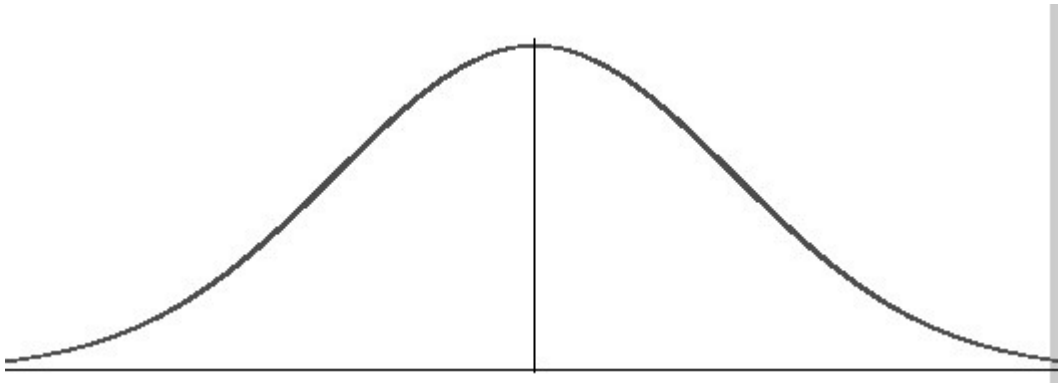
- Discrete attributes:
  - Frequency of each value
  - **Mode** = value with highest frequency
- Continuous attributes
  - Range of values, i.e. **min** and **max**
  - **Standard Deviation**
  - **Mean** (average)

    Sensitive to outliers
  - **Median**

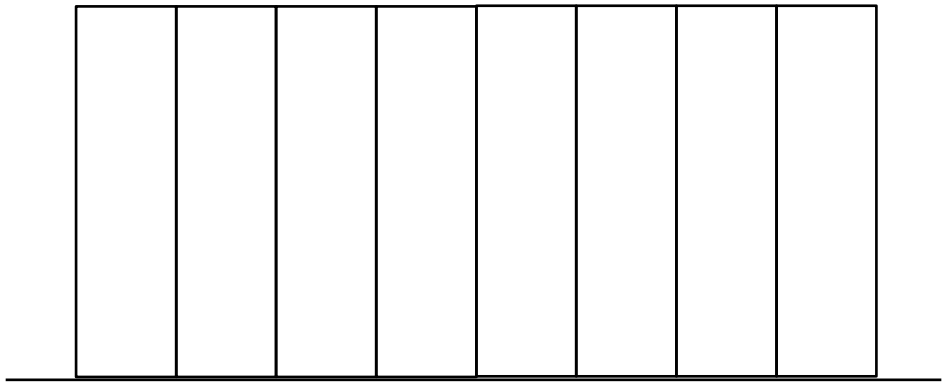
    Better indication of the ”middle” of a set of values in a skewed distribution
  - **Skewed distribution**

    mean and median are *quite* different, std. deviation is large

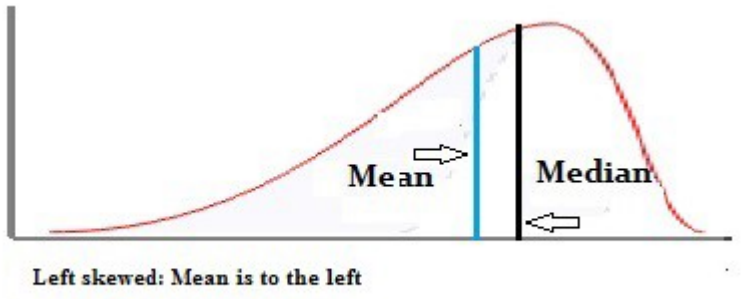
Data Imbalance: Distributions



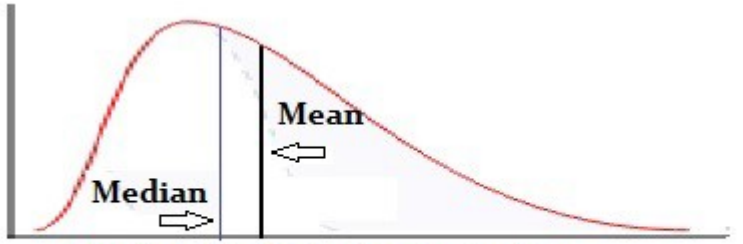
Normal distribution



Uniform distribution



Left skewed: Mean is to the left



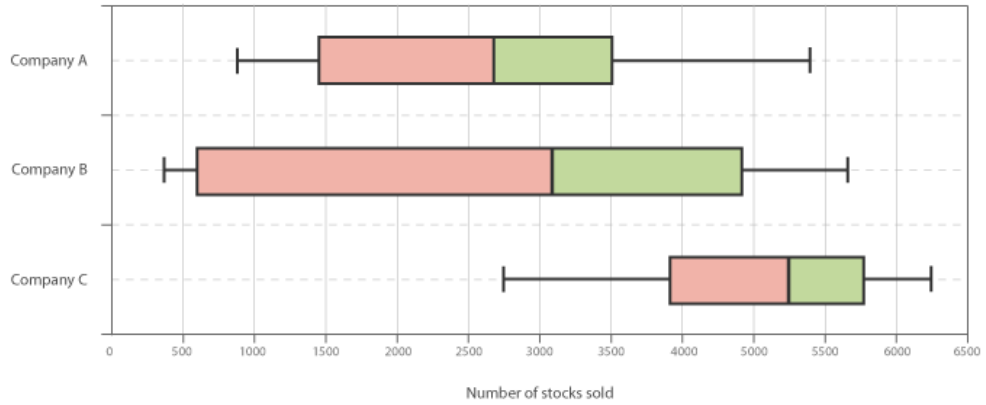
Right skewed distribution: Mean is to the right

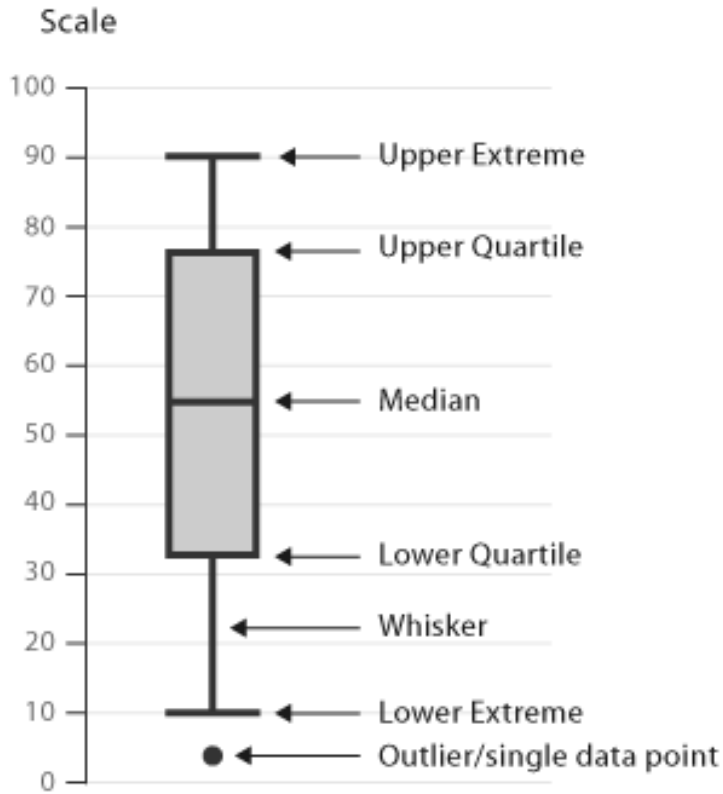
Caution: The reverse definition can also be found in literature.

Box (and Whisker) Plots

A box and whisker plot can provide information useful information about an attribute

- sample's range
- median
- normality of the distribution
- skew (asymmetry) of the distribution
- plot extreme cases within the sample





Dispersion of Data

- How do the values of an attribute spread?
  - Variance

$$var = \frac{1}{n}\sum_{i=0}^{n-1}(x_i - \bar{x})^2$$

    - Variance is sensitive to outliers
  - Interquartile range (**IQR**) (see a later slide)
  - What if the distribution of values is multimodal, i.e. data has several *bumps*?
    - Visualization tools are useful

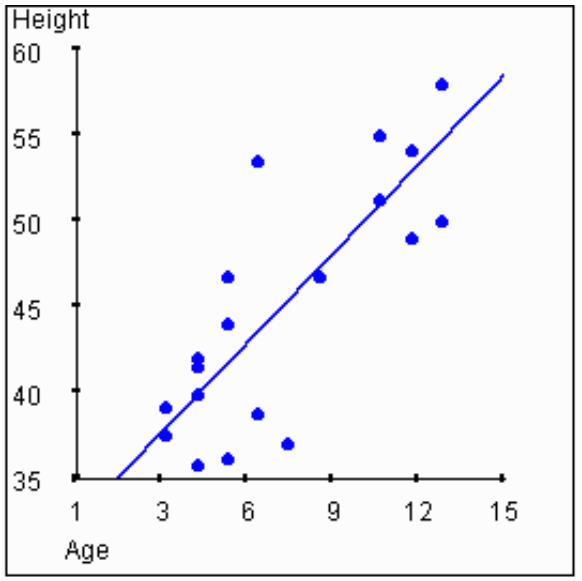
Attributes that Vary Together

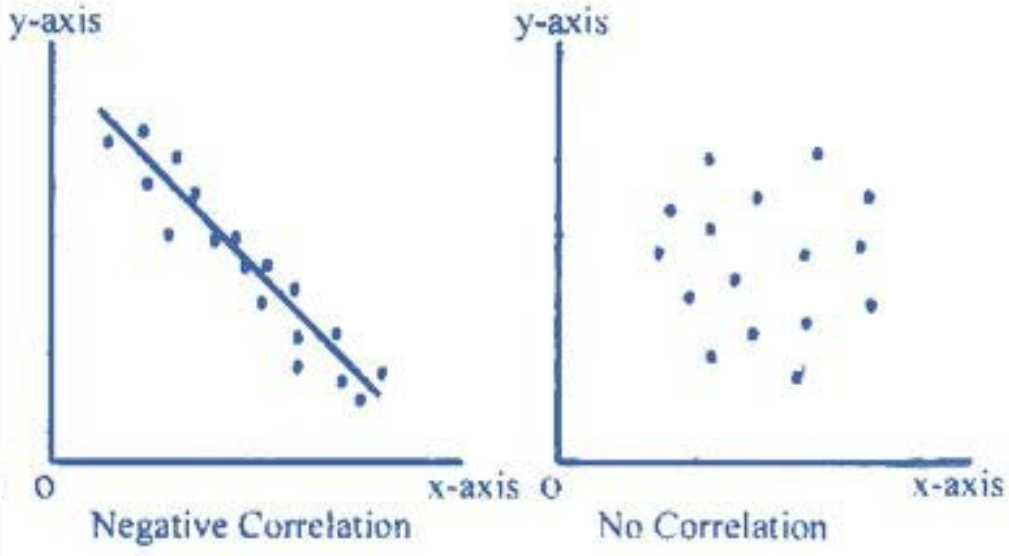
Correlation is a measure that describes how two attributes vary together.

$$corr = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{(n\sum x^2 - (\sum x)^2)(n\sum y^2 - (\sum y)^2)}}$$

n=number of samples
x=value of first attribute
y=value of second attribute

- If corr > 0 then data is positively correlated
- If corr < 0 then data is positively correlated
- Strong correlation: corr ≈ |1|
- Weak correlation: corr ≈ 0





Data quality

Issues affecting Data quality:

- **Outliers**:
  - An observation that is distant from other observations.
  - May be due to variability in the measurement, indicate an error, or indicate the presence of unusual observations.
  - An outlier can cause serious problems in statistical analyses.
- **Missing values**
  - Is data that has not been stored or gathered.
  - Can indicate faulty sampling but may also be due to cost restrictions, protocol changes, or limitations in the acquisition process.
- **Noise**
  - Is corrupted or distorted data containing false information
  - It can be difficult to distinguish between outliers and noise
  - Required data cleaning (polishing)

Outliers

- What constitutes an outlier is subjective.
- There is no rigid mathematical definition although some rules of thumb exist:
  - 3-σ method for normally distributed Data: observations that deviate by more than three time the standard deviation from the mean.
  - Interquartile method (IQR): Observations outside the range
    $[Q_1 - k(Q_3 - Q_1), Q_3 + k(Q_3 - Q_1)]$, using k=1.5

What to do with outliers?

- Retention: Keep outliers and choose mining algorithms that are robust to outliers.
- Exclusion: Set outliers aside and process them separately if necessary
- Value transformation: Transform value to become more similar to other values (i.e. through binning)

Missing values

- Handling missing values
  - Obtain missing values from a secondary source.
  - Use expert knowledge
  - Estimate missing values (exercise caution when using this approach)
    - Replace by most frequent or average
  - Use non-missing data to predict the missing values
    - Maintain the between-attribute relationships
    - Different replacements can be generated for the same attribute
  - Apply a data mining technique that can work in the presence of missing values (e.g. decision trees)
  - Eliminating objects with missing values only as a last resort!
    - No more than 5% of the records

Noise

- Noise can be obvious i.e. the presence of symbolic information in a numeric attribute.
- Noise can be hidden i.e. a patient of age 127 in a set of patient records, or a GPS location data deviating significantly from the true location.
  - Domain knowledge is often required to identify “hidden” noise.

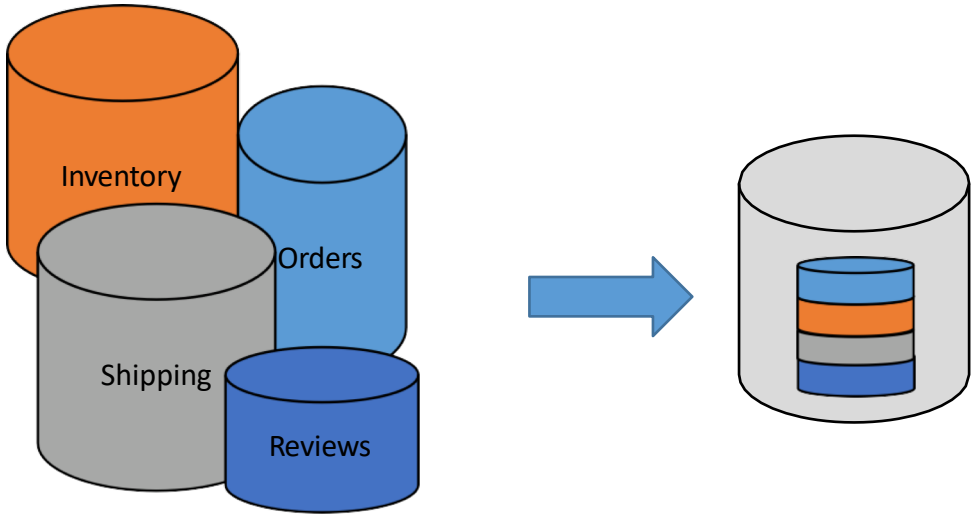
Dealing with noise:

- Data polishing:
  - Remove the noisy datum then treat as missing value.
- Noise filtering:
  - Identify and remove noisy instances (last resort measure)

Data Integration

Data integration:
Combining data from different sources

- Data is often created independently
- Combining data can help with
  - Finding missing values
  - Create a more complete description of an instance.
- The goal
  - tie together different sources, controlled by different people.



Data Integration: Example 1

Student Management System

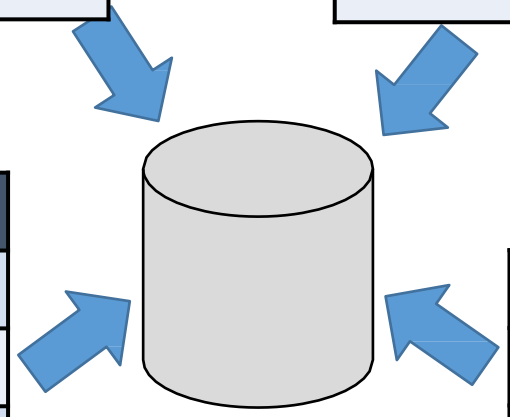
Student number	Name	Category
3204395	Charles Darwin	Undergraduate
3194284	Bill Smith	Master
3483509	Anne Wilson	Honors
…	…	…

Enrollment system

Student number	Name
3204395	CSCI319, CSCI316, CSCI424
3194284	CSCI424
3483509	CSCI316, INFO411, CSCI319
…	…

Subject Database

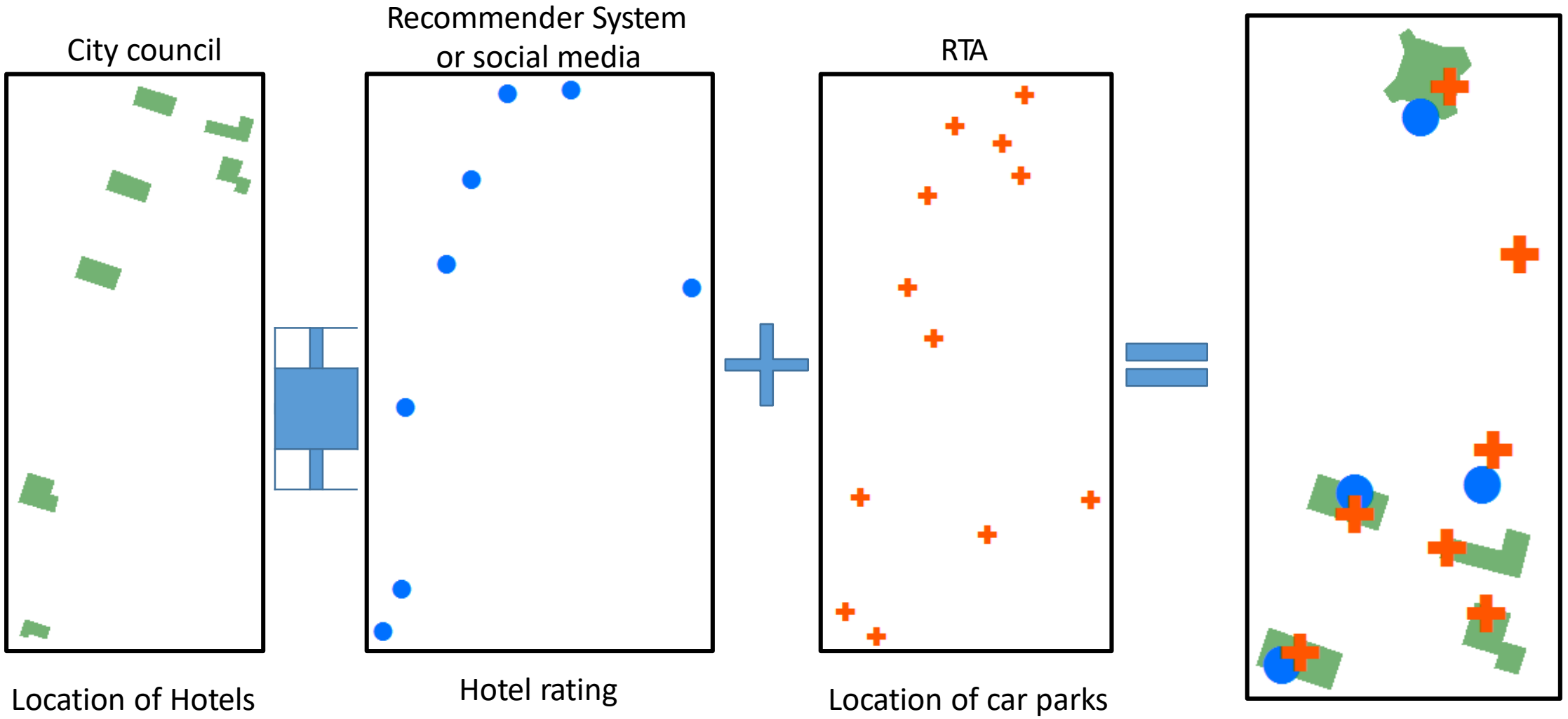
Code	Name	Session
CSCI316	Big Data Impl.	Spring
INFO411	Data Mining	Autumn
CSCI319	Dist. Systems	Autumn
…	…	…



CSCI316 Moodle site

Student number	Assignment	Lab
3204395	16	17
3483509	12	13
…	…	…

Data Integration: Example 2



Data Integration

- Can be challenging
  - Computationally expensive (relative to normal data retrieval operations)
  - Attribute matching is not always clear (i.e. is T. Kennedy” same person as “Ted Kennedy”? Is “Istanbul” same as “Constantinople”?)
  - Risk to make errors (introduce noise)?
  - Data may differ in format (i.e. integrate video of a newsreader with spoken text).

Data Integration

Approaches:

- Matching schema element names (if schema available)
  i.e. “BooksAndCDs/Categories” ~ “BookCategories/Category”
- Descriptions and documentation. For example:
  ItemID: unique identifier for a book or a CD
  ISBN: unique identifier for any book
- By data types, data instances.
  For example: DateTime ≠ Integer,
  addresses have similar formats
- By structure. For example: All books have similar attributes
- Use domain knowledge!

Data Aggregation

Data Aggregation:
Combining two or more objects into a single object.

- Purpose
  - Data reduction
    - Reduce the number of attributes or objects
  - High-level view of the data
    - Easier to discover patterns
  - More “stable” data
    - Aggregated data tends to have less variability

Caution: Aggregation can lead to the loss of *relevant* information

Data Aggregation

- Example:

ProductID	Location	Date
AX325	Sydney	17/08/2018
TN788	Brisbane	17/08/2018
PU376	Sydney	17/08/2018
TN788	Sydney	17/08/2018
AX325	Brisbane	18/08/2018
KA808	Sydney	18/08/2018
KA808	Brisbane	18/08/2018
PA123	Brisbane	19/08/2018
PA123	Sydney	19/08/2018

Aggregate location

ProductID	Date
AX325	17/08/2018
TN788	17/08/2018
PU376	17/08/2018
TN788	17/08/2018
AX325	18/08/2018
KA808	18/08/2018
KA808	18/08/2018
PA123	19/08/2018
PA123	19/08/2018

Aggregate duplicates

ProductID	Date
AX325	17/08/2018
TN788	17/08/2018
PU376	17/08/2018
AX325	18/08/2018
KA808	18/08/2018
PA123	19/08/2018

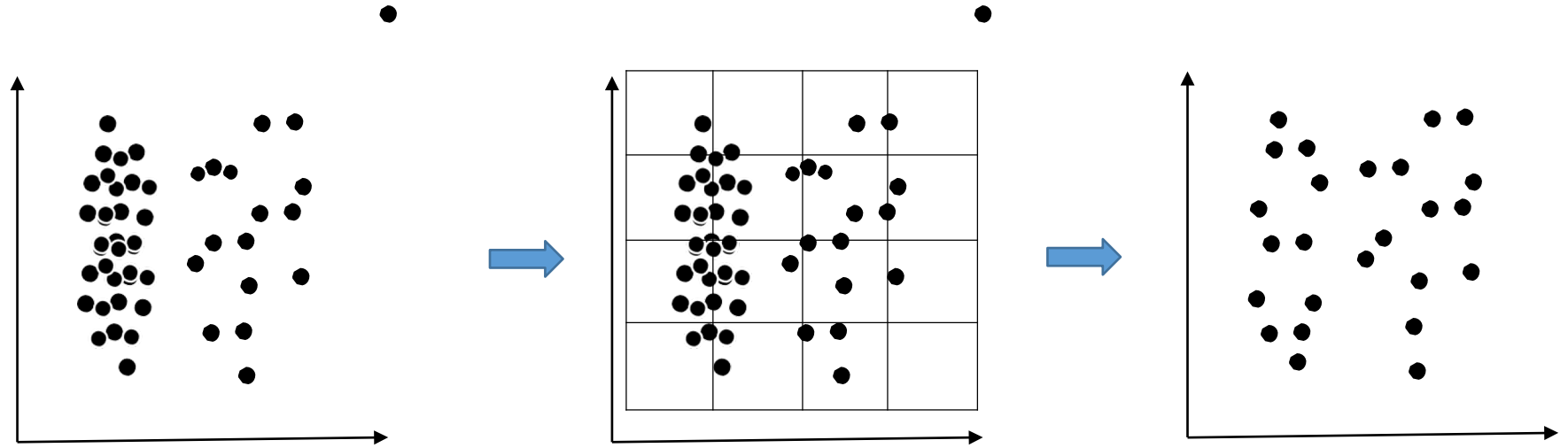
Instance Selection

Instance Selection:
Identify suitable examples from a very large amount of instances for a mining algorithm.

- Helps to remove noise.
- Helps to remove redundant instances.
- Helps to reduce the amount of data that needs to be processed.
- Helps to focus on important data.
- Can help to balance data.

Instance Selection

Example: Grid method



Original data

Place regular grid over the data

Keep at most 2 instances from each grid point.

Instance Generation

Instance Generation:
Generate new artificial data to either replace or extend the original data.

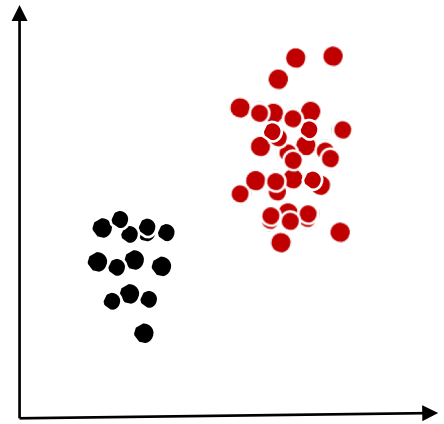
- To fill regions in the domain which have no representative samples in the original data.
- To condense large amounts of instances by using fewer examples.

Common approaches:

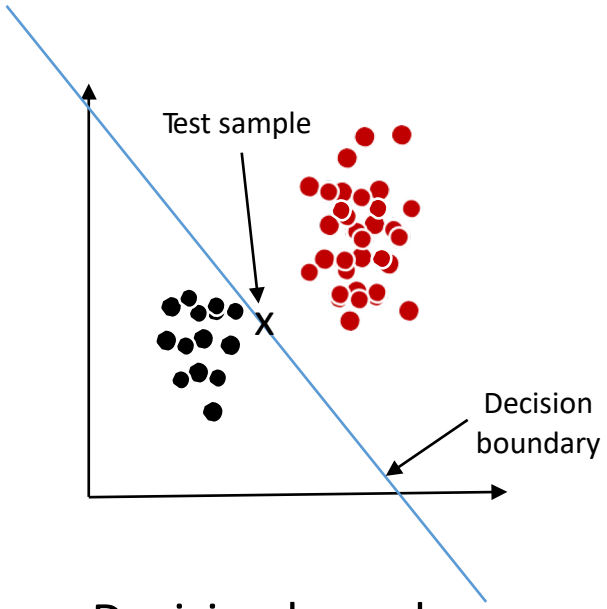
- Prototype methods (i.e. K-means)
- Sample distortions (i.e. via noise injection)

Instance Generation

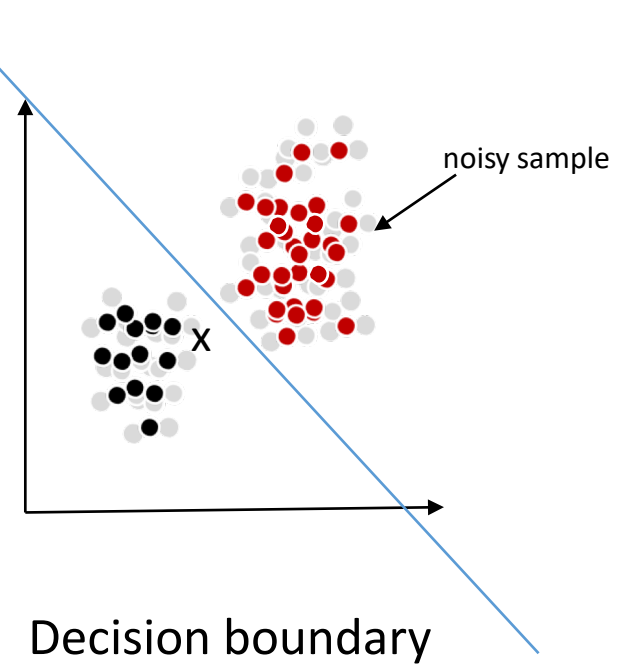
Example: Instance generation via sample distortion.



Two populations of data



Decision boundary and test sample



Decision boundary with noisy samples

Data or Feature Transformation

Data or Feature Transformation: Apply a function to an attribute or feature to obtain data suitable for further processing.

Example:
A database categorizes employees by income level “low”, “medium”, “high”. But many algorithm require numerical data. Convert income levels: “low” = 1, “medium = 2, “high” = 3.

Risk:

- Take care not to introduce artefacts or remove important characteristic of data.
  Example: encode “red” =1, “green” =2, “blue”=3. But in Euclidean space the value 1 is more similar to 2 than to 3 thus falsely implying that red is more similar to green than to blue.

Data or Feature Transformation

Common transformation techniques:

- Format transformation
- Discretization/Binning
- Normalization
- Scaling
- Filtering (remove noise)

Format Transformation

- Format transformation converts the format or type of an attribute to another format or type.
- Example: Working with dates can be cumbersome because dates can come in many forms and can be ambiguous.

  Tuesday, 7/1/2025 vs Tuesday, 1/7/2025 (in America: 1st of July, in Australia: 7th of January)
- Examples: convert date to number of seconds since a start date, or convert date to float (current_date - start_date)/(end_date-start_date)
  - similar to scaling (talk later)

One-hot encoding

Simple (direct) encoding is a technique that converts strings into numerical values (since many ML tools prefer to work with numbers).

One-hot encoding converts strings into vectors of bits (0 or 1) and 1 appears once in each vector (thus “one-hot”).
Advantage: If non-numeric categorcial values do not share similarities (are equidistant in their meaning) then a one-hot encoding will convert the values to numerical ones that share the property of equidistance with the original value.

Example: The values “red”, “green”, “blue” may be considered equidistant to each other. A one-hot encoding (i.e. “red” = [0,0,1], “green” = [0,1,0] and “blue” = [1,0,0]) would also be equidistant in Euclidian space.

Discretization/Binning

- Discretization/binning is a specific form of format transformation.
- Objective is to transform quantitative data into qualitative data by dividing numerical features into a limited number of non-overlapped intervals.
- Can alleviate the problem with outliers.
- Example: discretization of temperature values into 3 bins

  temps = [13.5, 23.2, 29.1, 15,7, 44.1, 27].

  Discretised, i.e.:

```
temps[temps < 20] = 1
temps[(temps >= 20.0) & (temps < 30)] = 2
temps[temps >= 30] = 3
```

  Note that accuracy of values can be compromised. This can be controlled by varying the number of discrete values, but leading to higher computational cost.

Normalization

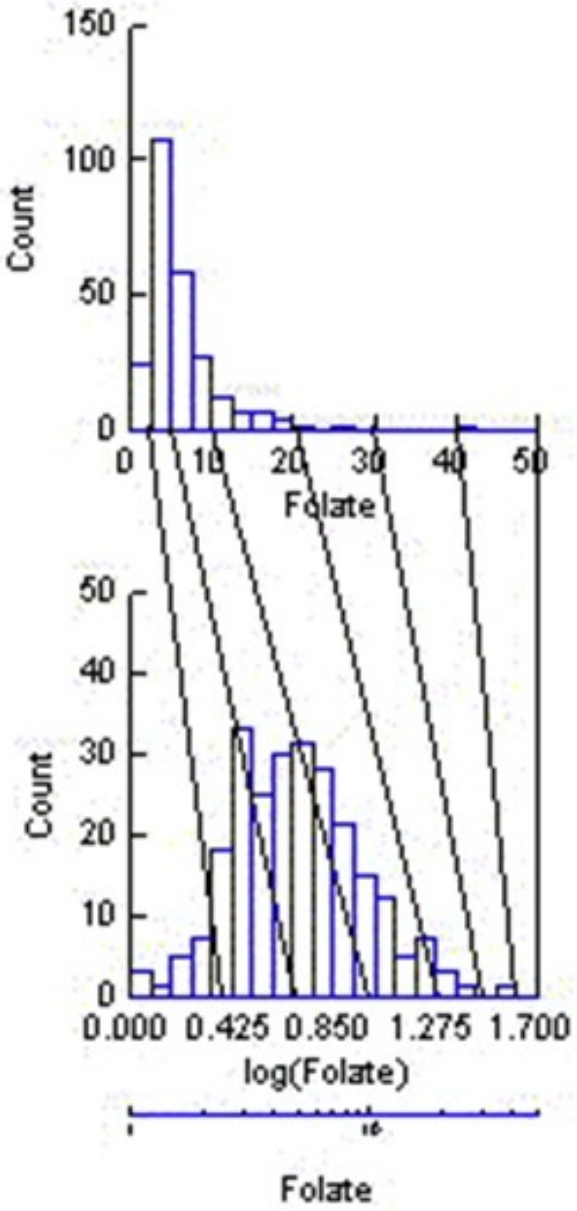
Normalization
Adjust values on different scales to a common scale

- Attributes can vary significantly in value range.
- Common normalization methods are:
  - Min-max normalization
    - $v = (x - x_{min}) / (x_{max} - x_{min})$ # values are scaled to [0:1]
    - $v = 1 - x = 2\left(\frac{x - x_{min}}{x_{max} - x_{min}}\right)$ # values are scaled to [-1:1]
  - Z-score normalization (also called zero-normalization or simply standardization)
    - Normalize values such that mean = 0 and std. dev. = 1 using $v = \frac{x - \bar{x}}{\sigma}$

Normalization

- Normalization can also help with skewed value distributions.
- For example, if it was found that the value distribution is exponential then this can be converted into a pseudo normal distribution by applying the log function (called log-transformation).
- Example: $x_{new} = \log(x_{old} + \text{offset})$

  i.e. use offset = x_{min} to avoid negative values.



Scaling

Scaling:
standardize the range of independent variables

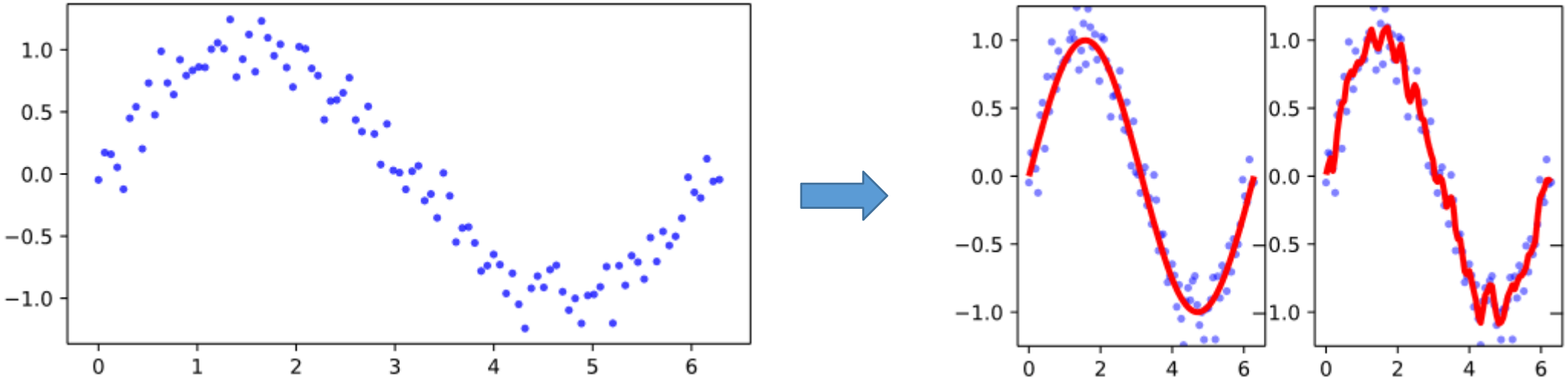
- Scaling is a form of normalization.
- While normalization can be applied to a set of attributes, scaling is applied to each attribute independently.
- Scaling normally involves a linear transformation of data.
- Thus, min-max normalization and z-score normalization when applied to a single attribute is called scaling.
- Another common scaling method is unit-length scaling: $v = x/||x||$

Filtering

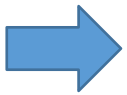
Types of filters:

- Instance filter: Remove one or more instances
- Attribute filters: Remove one or more attributes
- Value filter: Remove one or more data values

Application: Noise removal



Example: Median filter on images



Data Sampling

Sampling:
A technique employed for selecting a subset of the data.

- When is sampling useful to mining Big Data?
  - When it is too expensive or too time consuming to process all data.
  - To measure a classifier's performance, the whole data set is divided into a training set and a test set.
  - To obtain a better balance between class distributions.

Sampling techniques

- Simple Random Sampling
  - Every sample has the same chance of being selected
  - Perfect random sampling is difficult to achieve in practice
  - Two methods:
    1. Sampling without replacement: Sample is removed from the full dataset once selected. A selected item cannot be selected again.
    2. Sampling with replacement: Selected samples remain in the full set. Items can be picked up more than once.
       - Useful for small data sets

Drawback: by bad luck, none of the samples of a less frequent (rare) target attribute may end up in the sampled set.

Sampling techniques

- Stratified sampling
  - Split the data into several partitions (strata, i.e. by class); then draw random samples from each partition.
  - Each strata may correspond to each of the possible classes in the data.
  - The number of items selected from each strata can be:
    - the same number, or
    - proportional to the strata size.
  - However, stratification provides only a primitive safeguard against uneven representation of classes.

Imbalanced data

Class imbalance:

There is a significance difference in the number of samples for each of the classes (or target variables).

- Many supervised learning algorithms are biased towards the majority class.
- Example: Predict the credit worthiness of customers by using a given training set. 99% of the instances in the training set are credit worthy. A trained model may then classify all samples as being credit worthy (which is a bad model!). The model's accuracy would be 99% falsely implying a high level of accuracy.

Imbalanced data; Sampling

- **Undersampling**: Select all samples from the minority class then randomly select an equal number of samples from the majority class. This reduces the number samples in a training set.
- **Oversampling**: Select all samples, then add duplicates from the minority class. This increases the number of samples in a training set.
- Rule of thumb: Use oversampling when the dataset is small, use Undersampling when the dataset is very large.

Cost sensitive learning

- Cost sensitive learning can be applied to learning algorithms that support a learning rate to counteract class imbalance. The learning rate is adjusted to weight the influence of minority samples stronger than that of samples from a majority class.
- This is not actually a pre-processing technique as it modifies the mining algorithm.

Feature Generation/Creation

- Create new attributes that can capture the important information in a dataset more efficiently than the original attributes.
- Always problem dependent.
- Three general methodologies:
  - Extracte features from (raw) data
    - domain-specific
    - From an image as a set of pixels one might extract features such as whether certain types of edges are present
  - Define new features from other features
    - combining features **Ex**: *density = mass / volume*

Feature Selection

Feature selection:

Identification and removal of as much irrelevant and redundant information as possible.

- Goal is to select a subset of features that still describe the original problem.
- Good feature selection methods can
  - Reduce memory and processing requirements during mining.
  - Decrease the risk of overfitting.
  - Simplify the visualization of results

Feature Selection

- There are many feature selection techniques!
- Almost always problem dependent.
- Usually, select those are most correlated to the target variables, positively or negatively
- Use criteria like Information Gain measures (talked in a future lecture)
  - how informative a feature is (how much value a feature has in separating pattern classes).

Conclusions

- Pre-processing is:
  - An essential and very important step in Big Data!
  - Depends on the data, mining technique, and problem specification.
  - Can have a direct impact on the quality of results (and hence on value).
  - Can be very time consuming (sometimes even more so than mining itself)
  - Tight integration of pre-processing and mining is becoming increasingly popular.

Appendix: Preliminary Investigation of the Data

First steps:

- Data format (binary?, compressed?, ASC-II?, lossy compressed?, …)
- Data syntax (where can which attribute be found, definition of the attributes)
- Data source (on disk, on tape, streaming, location, owner, …)
- Size of the data
  - Number of objects (countable many? Finite?, …)
  - Dimension and type of attributes
- Is time a relevant factor?
  - Temporal information (i.e. time sequences)?
  - Property of data change over time?
  - Age of the data (still valid?)

Appendix

Answer the following questions:

- Assume that you have access to images that were captured by a dash camera mounted to a police vehicle and wirelessly uploaded to a central database.
  1. What defines the dimension of this data?

  2. Is the dimension fixed, variable, or continuous?
  3. Of what type is the data?
- Assume that the images are annotated by meta data that describe the location, date and time, exposure-duration and ISO setting of each image.
  1. What is the dimension of this meta data?
  2. Of what type is this meta data?

Appendix

Answer the following questions:

- Assume that you have access to images that were captured by a dash camera mounted to a police vehicle and wirelessly uploaded to a central database.
  1. What defines the dimension of this data? The resolution of the image (number of pixels) and the number of color channels per pixel.
  2. Is the dimension fixed, variable, or continuous? Dimension if fixed.
  3. Of what type is the data? Binary (possibly compressed).
- Assume that the images are annotated by meta data that describe the location, date and time, exposure-duration and ISO setting of each image.
  1. What is the dimension of this meta data? eight(nine if height is part of the location)
  2. Of what type is this meta data? Location=(float, float), date=date, time=float or time=(int, int, int), exposure=float, ISO=int

Appendix

Assume that your task is to perform sentiment analysis on ancient Egyptian text. The data given to you consists of scanned hieroglyphs stored on a DVD.

1. What is the most likely data format (binary, text, lossy/lossless compressed)?
2. What is the likely type of the attributes?
3. Would your answer change if you were to perform sentiment analysis on ancient Greek text?

Did you notice the importance of domain knowledge for this task?

The Rosetta Stone

Appendix

Assume that your task is to perform sentiment analysis on ancient Egyptian text. The data given to you consists of scanned hieroglyphs stored on a DVD.

1. What is the most likely data format (binary, ~~text~~, (~~lossy~~) or lossless compressed)
2. What is the likely type of the attributes? int
3. Would your answer change if you were to perform sentiment analysis on ancient Greek text? Yes, Greek uses an alphabet, data can be presented as plain text.

Did you notice the importance of domain knowledge for this task?

The Rosetta Stone