

Data Mining and Knowledge Discovery in Databases Process

Week 1

Nurul Afiqah Burhanuddin
nurul.afiqah@ukm.edu.my
Room 2119

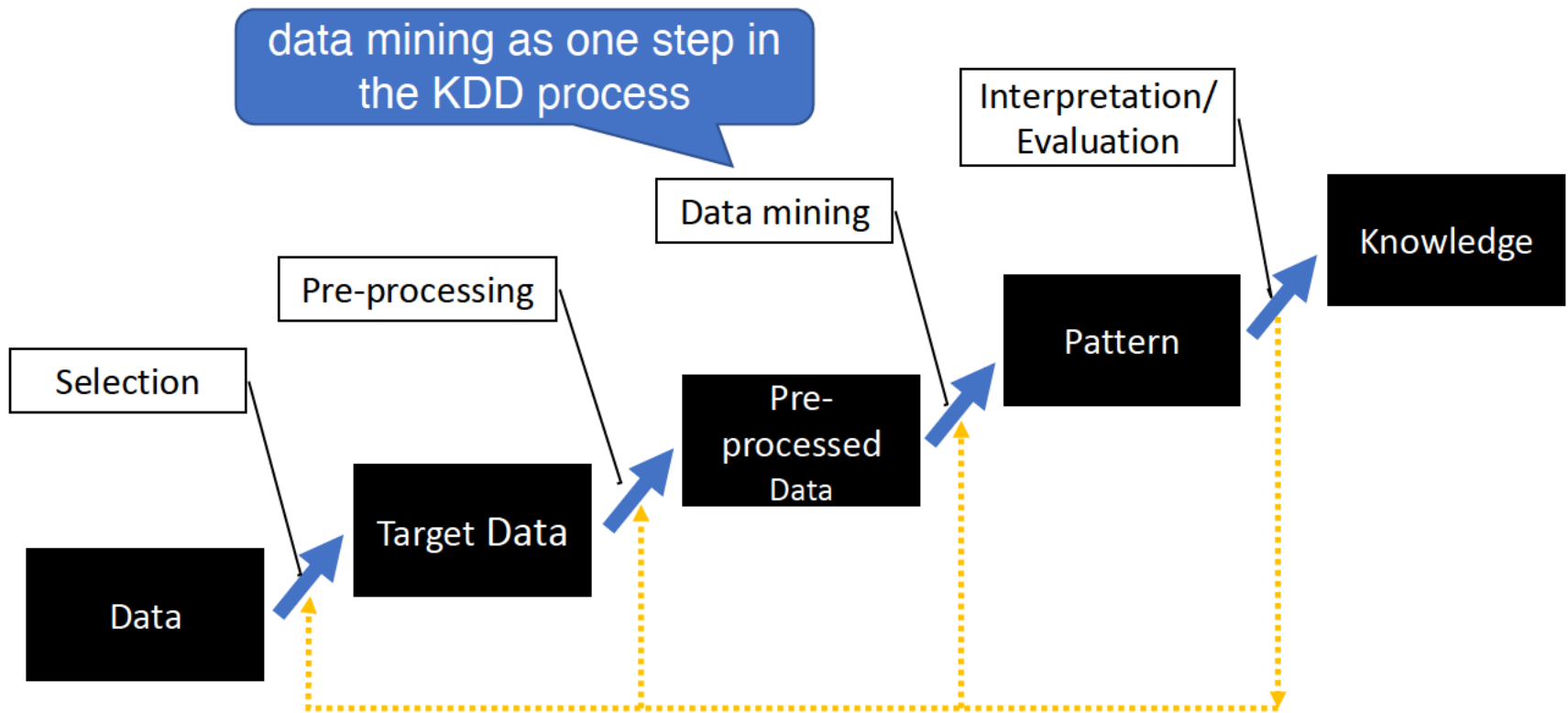
Introduction

- The term Knowledge discovery in databases (KDD) appeared around 1989 and is due to Piatetsky-Shapiro.
- **KDD** is defined as the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data.

The term “non-trivial”: some search or inference is involved, i.e. it is not a straightforward computation like computing the average.

- KDD has much in common with exploratory data analysis in statistic. In contrast to traditional approaches in statistics, KDD typically operate in the context of larger data sets with richer data structures.
- **KDD Process** is the process of using the database along with any required selection, pre-processing, subsampling, and transformations of it; to apply data mining methods (algorithms) to enumerate patterns from it; and to evaluate the products of data mining to identify the subset of the enumerated patterns deemed "knowledge".

KDD Process

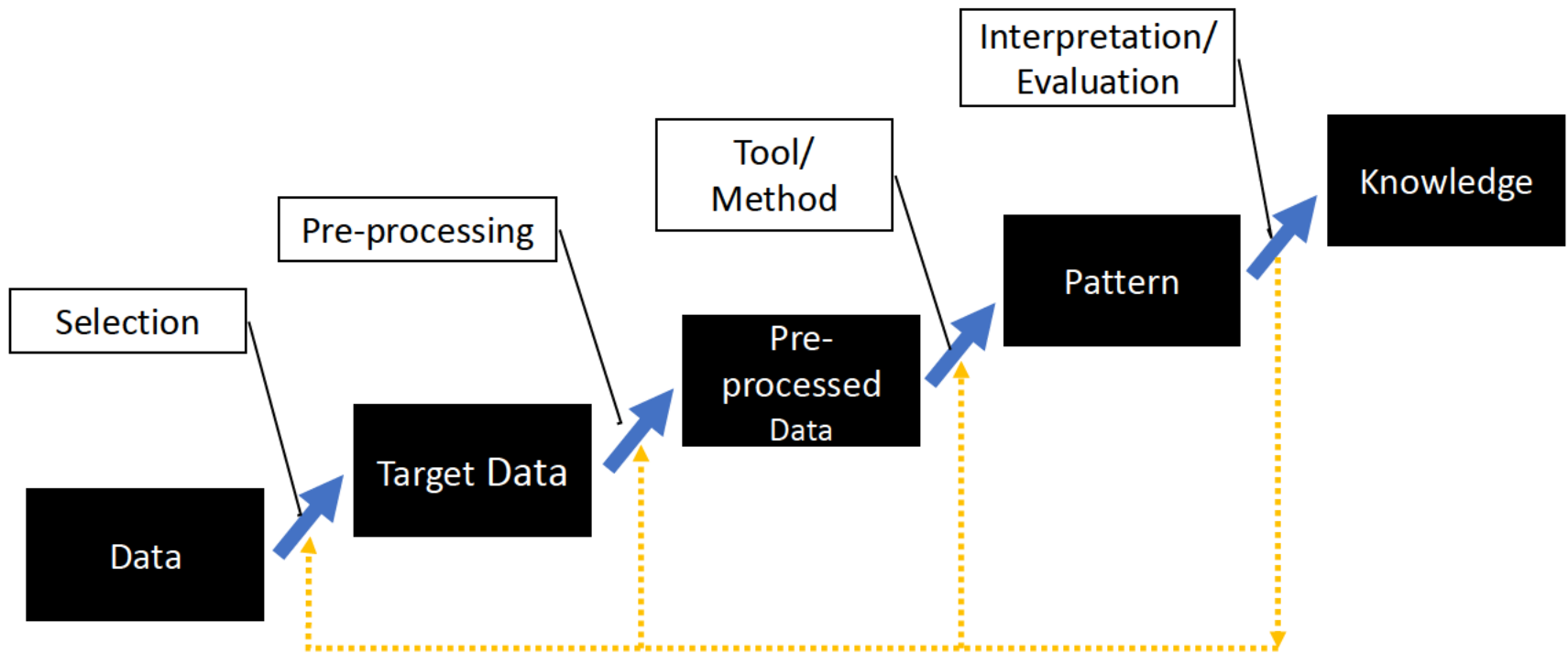


The process involve iteration and may contain loops between any two steps.

Introduction

- Relationship between data mining and KDD: two popular lines of thought exist. One is to treat data mining as a synonym for KDD, while others view data mining as merely a step in the KDD process.
- The term data mining is becoming more popular than the KDD.
- In this course, the term data mining is used to refer to the entire KDD process
- **Data mining** is the process of using the database along with any required selection and pre-processing, applying any tools to enumerate patterns from it, and evaluating and interpreting the enumerated patterns, which may be deemed as "knowledge".

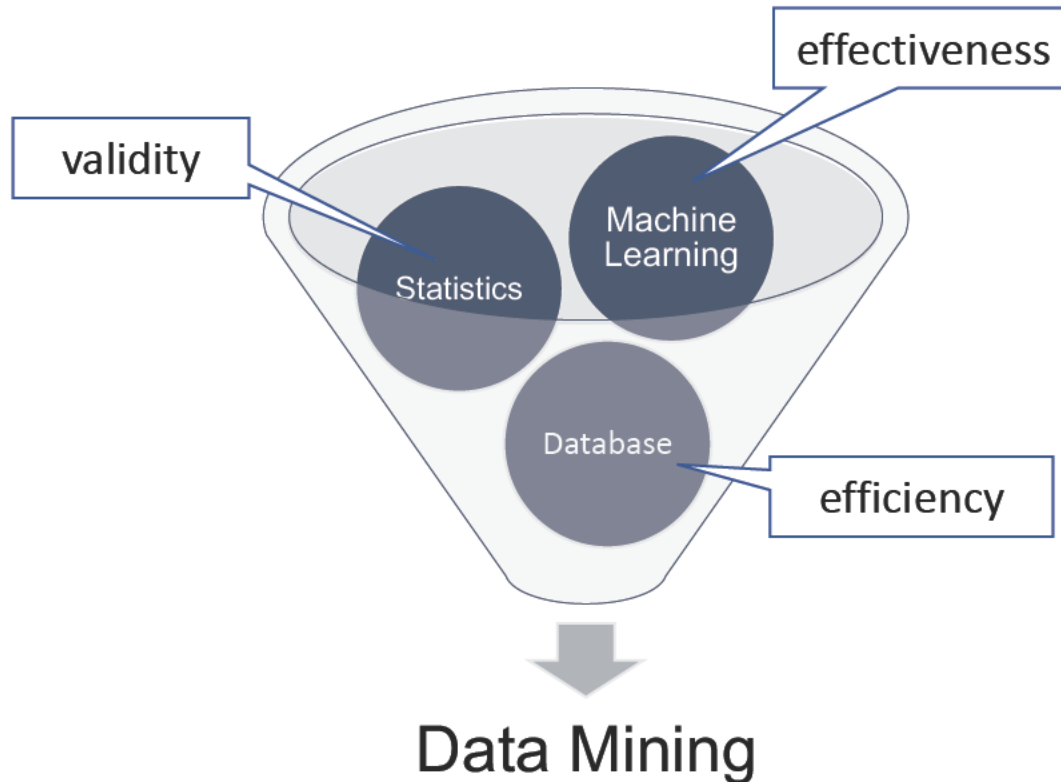
Data Mining



The process involve iteration and may contain loops between any two steps.

Introduction

- interdisciplinary nature: databases, machine learning, statistics, information retrieval, data visualization, parallel and distributed computing, etc.



Introduction

- What kinds of data can be mined?

In the past, many tools could only effectively analyze flat tables. However, with advances in information system technologies, more sophisticated tools have been developed for mining rich data formats, such as:

1. Numerical data
2. Time series data
3. Sequence data
4. Text data
5. Spatial data
6. Graph data
7. Web data

- Data pre-processing:

1. Cleaning: to remove noisy, incomplete and inconsistent data
2. Integration: to combine data from multiple sources
3. Reduction: to obtain a representation of the data set that is much smaller in volume
4. Transformation: to transform the data in the form which is appropriate for target application

Introduction to R

Setting up

R console:

- You can type commands directly into the console and press Enter to execute those commands. The results will also be shown here. However, you will lose all this when you close the session
- The console by default shows a **>** when R is ready to accept a command.
- The console will show a **+** when a command is not complete. Press esc to cancel the command.

R script: Script files are nothing more than text files of the commands that you enters. However, they offer several advantages over simply using the R console:

- You can execute your commands directly from the script file.
- You can run the entire file at once, executing multiple commands in quick succession, or a subset of commands by highlighting only those that you wish to execute.
- Easier to handle long commands, especially those with titles, labels and other arguments that make the command go longer than the visible window.
- You can save and load these files quickly.

Introduction to R

Setting up

Working directory: the location on your computer R will use for reading and writing files.

```
getwd() #Show the working directory  
setwd("file path") #Change the working directory  
setwd(choose.dir()) #Change the working directory interactively  
dir() #List files in the working directory  
dir.create("C:/test") #Create folder "test" in drive "C:"  
setwd("C:/test") #Change the working directory to "C:/test"
```

Packages: Packages are collections of R functions, data, and compiled code. R comes with a standard set of packages. Others are available for download and installation. Once installed, they have to be loaded into the workspace to be used.

```
library() #List of all packages installed  
search() #List of all packages currently loaded  
install.packages("XYZ") #Install the package "XYZ"  
library(XYZ), require(XYZ) #Load the package "XYZ" to your workspace  
detach(package:XYZ) #Detach package "XYZ" when no longer needed
```

Introduction to R

- **Basic data types:**

1. character
2. numeric
3. integer
4. logical

character > double > integer > logical
--

- **Basic data structure:**

1. vector: A one-dimensional object. All elements must be of the same data type.
2. matrix: A two-dimensional object. All elements must be of the same data type.
3. array: A three-dimensional object. All elements must be of the same data type.
4. dataframe: A two-dimensional object. Each column must be of the same data type, but data type may vary by column Use [as.data.frame\(\)](#) to convert a matrix to dataframe.
5. list: A set of objects. Each element in a list can be of any structures.

Introduction to R

Use `[]` and `[[[]]` to index an element within objects:

1. vector: `[i]` for the *i*th element
2. matrix: `[i,j]` for the *i*th row, *j*th column
3. array: `[i,j,k]` for the *i*th row, *j*th column, *k*th level
4. dataframe: `[i,j]` for the *i*th row, *j*th column
5. list: `[[i]]` for the *i*th element

Introduction to R

- Some functions to examine R objects:
 1. `class()` #determine the structure of an object
 2. `typeof()` #determine the type of an object
 3. `str()` #display the structure of an object
 4. `ls()` #to see all the objects in your environment

Introduction to R

Writing a function in R

- A function is an object containing a sequence of statements that are run in a predefined order.
- Writing a function is primarily done to avoid having to repeat the same block of code multiple times. This eliminates the possibility of errors from copy-pasting and makes the code more comprehensible.
- Some built-in functions: `sum()`, `mean()`, `sd()`, `var()`, `min()`, `max()`

```
function_name <- function(input){
```

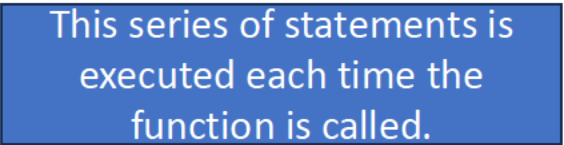
```
#statement
```

```
#statement
```

```
return(output)
```

```
}
```

```
function_name() #to call the function
```



This series of statements is executed each time the function is called.

Introduction to R

To load a dataset.:

- If the dataset is in an existing R package, load the package and use `data()`
- dataset in .RData format, use `load()`
- dataset in .txt or other text formats, use `read.table()`
- dataset in .csv format, use `read.csv()`

To save an object into:

- .RData format: `save()`
- .txt: `write.table()`
- .csv: `write.csv()`

Introduction to R

Plotting

`plot()`

`barplot()`

`pie()`

`boxplot()`

`hist()`

Reference

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI magazine*, 17(3), 37-37.