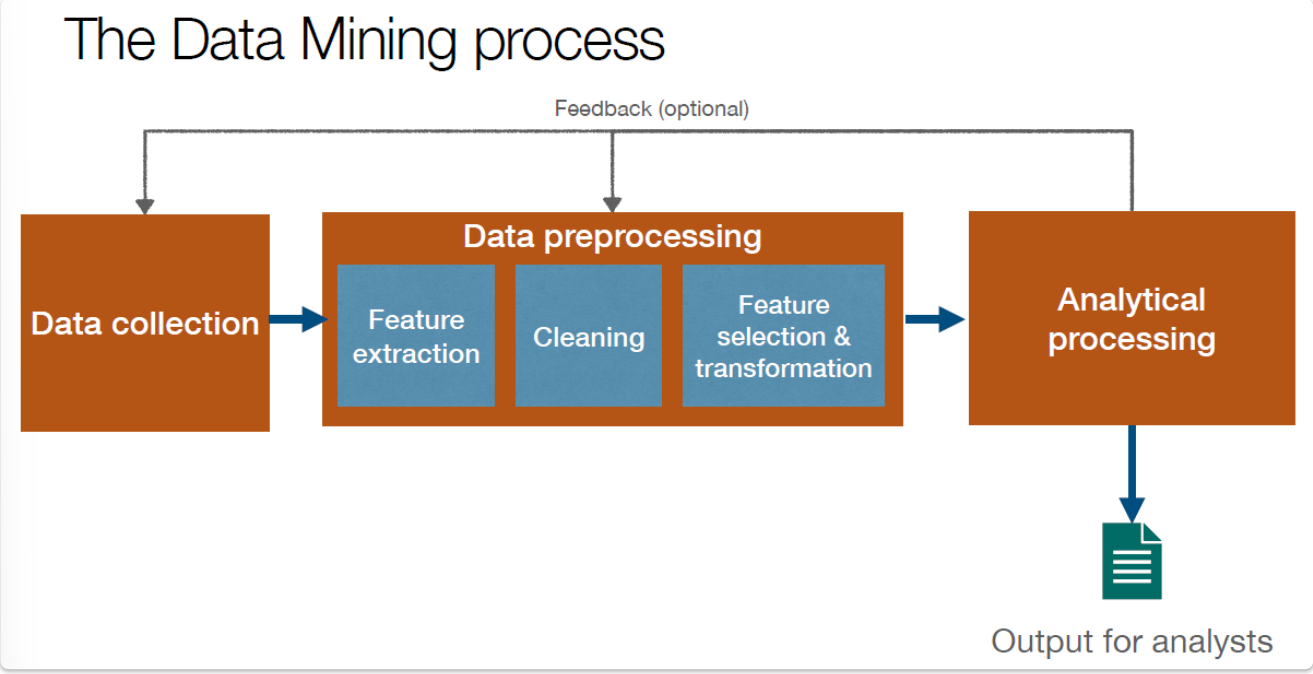
lec01 Introduction to Data Mining

Data Mining Process

1. Data collection (数据收集)
2. Data preprocessing (数据预处理)
3. Analytical processing (分析处理)



Data collection phase

- Highly application-specific (高度依赖于应用):
  - Sensor networks (传感器网络)
  - User surveys (用户调查)
  - Automatically collected documents (自动收集的文档)
- Critically important
  May significantly impact the whole data mining process
- Output of this phase are stored in a database of a data warehouse (仓库)

Data preprocessing phase

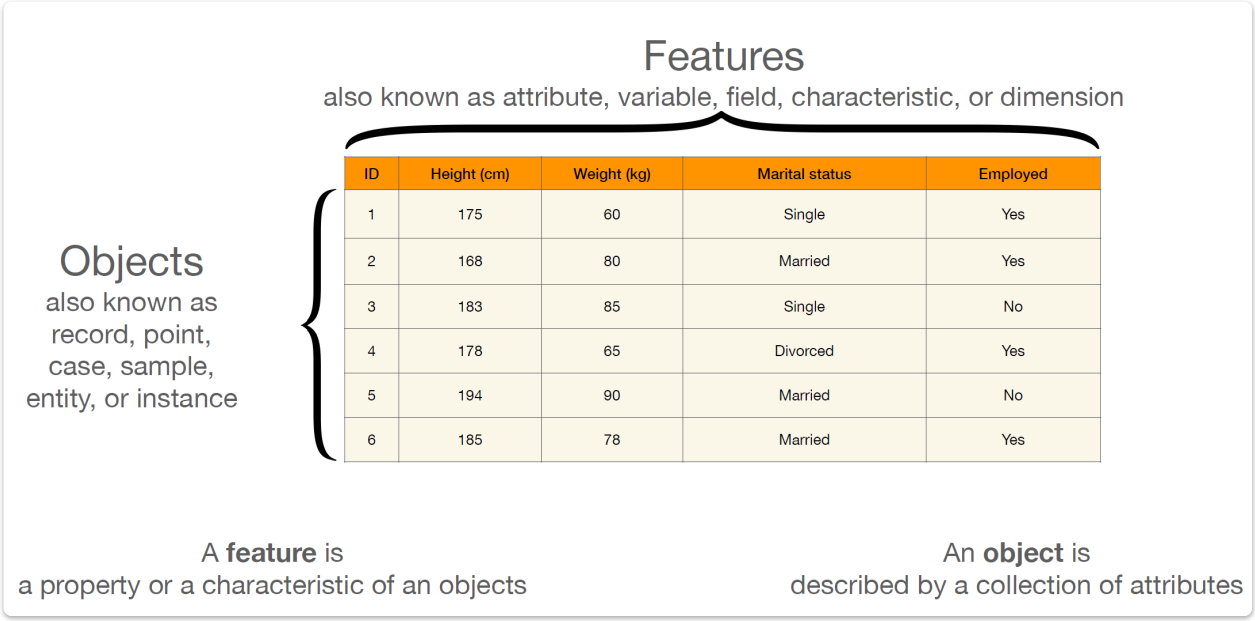
- Feature extraction (特征提取)

$$\text{Raw data} \rightarrow \text{Relevant feature} \rightarrow \text{Algo-friendly format}$$

  Convert data into a format that is friendly to data mining algorithms (将数据转换为对数据挖掘算法友好的格式).
  Below are the common methods

  - Multidimensional
    Every data point (object) is characterised by a sequence of fields

- Time series
- Semi-structured format



ID	Height (cm)	Weight (kg)	Marital status	Employed
1	175	60	Single	Yes
2	168	80	Married	Yes
3	183	85	Single	No
4	178	65	Divorced	Yes
5	194	90	Married	No
6	185	78	Married	Yes

Features:

- Features are central to data mining
- Abstract data points and learn rules → predict things for unseen/future data points

> *We learn that* `if colour == red then flower = rose`
> *We can use this rule to classify not only flowers in our train dataset, but also all flowers including ones that we do not have in our train dataset*

- Coming up of good features is an art

> *feature-engineering*
> *https://en.wikipedia.org/wiki/Feature_engineering*

- Recent work in machine learning such as deep learning focus on *automatically* discovering good features from data, *without* any human intervention

- Data cleaning (数据清理)
  Extracted data may have erroneous or missing fields. The missing and erroneous parts of the data are either estimated or corrected (提取的数据可能有错误或缺失字段，我们需要估计或纠正数据的缺失和错误部分).
  Common methods:
  - Drop a record
  - Estimate a missing value
  - Remove inconsistencies (消除不一致之处)
- Feature selection and transformation (特征选择与转换)
  Many data mining algorithms do not work efficiently on high dimensional data.
  Common methods:
  - Identify and remove irrelevant features (识别并删除不相关的特征)
  - Transform existing features to features of different scale or format (将现有特征转换为不同规模或格式的特征)
  - For data transformation: Transform attributes to new attributes (e.g., numerical age → {young, middle-aged, elderly})

Analytical processing phase

- Design and apply analytical methods to the preprocessed data (设计分析方法并将其应用于预处理数据)
- Break the problem into subproblems of 4 main types (将问题分解为 4 种主要类型的子问题):
  - Association pattern mining (关联模式挖掘/关联规则挖掘)
  - Clustering (聚类)
  - Classification (分类)
  - Outlier detection (异常检测)

Types of Data

Two general classes of data

- Non-dependency-oriented data (非依赖型数据): Objects do not have dependencies
- Dependency-oriented data (面向依赖的数据): Implicit or explicit dependencies between objects may exist
  - Networks: nodes (objects) are connected by edges (relationships)
  - Successive measurements collected from a sensor (从传感器收集的连续测量值)

Non-dependency-oriented data (multidimensional data)

It's the simplest form of data.

For a multidimensional data set $\mathscr{D}$ typically contains a set of records $\bar{X}_1, \ldots, \bar{X}_n$. Each record $\bar{X}_i$ contains a set of d features $(x_i^1, \ldots, x_i^d)$. This data set can be represented by an $n \times d$ data matrix

$$\begin{pmatrix} x_1^1 & x_1^2 & \cdots & x_1^d \\ x_2^1 & x_2^2 & \cdots & x_2^d \\ \vdots & \vdots & \ddots & \vdots \\ x_n^1 & x_n^2 & \cdots & x_n^d \end{pmatrix}$$

Types:

- Numerical or quantitative (values have natural ordering, includes both integers and real values)
- Categorical or unordered discrete-valued (离散的无序值/类)
- Binary data: 可以看作是两个类别的分类数据 (categorical data) 或者数值数据 (numerical data)，同时可用于通过特征向量 (characteristic vectors) 表示集合数据 (set data)
- Text data
  - Document as a string (dependency-oriented data type (面向依赖性的数据类型))
  - Document as a set of words or terms (vector-space representation: frequencies of the words in the document (向量空间表示：文档中单词的频率))

Dependency-oriented data

- Implicit dependencies:
  - Are not explicitly specified but are known to exist (未明确指定但已知存在). (e.g. temperature values collected by a sensor).
  - Contextual attributes (上下文属性): determine implicit dependencies in the current context

    - Behavioral attributes (行为属性): represent the values that are measured in a particular context
- Explicit dependencies: Graphs or network data (edges specify explicit relationships (边指定显式关系))

Types:

- Implicit
    - Time-series: values that are generated by sequential measurements over time. Time-stamp or index value is a contextual attribute; the measurement is behavioral attribute (随着时间的推移通过连续测量生成的值. 时间戳或索引值是上下文属性；测量是行为属性)
    - Discrete Sequences and Strings (离散序列和字符串): The categorical analog of time-series data (时间序列数据的分类模拟)
    - Spatial data (空间数据): every record has a location attribute (每条记录都有一个位置属性). e.g. temperature, pressure are measured at spatial locations
    - Spatiotemporal data (时空数据): contain both spatial and temporal attributes (包含空间和时间属性)

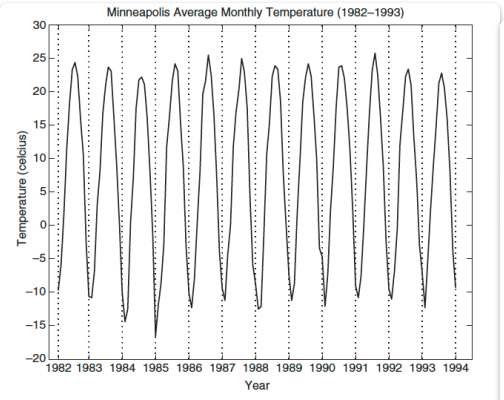
Time	Customer	Items Purchased
t1	C1	A, B
t2	C3	A, C
t2	C1	C, D
t3	C2	A, D
t4	C2	E
t5	C1	A, E

Customer	Time and Items Purchased
C1	(t1: A,B) (t2:C,D) (t5:A,E)
C2	(t3: A, D) (t4: E)
C3	(t2: A, C)

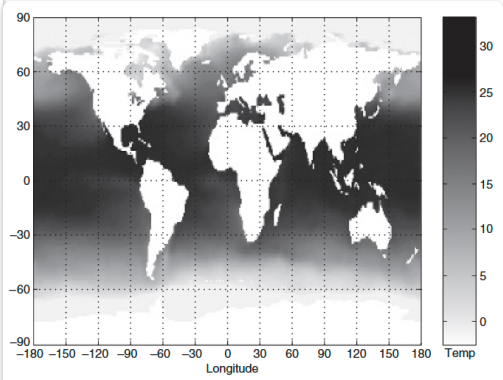
(a) Sequential transaction data.

```
GGTTCCGCCTTCAGCCCCGCGCC
CGCAGGGCCCGCCCCGCGCCGTC
GAGAAGGGCCCGCCTGGCGGGCG
GGGGGAGGCGGGGCCGCCCGAGC
CCAACCGAGTCCGACCAGGTGCC
CCCTCTGCTCGGCCTAGACCTGA
GCTCATTAGGCGGCAGCGGACAG
GCCAAGTAGAACACGCGAAGCGC
TGGGCTGCCTGCTGCGACCAGGG
```

(b) Genomic sequence data.



(c) Temperature time series.



(d) Spatial temperature data.

- Explicit: Network/Graph data (e.g. Web graph, Facebook/Instagram/LinkedIn social networks)
    - Objects correspond to nodes of the network (对象对应于网络的节点)
    - Relationships between the objects correspond to the edges of the network (对象之间的关系对应于网络的边缘)
    - Edges may be directed or undirected (边可以是有向的也可以是无向的)
    - A set of attributes may be associated with a node/an edge (一组属性可以与一个节点/一条边相关联)

Data Representation

Data representation is one of the first things we must do in data mining. What we can mine is largely determined by our data representation. There is no one best data representation method for all data mining tasks.

For example, unsuitable (无论) data representations will lead to poor classification performance irrespective of the classification algorithm.

Numerical data

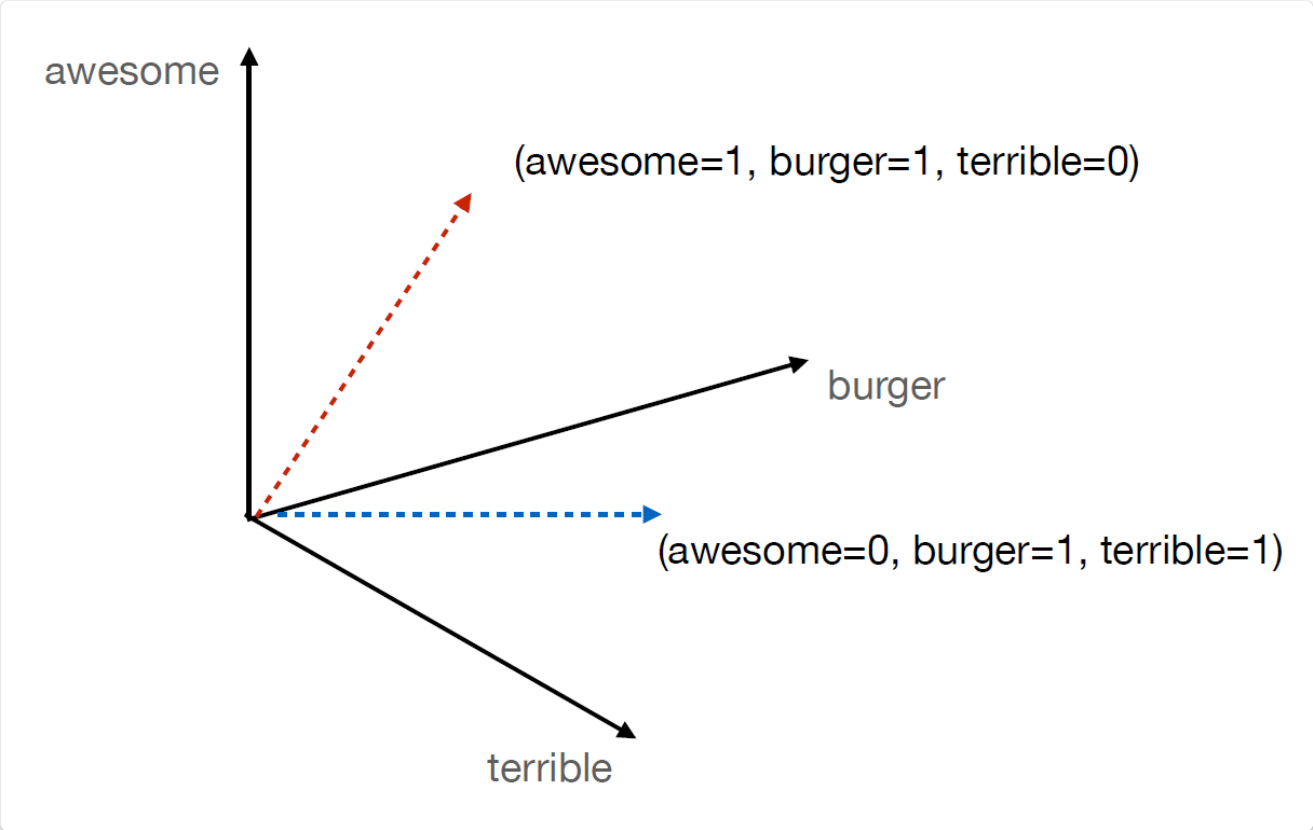
There is a clearly defined ordering among the values and the algebraic operations are well-defined.

Many machine learning algorithms assume you have your data points represented in $d-\textbf{dimensional}$ real space: $\bar{X}=(x_1,x_2,\ldots,x_d)\in\mathbb{R}^d$

Challenges when handing numerical data: Different features will be in different ranges like: $\textbf{height}\in[110,230]$ (cm), $\textbf{weight}\in[40,120]$ (km)

Categorical data

If you must represent categorical data, then you could do so by representing each category as a separate dimension of a vector (将每个类别表示为向量的单独维度来实现).



For example: we want to represent the sentence "The burger I ate was an awesome burger!"

- By a list of words:
  `["the", "burger", "i", "ate", "was", "an", "awesome", "burger"]`
- By the set of words:
  `{"the", "burger", "i", "ate", "was", "an", "awesome"}`
- By a vector of word frequency
  `{"the": 1, "burger": 2, "i": 1, "ate": 1, "was": 1, "an": 1, "awesome": 1}`
- By a vector of letter frequency
  `{'a': 3, ' ': 7, 'b': 2, 'e': 6, 'g': 2, 'i': 2, 'h': 1, 'm': 1, 'o': 1, 'n': 1,`

```
's': 2, 'r': 4, 'u': 2, 't': 2, 'w': 1}
```

Feature pruning (特征剪枝)

- There might be many irrelevant features. (i.e. features that are completely uncorrelated with the prediction)
- Can an algorithm at hand deal well with redundant features?
- Throw away irrelevant feature?
  - Irrelevant features for text data: a word appear either almost always or almost never (单词几乎总是出现或几乎从不出现)
  - Irrelevant features for numerical data: low variance features (低方差特征)