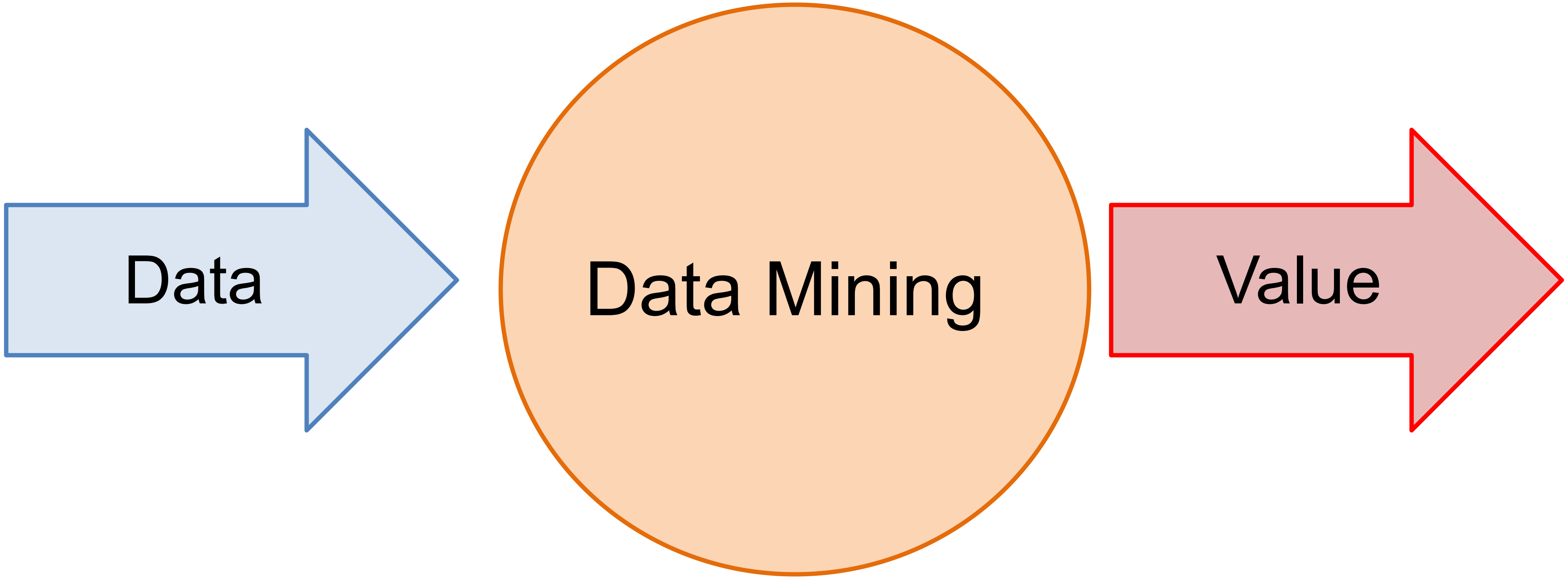
DATA MINING
THE DATA MINING PIPELINE

What is data?

The data mining pipeline: collection, preprocessing, mining, and post-processing

What is data mining again?

- “Data Mining is the study of collecting, processing, analyzing, and gaining useful insights from data” – Charu Aggarwal



Why do we need data mining?

- Really huge amounts of complex data generated from multiple sources and interconnected in different ways
  - Scientific data from different disciplines
    - Weather, astronomy, physics, biological microarrays, genomics
  - Huge text collections
    - The Web, scientific articles, news, tweets, facebook postings.
  - Transaction data
    - Retail store records, credit card records
  - Behavioral data
    - Mobile phone data, query logs, browsing behavior, ad clicks
  - Networked data
    - The Web, Social Networks, IM networks, email network, biological networks.
- We need to analyze this data to extract knowledge
  - Knowledge can be used for commercial or scientific purposes.
  - Our solutions should scale to the size of the data

DATA

What is Data?

- Collection of data objects (对象) and their attributes (属性)

- An attribute is a property or characteristic of an object
  - Examples: name, date of birth, height, occupation.
  - Attribute is also known as variable (变量), field (字段), characteristic (特性), or feature (特征)

- For each object the attributes take some values (值).

- The collection of attribute-value pairs describes a specific object
  - Object is also known as record (记录), point (数据点), case (案例), sample (样本), entity (实体), or instance (实例)

Attributes

Objects

Tid	**Refund**	**Marital Status**	**Taxable Income**	**Cheat**
1	Yes	Single	125K	**No**
2	No	Married	100K	**No**
3	No	Single	70K	**No**
4	Yes	Married	120K	**No**
5	No	Divorced	95K	**Yes**
6	No	Married	60K	**No**
7	Yes	Divorced	220K	**No**
8	No	Single	85K	**Yes**
9	No	Married	75K	**No**
10	No	Single	90K	**Yes**

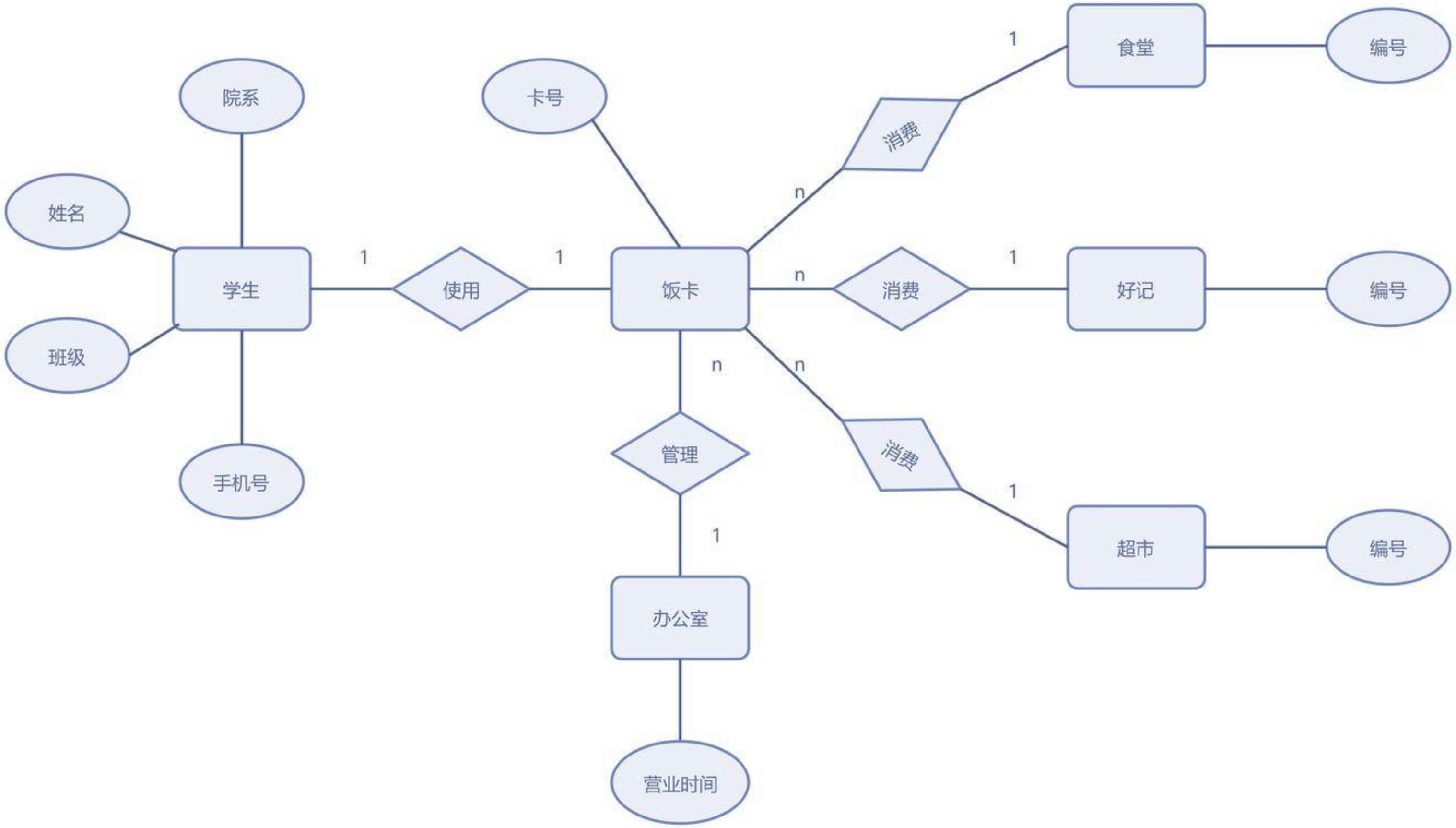
Size (n): Number of objects
Dimensionality (d): Number of attributes

Relational data

- The term comes from DataBases, where we assume data is stored in a relational table with a fixed schema (模式) (fixed set of attributes)

- 关系型数据库的基础是关系(relations)，实体之间有entity-relation图

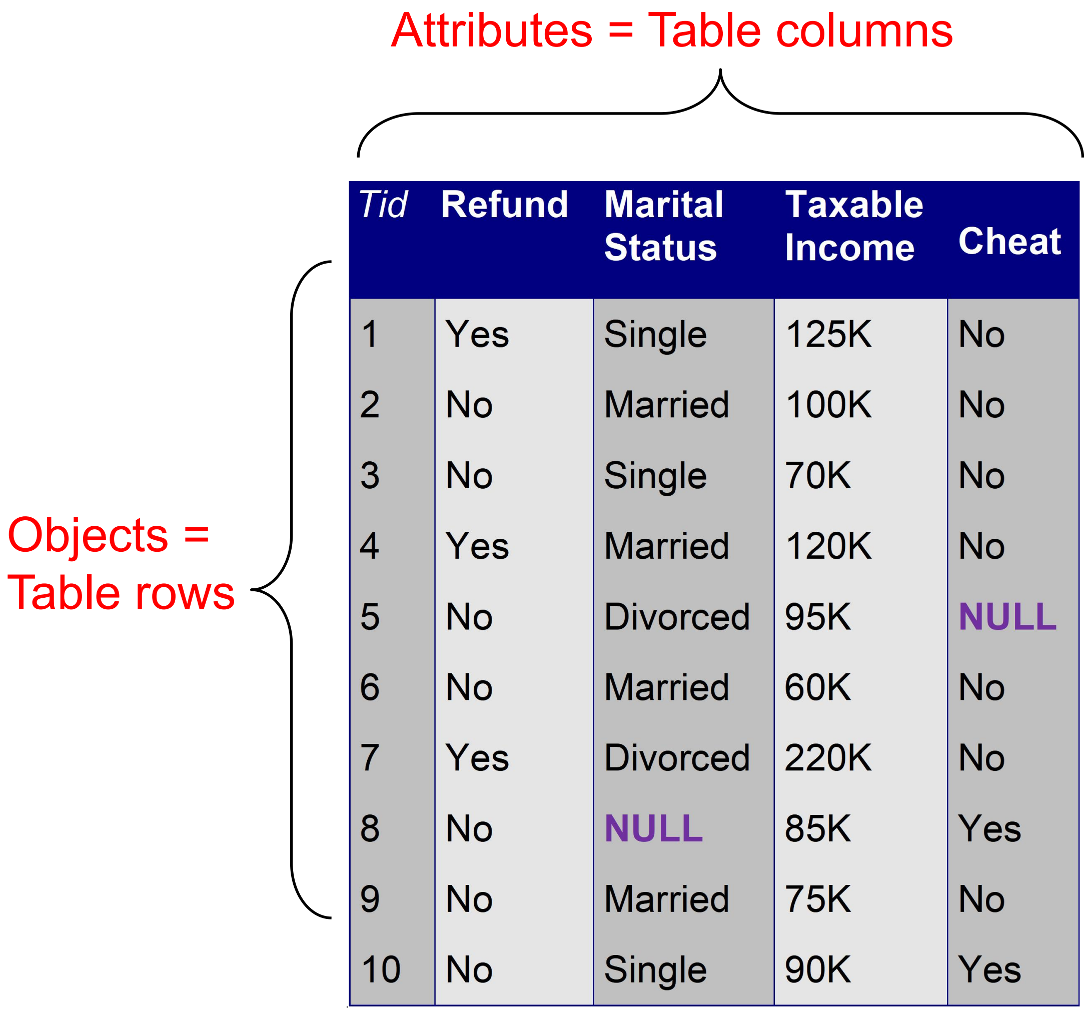
学生饭卡E-R图



Relational data

- The term comes from DataBases, where we assume data is stored in a relational table with a fixed schema (模式) (fixed set of attributes)
  - In Databases, it is usually assumed that the table is dense (紧密) (few null values)
- There are a lot of data in this form
  - E.g., census data, market transaction data
- There are also a lot of data which do not fit well in this form (in many cases, heterogeneous web data)
  - Sparse data: Many missing values
  - Not easy to define a fixed schema

Example of a relational table



Attributes = Table columns

Objects = Table rows

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	NULL
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	NULL	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Types of Attributes

- There are different types of attributes
  - Numeric （数值）
    - Can be compared and support calculations
    - Examples: dates, temperature, time, length, value, count.
    - Discrete (counts) vs Continuous (temperature)
    - Special case: Binary/Boolean attributes (yes/no, exists/not exists)
  - Categorical （分类）
    - Used for naming or labelling variables
    - Nominal (定类变量, without order) e.g. 民族：汉，藏，蒙 邮编：100000，200000
    - Ordinal (定序变量, with order) e.g. 受教育程度：博士，硕士，本科

Numeric Relational Data

- If data objects have the same fixed set of numeric attributes, then the data objects can be thought of as points/vectors in a multi-dimensional space, where each dimension represents a distinct attribute

- Such data set can be represented by an n-by-d data matrix, where there are n rows, one for each object, and d columns, one for each attribute

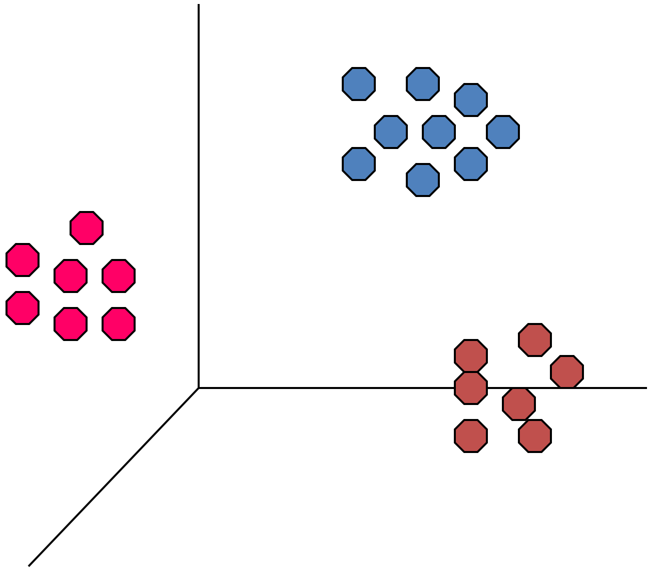
	Temperature	Humidity	Pressure
O1	30	0.8	90
O2	32	0.5	80
O3	24	0.3	95

30	0.8	90
32	0.5	80
24	0.3	95

Numeric data

- Thinking of numeric data as points or vectors is very convenient

- For small dimensions (e.g. 3D) we can plot the data
- We can use geometric analogues to define concepts like distance or similarity
- We can use linear algebra to process the data matrix
- We will often talk about points or vectors



Categorical Relational Data

- Data that consists of a collection of records, each of which consists of a fixed set of categorical attributes

ID Number	Zip Code	Marital Status	Income Bracket
1129842	45221	Single	High
2342345	45223	Married	Low
1234542	45221	Divorced	High
1243535	45224	Single	Medium

Mixed Relational Data

- Data that consists of a collection of records, each of which consists of a fixed set of both numeric and categorical attributes

ID Number	Zip Code	Age	Marital Status	Income	Income Bracket
1129842	45221	55	Single	250000	High
2342345	45223	25	Married	30000	Low
1234542	45221	45	Divorced	200000	High
1243535	45224	43	Single	150000	Medium

Mixed Relational Data

- Data that consists of a collection of records, each of which consists of a fixed set of both numeric and categorical attributes

ID Number	Zip Code	Age	Marital Status	Income	Income Bracket	Refund
1129842	45221	55	Single	250000	High	No
2342345	45223	25	Married	30000	Low	Yes
1234542	45221	45	Divorced	200000	High	No
1243535	45224	43	Single	150000	Medium	No

Mixed Relational Data

- Data that consists of a collection of records, each of which consists of a fixed set of both numeric and categorical attributes

Takes numerical values but it is actually categorical

ID Number	Zip Code	Age	Marital Status	Income	Income Bracket	Refund
1129842	45221	55	Single	250000	High	0
2342345	45223	25	Married	30000	Low	1
1234542	45221	45	Divorced	200000	High	0
1243535	45224	43	Single	150000	Medium	0

Boolean attributes can be thought as both numeric and categorical

Mixed Relational Data

- Sometimes it is convenient to represent categorical attributes as boolean.
  - Add a Boolean attribute for each possible value of the attribute

ID	Zip 45221	Zip 45223	Zip 45224	Age	Single	Married	Divorced	Income	Refund
1129842	1	0	0	55	0	0	0	250000	0
2342345	0	1	0	25	0	1	0	30000	1
1234542	1	0	0	45	0	0	1	200000	0
1243535	0	0	1	43	0	0	0	150000	0

We can now view the whole vector as numeric

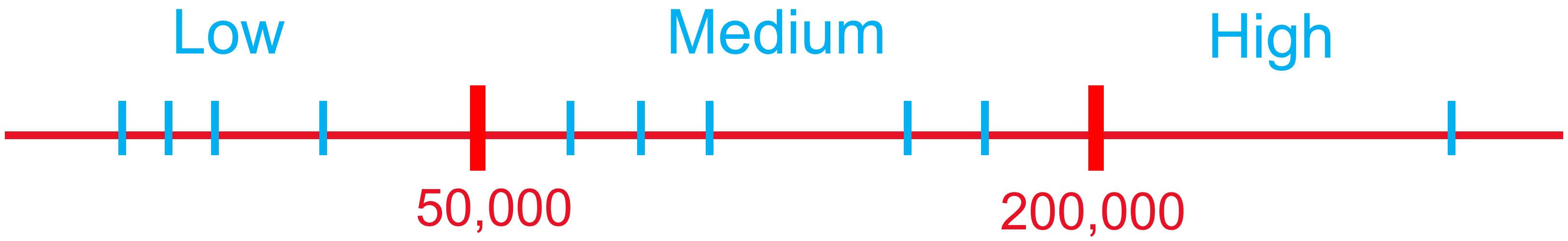
Mixed Relational Data

- Sometimes it is convenient to represent numerical attributes as categorical.
  - Group the values of the numerical attributes into bins （箱）

ID Number	Zip Code	Age	Marital Status	Income	Income Bracket	Refund
1129842	45221	50s	Single	High	High	0
2342345	45223	20s	Married	Low	Low	1
1234542	45221	40s	Divorced	High	High	0
1243535	45224	40s	Single	Medium	Medium	0

Binning（分箱）

- Idea: split the range of the domain of the numerical attribute into bins (intervals).
- Every bucket defines a categorical value

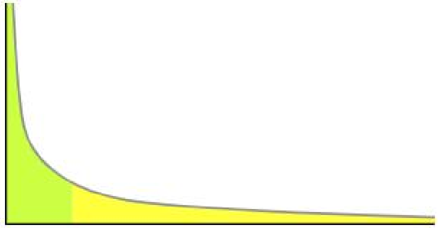


- How do we decide the number of bins?
  - Depends on the granularity of the data that we want

Bucketization

- How do we decide the size of the bucket?
  - Depends on the data and our application
- Equi-width bins: All bins have the same size, i.e. end – start is constant
  - Example: split time into decades
  - Problem: some bins may be very sparse or empty
- Equi-size (depth) bins: Select the bins so that they all contain the same number of elements
  - This splits data into quantiles: top-10%, second 10% etc
  - Some bins may be very small
- Equi-log bins:$\log end - \log start$ is constant
  - The size of the previous bin is a fraction of the current one
  - E.g. range = [2, 1024], base=2, $\log end - \log start$=1, log ($\frac{end}{start}$)=1 -> $\frac{end}{start}$=2

  2,4,8,16,32, 64, 128, 256, 512,1024
  - Better for skewed distributions (e.g. power law distribution幂率分布)
- Optimized bins: Use a 1-dimensional clustering algorithm to create the bins



Physical data storage

- Stored in a Relational Database
  - Assumes a strict schema and relatively dense data (few missing/Null values)
- Tab or Comma separated files (TSV/CSV), Excel sheets
  - Assumes a strict schema and relatively dense data (few missing/Null values)
- Flat file with triplets (record id, attribute, attribute value)
  - A very flexible data format, allows multiple values for the same attribute (e.g., phone number)
- JSON, XML format
  - Standards for data description that are more flexible than relational tables
  - There exist parsers for reading such data.

Examples

Comma Separated File

```
id,Name,Surname,Age,Zip
1,John,Smith,25,10021
2,Mary,Jones,50,96107
3,Joe,Doe,80,80235
```

- Can be processed with simple parsers, or loaded to excel or a database

Triple-store

```
1, Name, John
1, Surname, Smith
1, Age, 25
1, Zip, 10021
2, Name, Mary
2, Surname, Jones
2, Age, 50
2, Zip, 96107
3, Name, Joe
3, Surname, Doe
3, Age, 80
3, Zip, 80235
```

- Easy to deal with missing values

Examples

JSON EXAMPLE – Record of a person

```
{
  "firstName": "John",
  "lastName": "Smith",
  "isAlive": true,
  "age": 25,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021-3100"
  },
  "phoneNumbers": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "office",
      "number": "646 555-4567"
    }
  ],
  "children": [],
  "spouse": null
}
```

XML EXAMPLE – Record of a person

```
<person>
  <firstName>John</firstName>
  <lastName>Smith</lastName>
  <age>25</age>
  <address>
    <streetAddress>21 2nd
Street</streetAddress>
    <city>New York</city>
    <state>NY</state>
    <postalCode>10021</postalCode>
  </address>
  <phoneNumbers>
    <phoneNumber>
      <type>home</type>
      <number>212 555-1234</number>
    </phoneNumber>
    <phoneNumber>
      <type>fax</type>
      <number>646 555-4567</number>
    </phoneNumber>
  </phoneNumbers>
  <gender>
    <type>male</type>
  </gender>
</person>
```

Beyond relational data: Set data

- Each record is a set of items from a space of possible items
- Example: Transaction data
  - Also called market-basket data

TID	Items
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

Set data

- Each record is a set of items from a space of possible items
- Example: Document data
    - Also called bag-of-words representation

Doc Id	Words
1	the, dog, followed, the, cat
2	the, cat, chased, the, cat
3	the, man, walked, the, dog

Vector representation of market-basket data

- Market-basket data can be represented, or thought of, as numeric vector data
  - The vector is defined over the set of all possible items
  - The values are binary (the item appears or not in the set)

TID	Items
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

TID	Bread	Coke	Milk	Beer	Diaper
1	1	1	1	0	0
2	1	0	0	1	0
3	0	1	1	1	1
4	1	0	1	1	1
5	0	1	1	0	1

Sparsity: Most entries are zero. Most baskets contain few items

Vector representation of document data

- Document data can be represented, or thought of, as numeric vector data
  - The vector is defined over the set of all possible words
  - The values are the counts (number of times a word appears in the document)

Doc Id	Words
1	the, dog, follows, the, cat
2	the, cat, chases, the, cat
3	the, man, walks, the, dog

Doc Id	the	dog	follows	cat	chases	man	walks
1	2	1	1	1	0	0	0
2	2	0	0	2	1	0	0
3	1	1	0	0	0	1	1

Sparsity: Most entries are zero. Most documents contain few of the words

Physical data storage

- Usually set data is stored in flat files
  - One line per set

```
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29
30 31 32
33 34 35
36 37 38 39 40 41 42 43 44 45 46
38 39 47 48
38 39 48 49 50 51 52 53 54 55 56 57 58
32 41 59 60 61 62
3 39 48
```

- I heard so many good things about this place so I was pretty juiced to try it. I'm from Cali and I heard Shake Shack is comparable to IN-N-OUT and I gotta say, Shake Shake wins hands down.   Surprisingly, the line was short and we waited about 10 MIN. to order. I ordered a regular cheeseburger, fries and a black/white shake. So yummerz.   I love the location too! It's in the middle of the city and the view is breathtaking.   Definitely one of my favorite places to eat in NYC.
- I'm from California and I must say, Shake Shack is better than IN-N-OUT, all day, err'day.

Dependent data

- In tables we usually consider each object independent of each other.
- In some cases, there are explicit dependencies between the data
  - Ordered/Temporal data: We know the time order of the data
  - Spatial data: Data that is placed on specific locations
  - Spatiotemporal data: data with location and time
  - Networked/Graph data: data with pairwise relationships between entities

Ordered Data

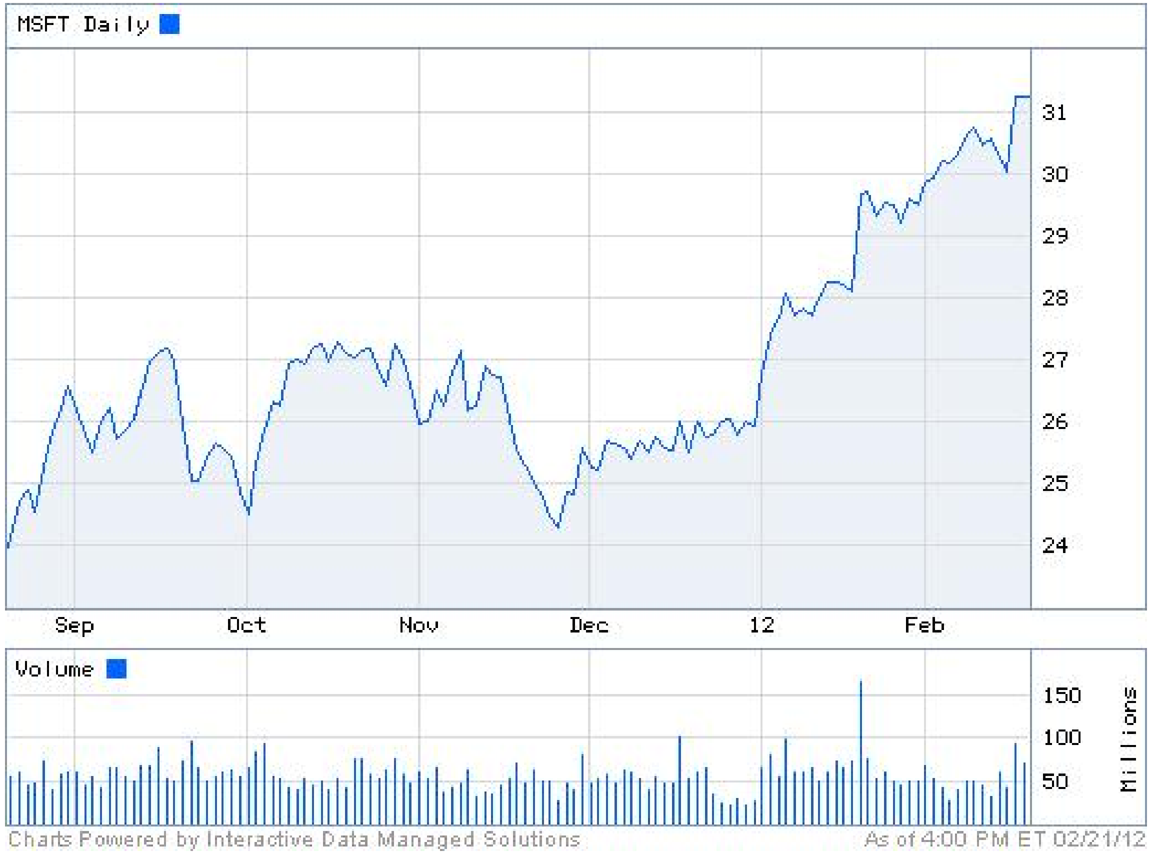
- Genomic sequence data

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

- Data is a long ordered string

Ordered Data

- Time series
  - Sequence of ordered (over "time") numeric values.



Ordered Data

- Sequence data: Similar to the time series but in this case we have categorical values rather than numerical ones.
- Example: Event logs

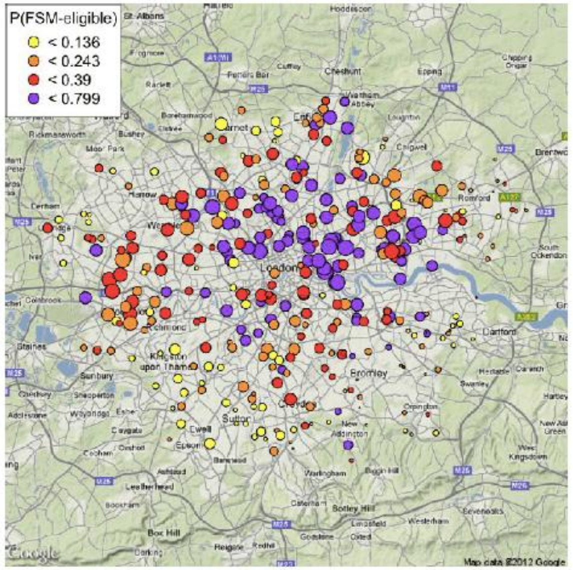
```
fcrawler.looksmart.com - - [26/Apr/2000:00:00:12 -0400] "GET /contacts.html HTTP/1.0" 200 4595 "-" "FAST-WebCraw
fcrawler.looksmart.com - - [26/Apr/2000:00:17:19 -0400] "GET /news/news.html HTTP/1.0" 200 16716 "-" "FAST-WebCr

ppp931.on.bellglobal.com - - [26/Apr/2000:00:16:12 -0400] "GET /download/windows/asctab31.zip HTTP/1.0" 200 1540

123.123.123.123 - - [26/Apr/2000:00:23:48 -0400] "GET /pics/wpaper.gif HTTP/1.0" 200 6248 "http://www.jafsoft.co
123.123.123.123 - - [26/Apr/2000:00:23:47 -0400] "GET /asctortf/ HTTP/1.0" 200 8130 "http://search.netscape.com/
123.123.123.123 - - [26/Apr/2000:00:23:48 -0400] "GET /pics/5star2000.gif HTTP/1.0" 200 4005 "http://www.jafsoft
123.123.123.123 - - [26/Apr/2000:00:23:50 -0400] "GET /pics/5star.gif HTTP/1.0" 200 1031 "http://www.jafsoft.com
123.123.123.123 - - [26/Apr/2000:00:23:51 -0400] "GET /pics/a2hlogo.jpg HTTP/1.0" 200 4282 "http://www.jafsoft.c
123.123.123.123 - - [26/Apr/2000:00:23:51 -0400] "GET /cgi-bin/newcount?jafsof3&width=4&font=digital&noshow HTTP
```

Spatial data

- Attribute values that can be arranged with geographic co-ordinates
  - Satellite data
  - Photos/Weibo with GPS locations
  - Web visits/search with IP addresses that can be mapped to locations
- Such data can be nicely visualized.



Spatiotemporal data

- Data that have both spatial and temporal aspects
  - Measurements in different locations over time
    - Pressure, Temperature, Humidity
  - Measurements that move in space over time
    - Traffic, Trajectories of moving objects

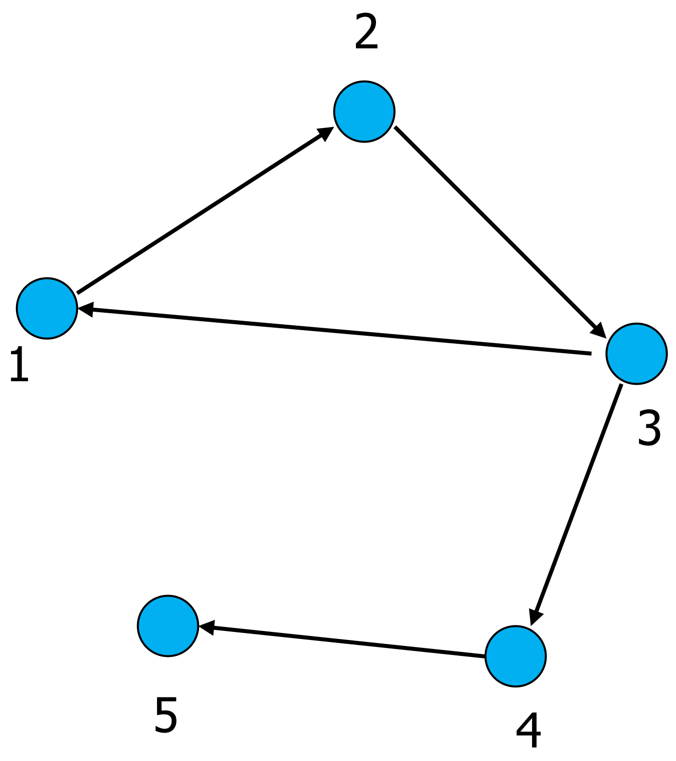
Graph Data

- Graph data: a collection of entities and their pairwise relationships.
- Examples:
  - Web pages and hyperlinks
  - Facebook users and friendships
  - The connections between brain neurons
  - Genes that regulate each other

In this case the data consists of pairs:

Who links to whom

We may have directed links



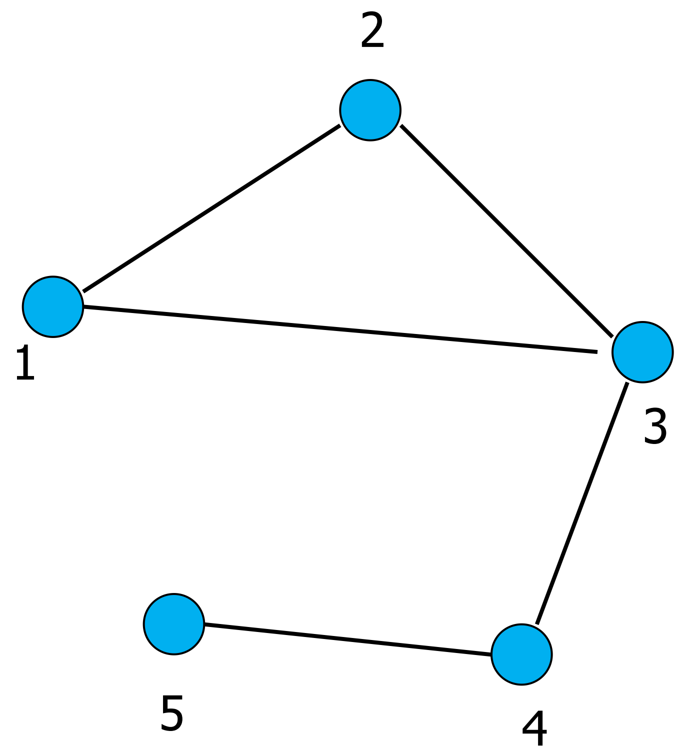
Graph Data

- Graph data: a collection of entities and their pairwise relationships.
- Examples:
  - Web pages and hyperlinks
  - Facebook users and friendships
  - The connections between brain neurons
  - Genes that regulate each other

In this case the data consists of pairs:

Who links to whom

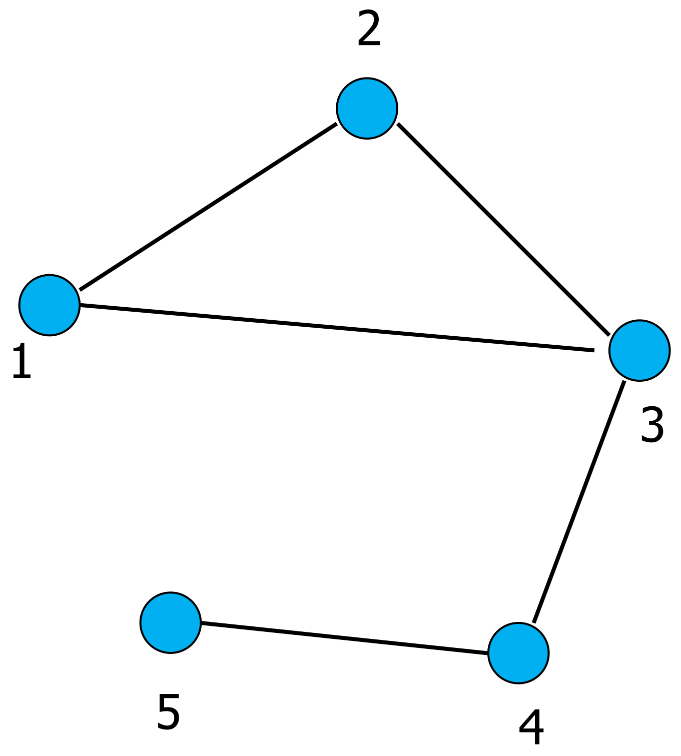
Or undirected links



Representation

- Adjacency matrix
  - Very sparse, very wasteful, but useful conceptually

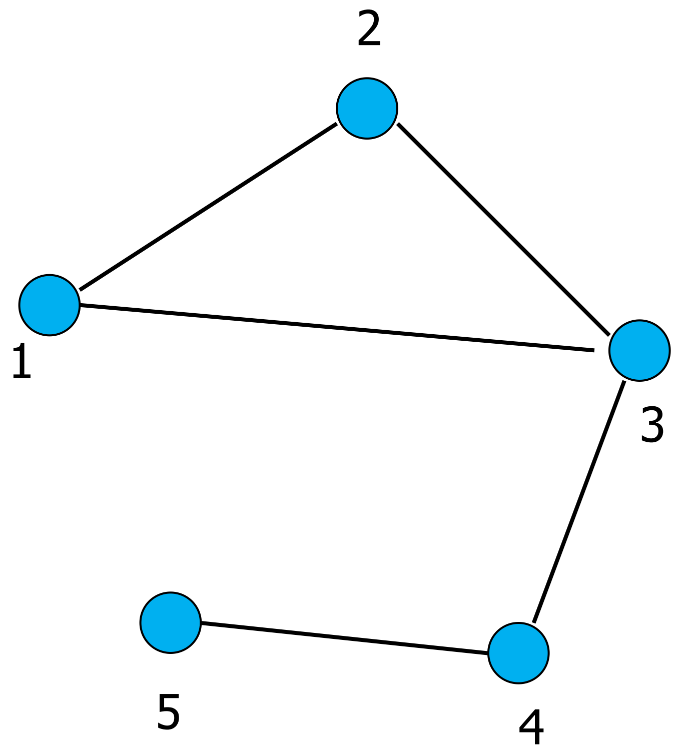
$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$



Representation

- Adjacency list
  - Not so easy to maintain

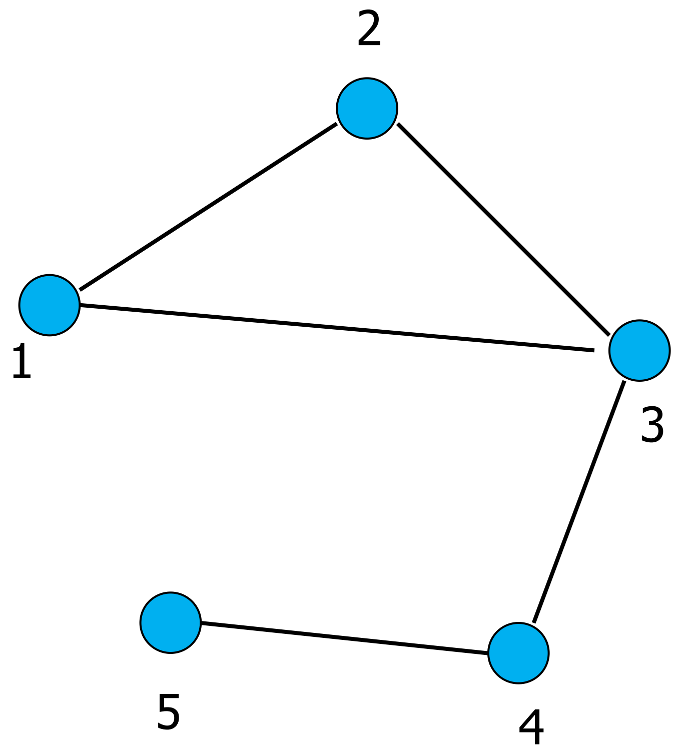
1: [2, 3]
2: [1, 3]
3: [1, 2, 4]
4: [3, 5]
5: [4]



Representation

- List of pairs
  - The simplest and most efficient representation

(1,2)
(2,3)
(1,3)
(3,4)
(4,5)



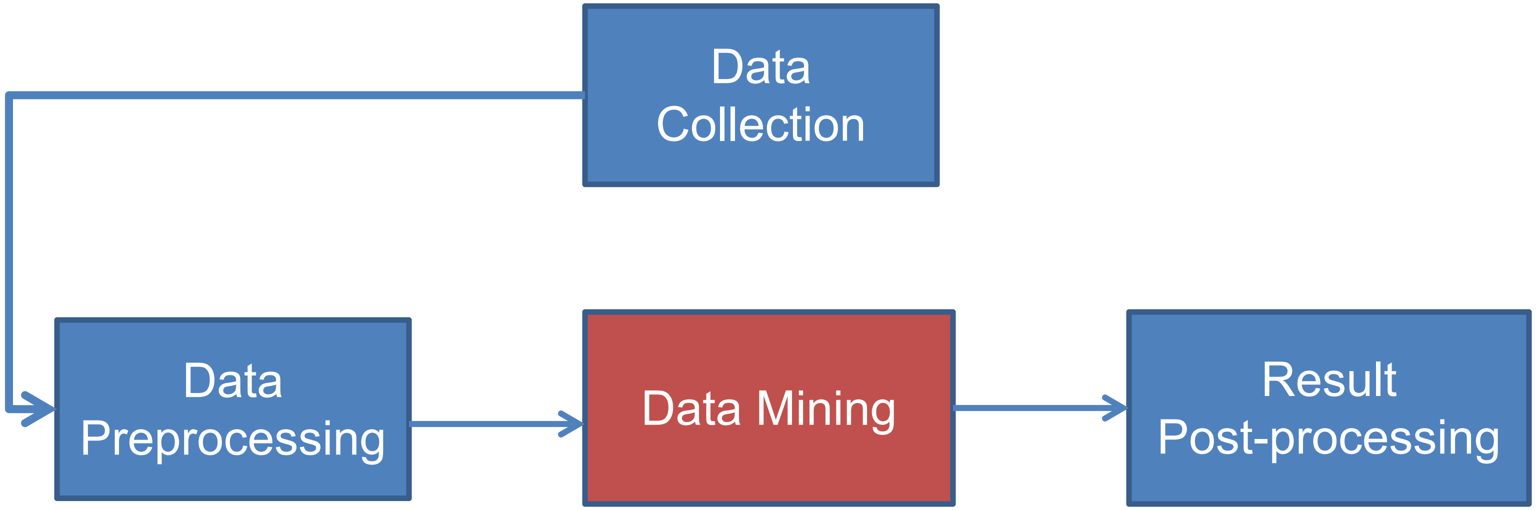
Types of data: summary

- Numeric data: Each object is a point in a multidimensional space
- Categorical data: Each object is a vector of categorical values
- Set data: Each object is a set of values (with or without counts)
  - Sets can also be represented as binary vectors, or vectors of counts
- Dependent data:
  - Ordered sequences: Each object is an ordered sequence of values.
  - Spatial data: objects are fixed on specific geographic locations
  - Graph data: A collection of pairwise relationships

DATA MINING PIPELINE

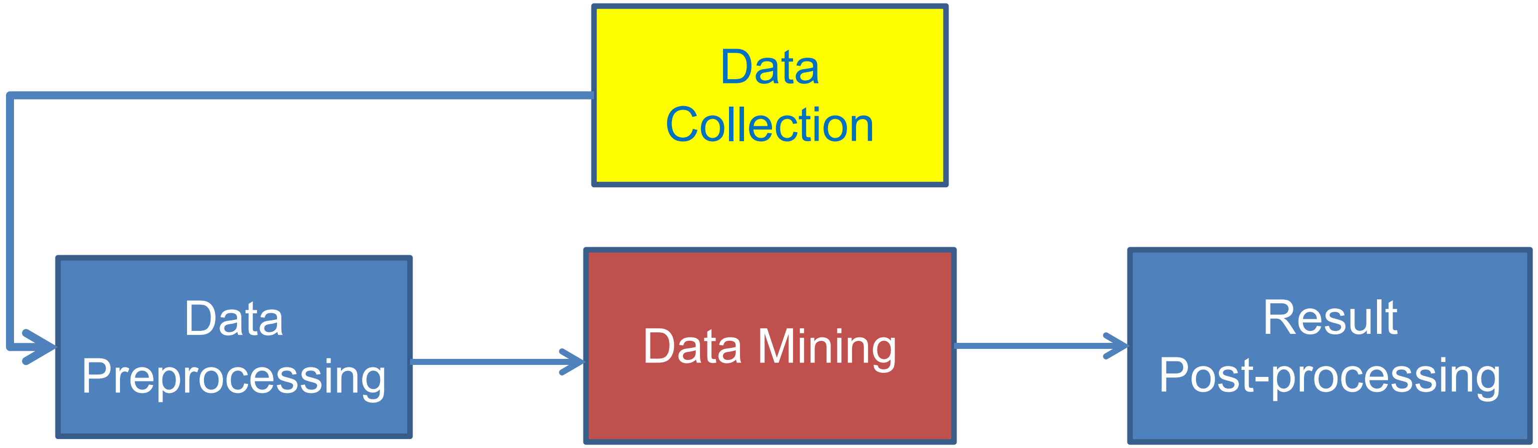
The data analysis pipeline

Mining is not the only step in the analysis process



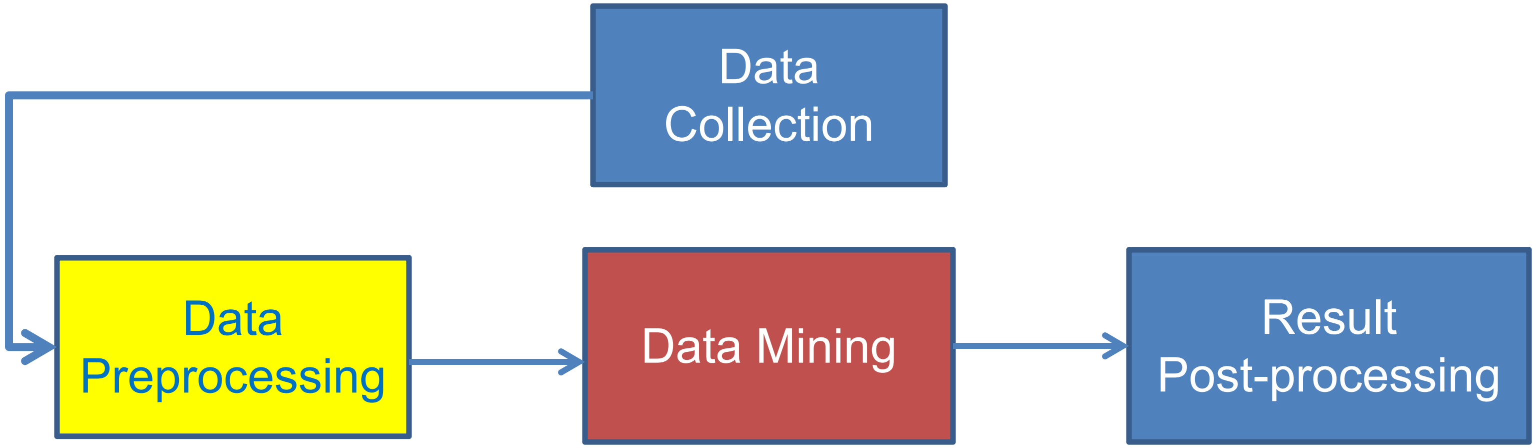
The data mining part is about the analytical methods and algorithms for extracting useful knowledge from the data.

The data analysis pipeline



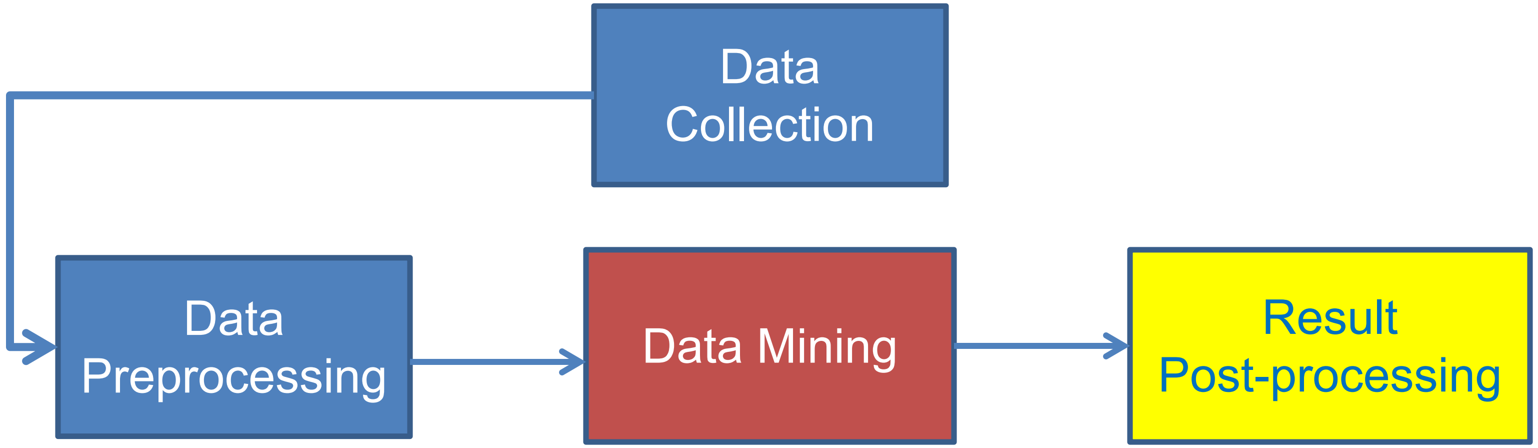
- Today there is an abundance of data online (Twitter, Wikipedia, Web, Open data initiatives, etc)
- Collecting the data is a separate task
  - Customized crawlers, use of public APIs. Respect of crawling etiquette
- Which data should we collect?
  - We cannot necessarily collect everything so we need to make some choices before starting.
- How should we store them?
- In many cases when collecting data we also need to label them
  - E.g., how do we identify fraudulent transactions?
  - E.g., how do we elicit user preferences?

The data analysis pipeline



- Preprocessing: Real data is large, noisy, incomplete and inconsistent.
  - Reducing the data: Sampling, Dimensionality Reduction
  - Data cleaning: deal with missing or inconsistent information
  - Feature extraction and selection: create a useful representation of the data by extracting useful features
- The preprocessing step determines the input to the data mining algorithm
  - A dirty work, but someone has to do it.
  - It is often the most important step for the analysis

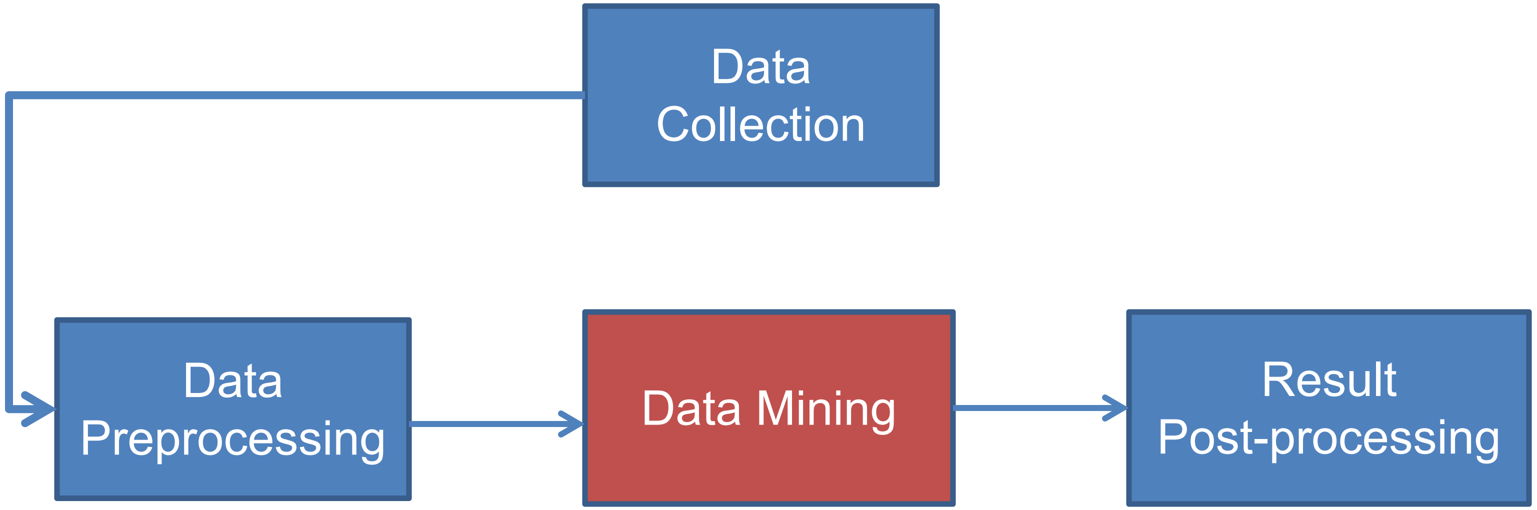
The data analysis pipeline



- Post-Processing: Make the data actionable and useful to the user
  - Statistical analysis of importance of results
  - Visualization

The data analysis pipeline

Mining is not the only step in the analysis process



- Pre- and Post-processing are often data mining tasks as well