上海外国语大学
SHANGHAI INTERNATIONAL STUDIES UNIVERSITY

New Media
Data Analytics and Application

Lecture 10: Text Mining and Data Visualization

Ting Wang

Outlines

- Text Mining
- Data Visualization using Python
- Data Mining Essentials

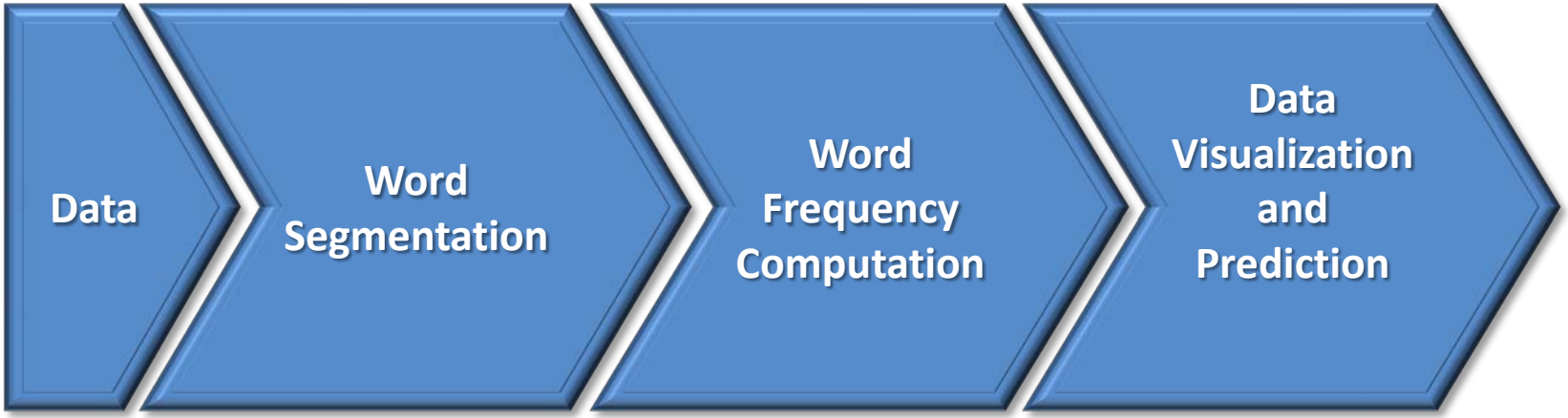
上海外国语大学
SHANGHAI INTERNATIONAL STUDIES UNIVERSITY

online text data mining based on natural language processing

Text Mining

Text Mining

Now, we have data, how to mining it?



Case Description

Motivations:

- To measure a news objectively
- To obtain new information efficiently

Methodologies:

- Describe a news report by quantitative method
- Technical integration by computer science, statistics and journalism

Text Mining

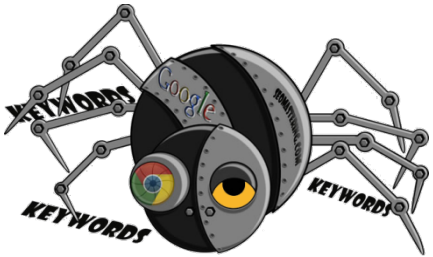
Steps:

1. Download a news report
2. Word segmentation
3. Word tag extraction and statistical computing
4. Data visualization and news summarization

Step 1: Download a News Report

- Example: http://news.sina.com.cn/c/nd/2016-12-10/doc-ifxypipt0842414.shtml

News also can be obtained by web crawler or databases



Step 2: Word segmentation (1)

Database Preparation

- Word Dictionary (required)
- Stop Word Dictionary (required)
- Dictionaries of Terms (optional)
- Word Chains (required if using N-gram)
- Part of Speech (optional)
- Word Sentiment (optional for Sentiment Analysis)

Step 2: Word segmentation (2)

Chinese Word Segmentation

- FMM
- BMM
- N-gram

```
def word_seg_fmm(content): #正向匹配
    MaxLen=10 #最大词长
    Len=MaxLen #动态切割词长
    Seg_Content="" #返回的切割结果

    while len(content)>0:
        if content[0:Len] in WordMap: #词典中有匹配
            Seg_Content=Seg_Content+content[0:Len]+" | "
            content=content[Len:]
            Len=MaxLen
            #print("Seg_Content1:"+Seg_Content)
            continue
        else: #词典中无匹配
            Len=Len-1
            if Len==1:#仅剩一个词还没匹配到
                Seg_Content = Seg_Content + content[0:Len] + " | "
                content = content[Len:]
                Len = MaxLen
                #print("Seg_Content2:" + Seg_Content)
    return Seg_Content[:-1]
```

```
def word_seg_bmm(content): #逆向匹配
    MaxLen=10 #最大词长
    Len=MaxLen #动态切割词长
    Seg_Content="" #返回的切割结果

    while len(content)>0:
        if content[-Len:] in WordMap: #词典中有匹配
            Seg_Content=content[-Len:]+" | "+Seg_Content
            content=content[:-Len]
            Len=MaxLen
            #print("Seg_Content1:"+Seg_Content)
            continue
        else: #词典中无匹配
            Len=Len-1
            if Len==1:#仅剩一个词还没匹配到
                Seg_Content = content[-Len:] + " | " + Seg_Content
                content = content[:-Len]
                Len = MaxLen
                #print("Seg_Content2:" + Seg_Content)
    return Seg_Content[:-1]
```

Step 2: Word segmentation (3)

- Tips for Chinese Word Segmentation
  - Initialization is very important
  - Segment in the memory (not hard disk or data bases) to accelerate the segmentation speed
  - Using “set” to store the dictionary, and “list” for segmented words in Python
  - For Tag Analysis, a precise word segmentation is unnecessary

Step 3:Word Tag Extraction and Statistical Computing

- str.split() for all tags
- Discarding One-Char tags
- Discarding Stop-Word tags
- Select tags whose term frequencies are larger than a threshold (for example >2)
- Other statistical computing

Source code of Tagging

上海外国语大学
SHANGHAI INTERNATIONAL STUDIES UNIVERSITY

```
def Tagging(tagging_text):
    tagging = {}
    #stop_word = set()
    word_tags_list = str(tagging_text.strip()).split('|')
    for j in range(len(word_tags_list)):#去掉单个字
        if len(word_tags_list[j]) <= 1:
            word_tags_list[j]=''  '#最终由停用词过滤
    for i in word_tags_list:#统计每个词多少次
        if word_tags_list.count(i) > 1:
            tagging[i] = word_tags_list.count(i)
    # 连接到MySQL数据库
    # 1.Connection Open
    conn = pymysql.connect(user='root', password='123456', database='nlp', charset="utf8")
    # 2.Cursor Creating:
    cursor = conn.cursor()
    # 3.SQL Execution
    # 执行SQL语句，循环插入记录:
    sqlstr="SELECT DISTINCT WORD_NAME FROM NLP_STOP_WORD"
    # 4.Cursor Moving
    # 执行, 游标移至当前位置
    cursor.execute(sqlstr)
    # 使用fetchall函数，将结果集（多维元组）存入rows里面
    rows = cursor.fetchall()
    # 依次遍历结果集，发现每个元素，就是表中的一条记录，用一个元组来显示
    for row in rows:
        if str(row[0]) in tagging:
            tagging.pop(str(row[0]))  # 停用词过滤
    # 提交事务:
    conn.commit()
    # 5.Connection Close
    # 关闭Cursor:
    cursor.close()
    # 关闭Connection:
    conn.close()
    return tagging
```

Text Mining

Step 4: Data Visualization and News Summarization

```
def word_cloud_generate(word_tagging):
    # 获取当前文件路径, __file__ 为当前文件, 在ide中运行此行会报错,可改为
    # d = path.dirname('.')
    d = path.dirname(__file__)
    tlist=[]
    alice_coloring = imread(path.join(d, "China.png")) # 设置背景图片
    wc = WordCloud(background_color="white", # 背景颜色max_words=2000,# 词云显示的最大词数
                   mask=alice_coloring, # 设置背景图片
                   # stopwords=STOPWORDS.add("said"),
                   max_font_size=80, # 字体最大值
                   font_path="C:\\Windows\\Fonts\\simhei.ttf",
                   random_state=42)
    # 生成词云, 可以用generate输入全部文本(中文不好分词),也可以我们计算好词频后使用generate_from_frequencies函数
    # wc.generate(text)
    word_tagging_keys = word_tagging.keys()
    for word_tagging_key in word_tagging_keys:
        t = (word_tagging_key,word_tagging[word_tagging_key])
        tlist.append(t)
    wc.generate_from_frequencies(tlist)   # txt_freq例子为[('词a', 100),('词b', 90),('词c', 80)]
    image_colors = ImageColorGenerator(alice_coloring) # 从背景图片生成颜色值
    plt.imshow(wc) # 以下代码显示图片
    plt.axis("off")
    plt.figure() # 绘制词云
    plt.imshow(wc.recolor(color_func=image_colors)) # recolor wordcloud and show, we could also give color_func=image_colors directly in the constructor
    plt.axis("off")
    # 绘制背景图片为颜色的图片
    plt.figure()
    plt.imshow(alice_coloring, cmap=plt.cm.gray)
    plt.axis("off")
    plt.show()
    wc.to_file(path.join(d, "Tag_test.png")) # 保存图片
```

上海外国语大学
SHANGHAI INTERNATIONAL STUDIES UNIVERSITY

data visualization using python

Data Visualization

Data Visualization using Python

- Necessity:
  - NumPy (Computing Package)
  - Scipy (Scientific Computing Package)
  - Pillow(Image)
  - Matplotlib (Diagram Package)
  - wordcloud (Word Cloud Package)

Installation Sequence

- Some packages also need some other required packages

Recommendation Installation Method

- Download your corresponding package from:
  http://www.lfd.uci.edu/~gohlke/pythonlibs/

- Install them using “pip install” according to the installation sequence

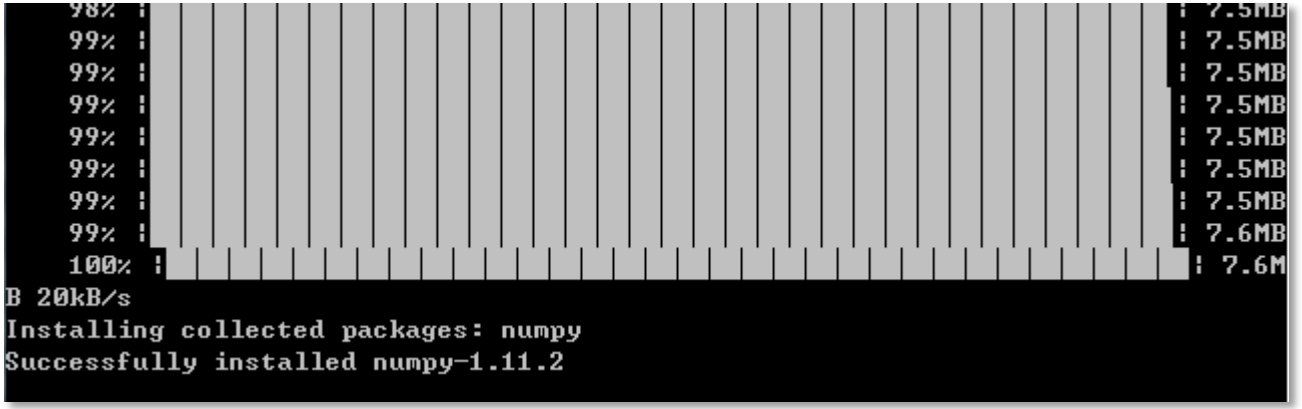
```
C:\Users\Ting>pip install D:\software\numpy-1.11.2+mkl-cp35-cp35m-win_amd64.whl
Requirement already satisfied (use --upgrade to upgrade): numpy==1.11.2+mkl from
 file:///D:/software/numpy-1.11.2%2Bmkl-cp35-cp35m-win_amd64.whl in c:\program f
iles\python35\lib\site-packages

C:\Users\Ting>pip install D:\software\scipy-0.18.1-cp35-cp35m-win_amd64.whl
Processing d:\software\scipy-0.18.1-cp35-cp35m-win_amd64.whl
Installing collected packages: scipy
Successfully installed scipy-0.18.1

C:\Users\Ting>pip install D:\software\matplotlib-2.0.0rc1-cp35-cp35m-win_amd64.w
hl
Processing d:\software\matplotlib-2.0.0rc1-cp35-cp35m-win_amd64.whl
Collecting pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=1.5.6 (from matplotlib==2.0.0rc1)
```

Data Visualization

- Ref: NumPy
- Official web site: http://www.numpy.org/
- pip install numpy (may be fail in Windows)



Data Visualization

- Ref: Scipy
- Official web site: http://www.scipy.org/
- pip install scipy (may be fail in Windows)

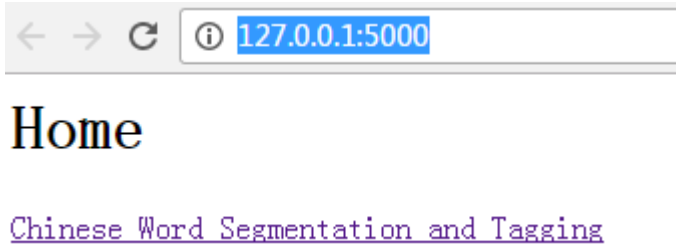
Data Visualization

wordcloud

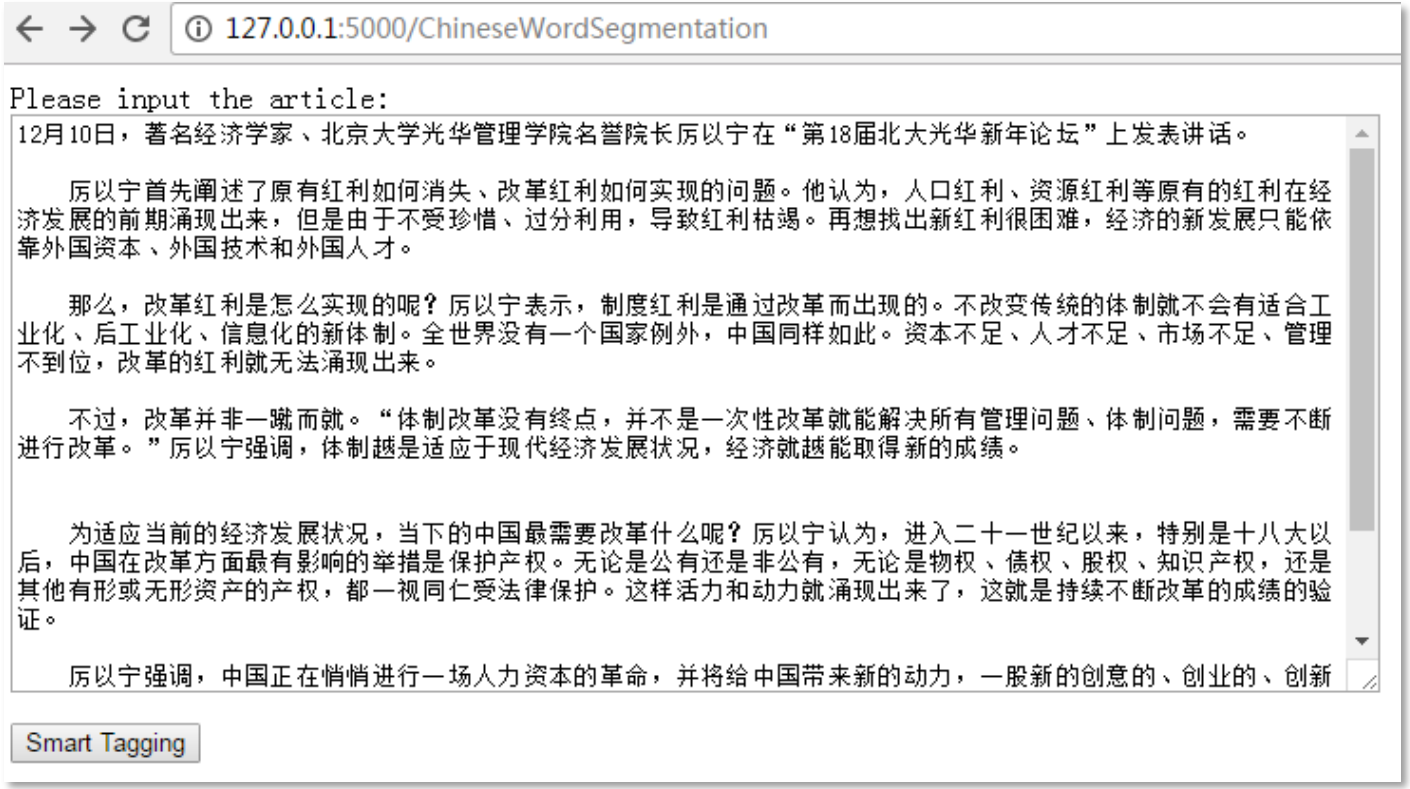
- Official web site: https://github.com/amueller/word_cloud
- Install wordcloud: pip install wordcloud

Data Visualization

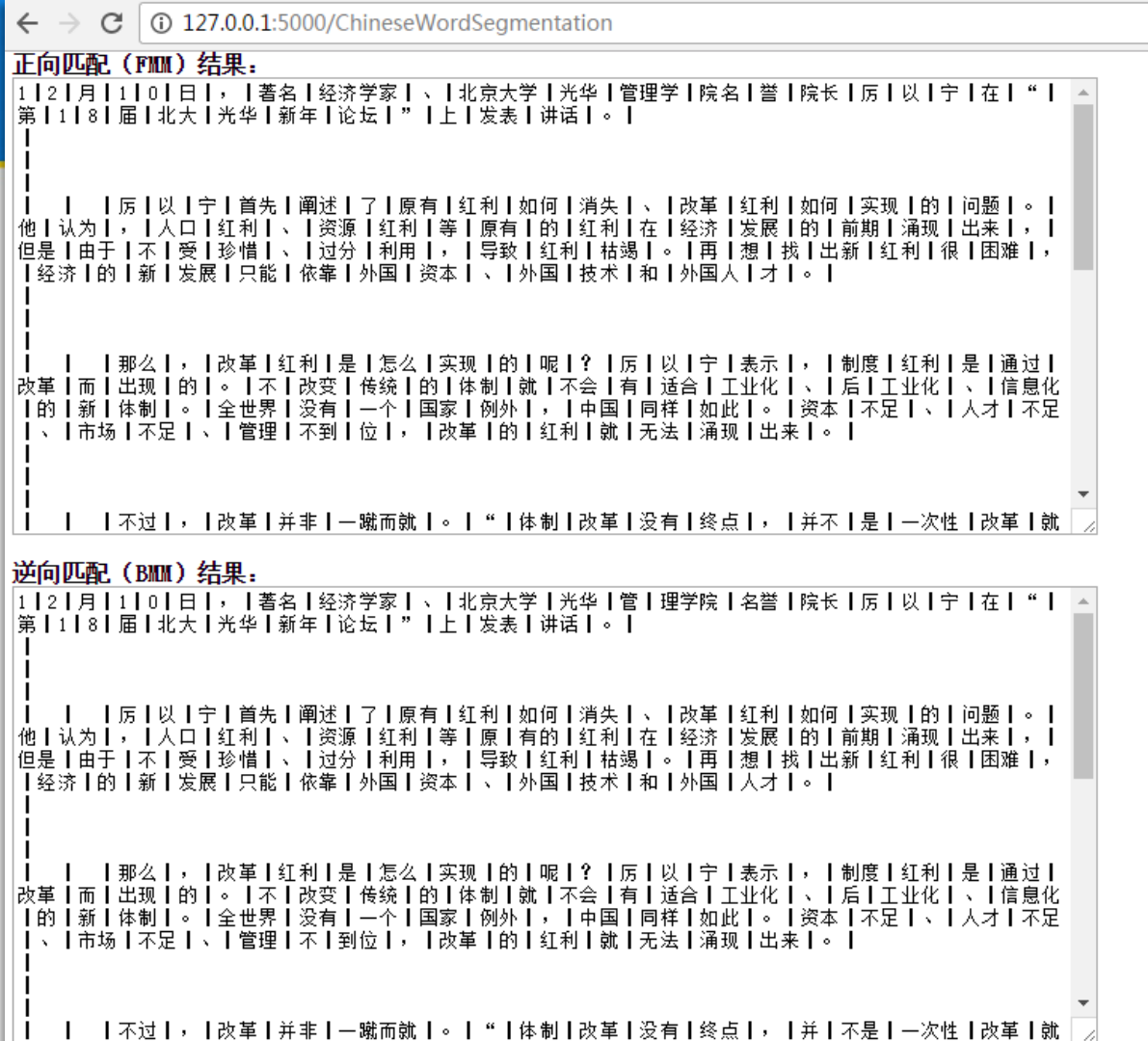
- Step1: Program Running



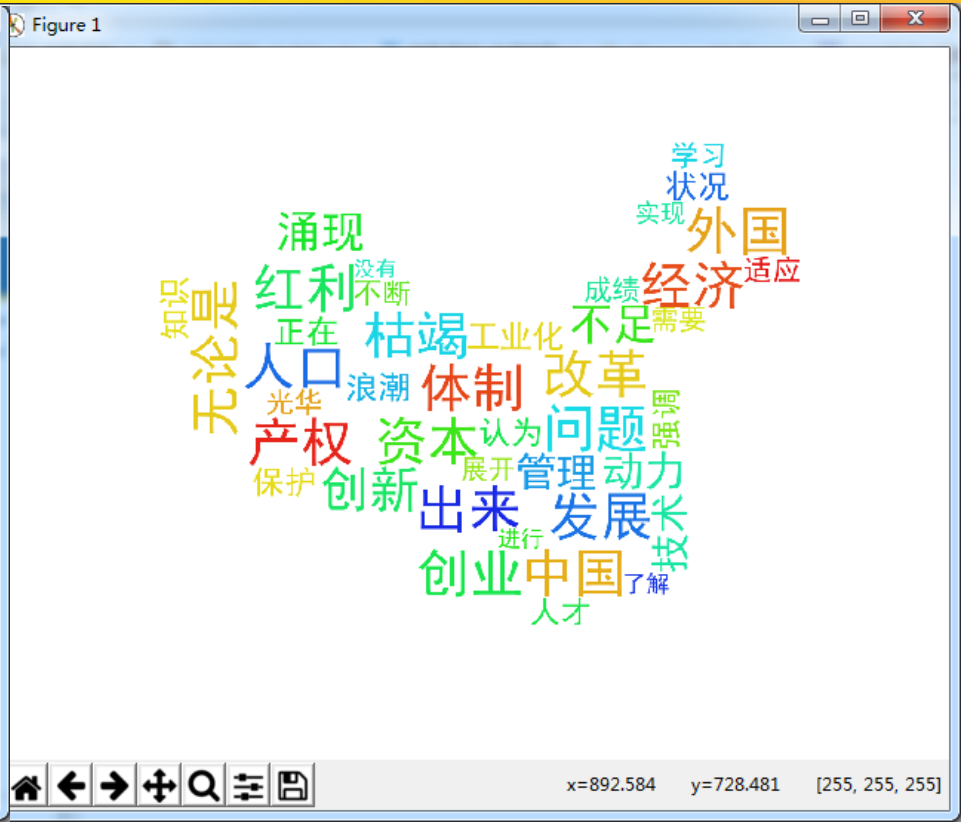
Step2: Input Text



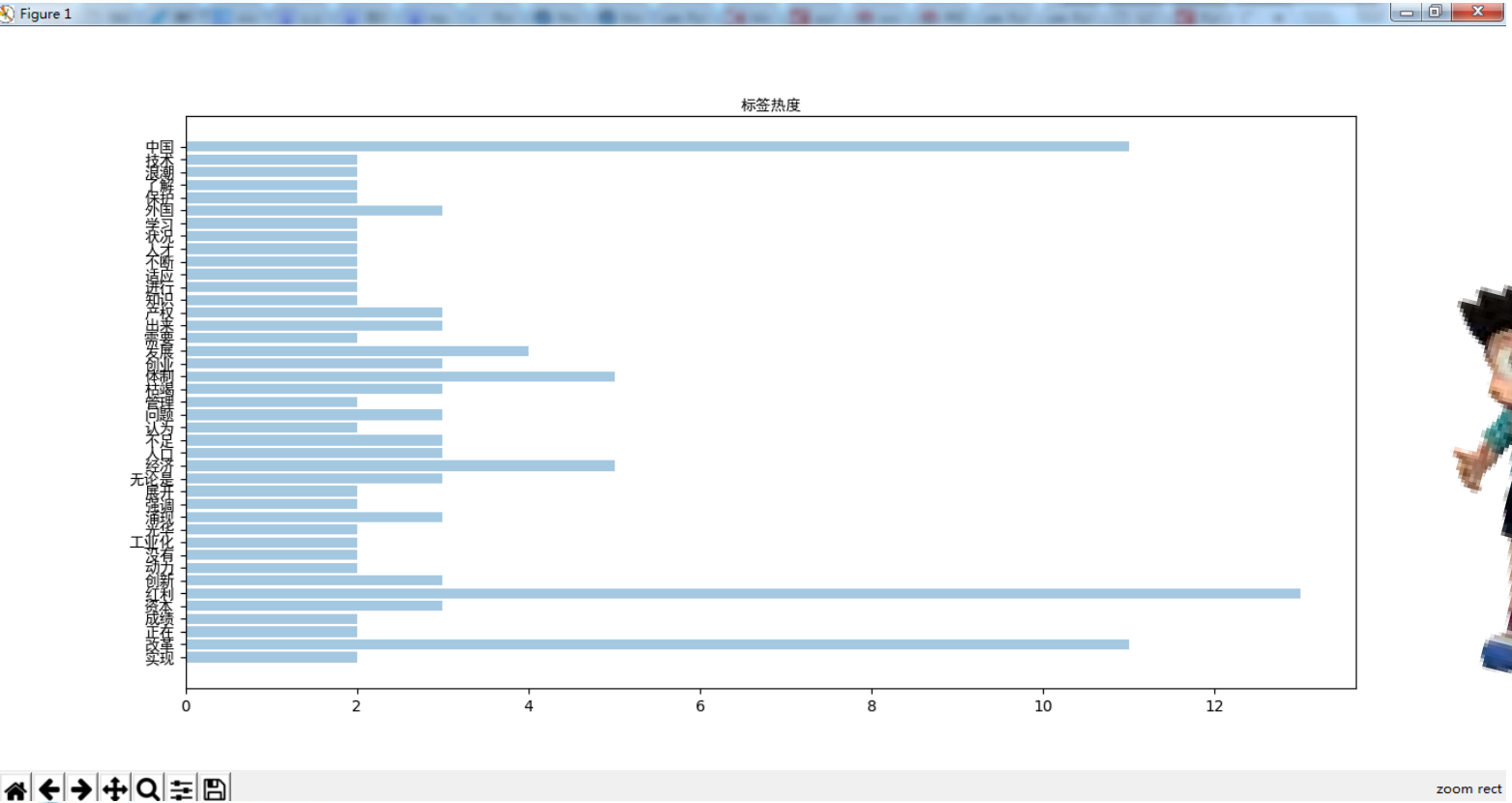
Step 3: Word Segmentation Result



Data Visualization



Data Visualization



标签分析：

了解
保护
工业化
创新
强调
动力
没有
出来
不断
状况
适应
资本
成绩
进行
展开
学习
技术
创业
枯竭
认为
无论
是
知识
体制
浪潮
发展
中国
涌现
管理
改革
人才
不足
实现
人口
产权
外国
问题
需要
红利

Conclusions

本文与中国改革红利有关，从经济、体制、发展等方面进行了论述，中国未来的发展与外国、创业、人口、知识产权等有关，但也存在不足和问题。

Data Visualization

How can we compare multiple different news reports?

上海外国语大学
SHANGHAI INTERNATIONAL STUDIES UNIVERSITY

machine learning approaches for data mining

Data Mining Essentials

Data Mining 数据挖掘

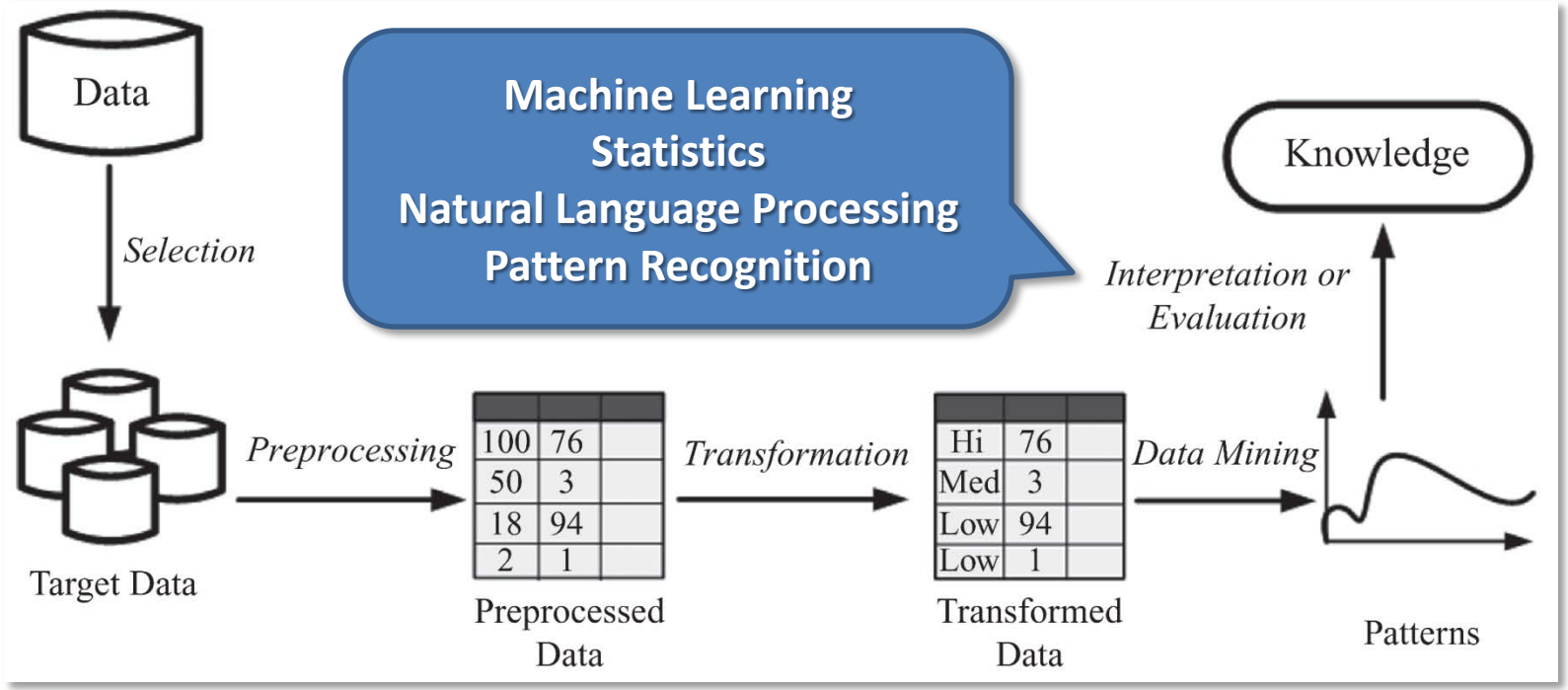
- Data Mining is the power for producing high-quality journalism.
- Data Mining is an interdisciplinary subfield of computer science, and statistics.

Social Demands

- Data production rate has increased dramatically (**Big Data**) and we are able store much more data
  - E.g., purchase data, social media data, cell phone data
- Businesses and customers need <u>useful</u> or <u>actionable</u> knowledge to gain insight from raw data for various purposes
  - It's not just searching data or databases

The process of extracting useful patterns from raw data is known as Knowledge Discovery in Databases (KDD)

KDD from Data Bases



Data Mining Essentials

Data 数据

- Continuous Data 连续型数据
  - Regression
- Discrete Data 离散型数据
  - Classification

Data Feature (1) 数字特征

Feature also called as Measurement, Attribute

- **Nominal 名词性**
  - **Operations**:
    - Mode (most common feature value), Equality Comparison
  - E.g., {male, female}
- **Ordinal 序数性**
  - Feature values have an intrinsic order to them, but the difference is not defined
  - **Operations**:
    - same as nominal, feature value rank
  - E.g., {Low, medium, high}

Data Feature (2) 数字特征

- **Interval 间隔性**
  - **Operations:**
    - Addition and subtractions are allowed whereas divisions and multiplications are not
  - E.g., 3:08 PM, calendar dates
- **Ratio 比例性**
  - **Operations:**
    - divisions and multiplications are allowed
  - E.g., Height, weight, money quantities

Data Quality 数据质量

- **Noise 噪声数据**
  - Noise is the distortion of the data
- **Outliers 异常值**
  - Outliers are data points that are considerably different from other data points in the dataset
- **Missing Values 缺失值**
  - Missing feature values in data instances
  - **Solution:**
    - Remove instances that have missing values
    - Estimate missing values, and
    - Ignore missing values when running data mining algorithm
- **Duplicate data 重复数据**

- ***Data Preprocessing (1)***
  数据预处理
- **Aggregation 聚合**
  - It is performed when multiple features need to be combined into a single one or when the scale of the features change
  - Example: image width , image height -> image area (width x height)
- **Discretization 离散化**
  - From continues values to discrete values
  - Example: money spent -> {low, normal, high}

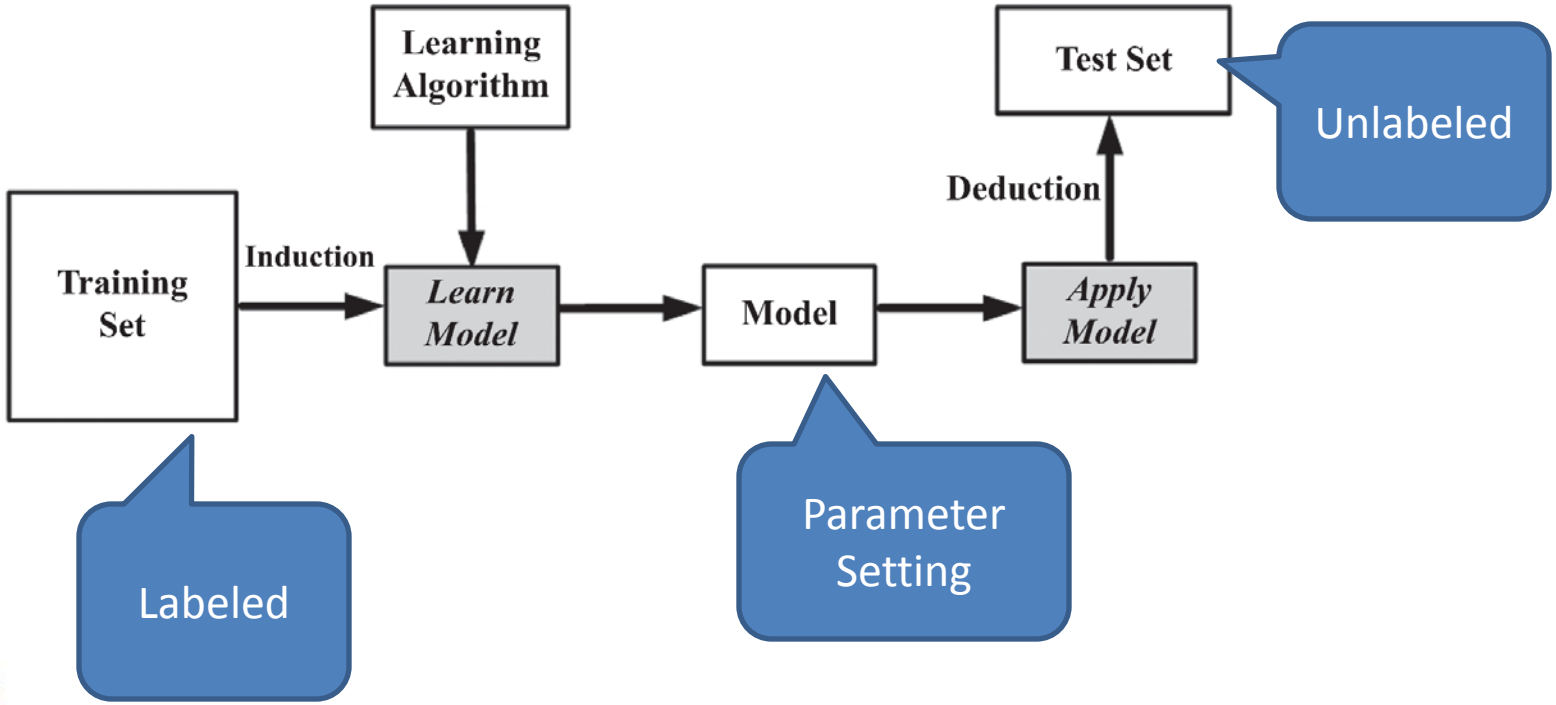
- ***Data Preprocessing (2)数据预处理***
- **Feature Selection 特征选择**
  - Choose relevant features
- **Feature Extraction 特征提取**
  - Creating new features from original features
  - Often, more complicated than aggregation
- **Sampling 取样**
  - Random Sampling
  - Sampling with or without replacement
  - Stratified Sampling: useful when having class imbalance
  - Social Network Sampling

Data Mining Essentials

Machine Learning 机器学习

- Supervised Learning
  - Classification
  - Regression
- Unsupervised Learning
  - Clustering
  - Dimensional Reduction

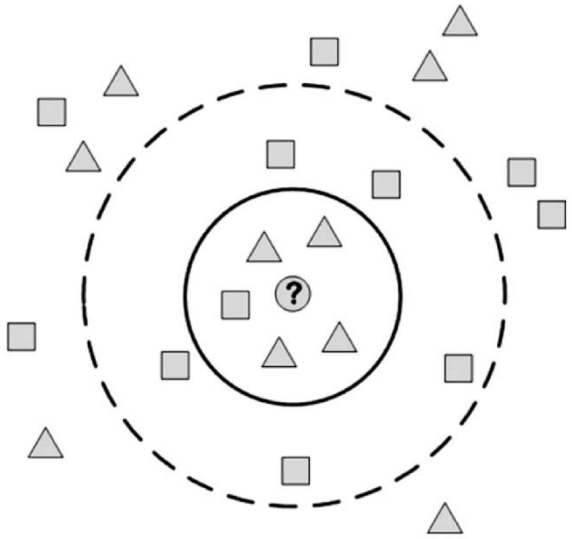
Supervised Machine Learning 有监督学习



Classification 分类

Prediction Result with Labeled Discrete Value

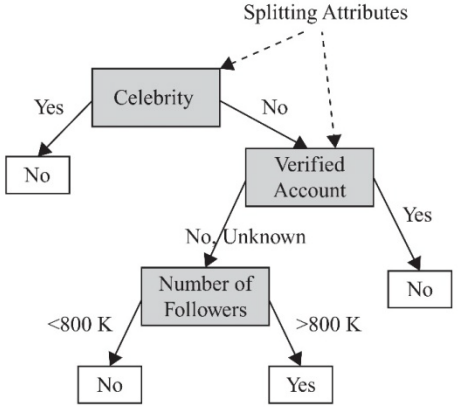
- KNN(K-Nearest Neighbors) K临近原则
- Linear Classifier 线性分类器
- Neural Networks 神经网络
- Support Vector Machine 支撑向量机
- Decision Tree 决策树



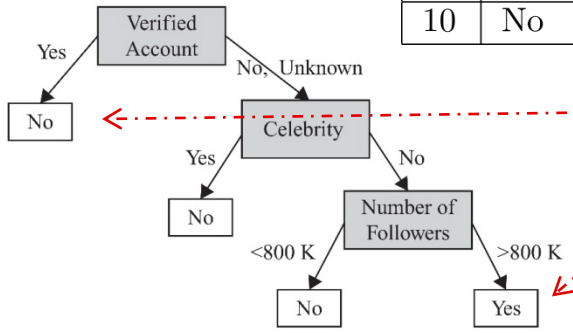
Data Mining Essentials

Multiple **decision trees** can be learned from the same dataset

ID	Celebrity	Verified Account	# Followers	Influential?
1	Yes	No	1.25M	No
2	No	Yes	1M	No
3	No	Yes	600K	No
4	Yes	Unknown	2.2M	No
5	No	No	850K	Yes
6	No	Yes	750K	No
7	No	No	900K	Yes
8	No	No	700K	No
9	Yes	Yes	1.2M	No
10	No	Unknown	950K	Yes



(a) Learned Decision Tree 1

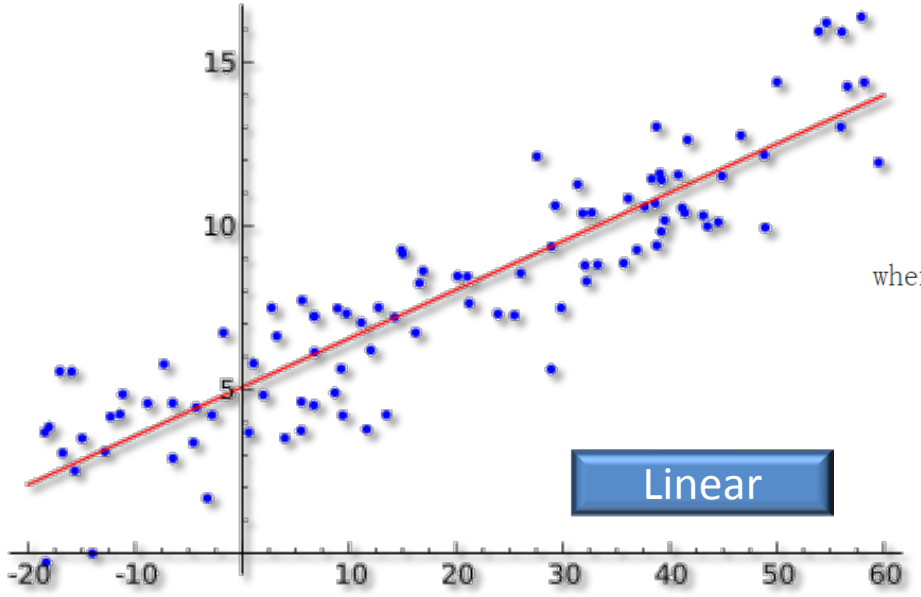


(b) Learned Decision Tree 2

Class Labels

Regression (1) *回归*

Prediction Result with Unlabeled Continuous Value



Eg. Linear least squares 线性最小二乘法

$$\mathbf{X}\boldsymbol{\beta} = \mathbf{y},$$

where

$$\mathbf{X} = \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1n} \\ X_{21} & X_{22} & \cdots & X_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ X_{m1} & X_{m2} & \cdots & X_{mn} \end{bmatrix}, \quad \boldsymbol{\beta} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}.$$

Regression (2)回归

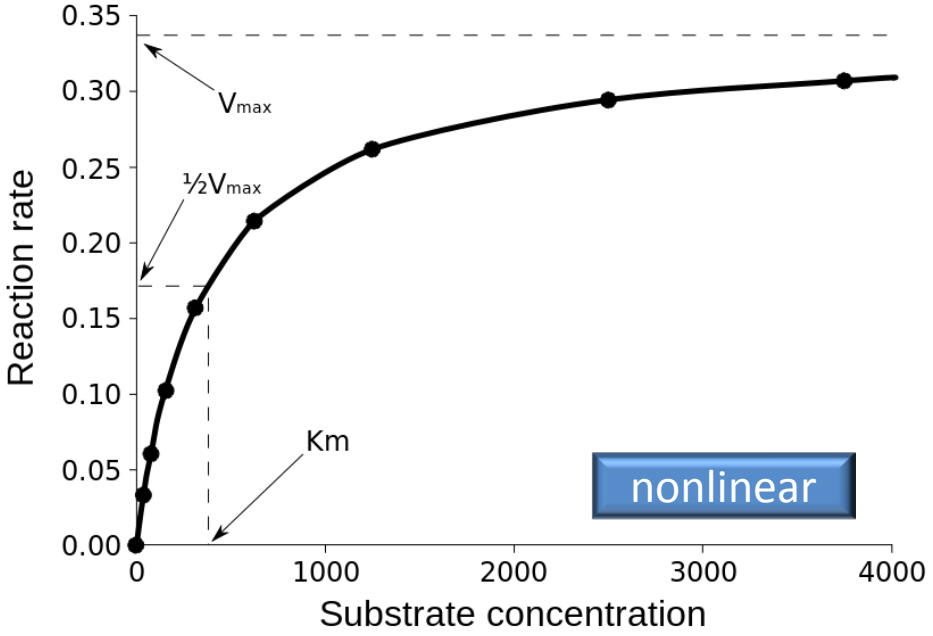
Nonlinear Regression 非线性回归计算

- Linearization 线性化方法
  1. Transformation 变形法

$$y = ae^{bx}U \quad \Rightarrow \quad \ln(y) = \ln(a) + bx + u$$

  2. Segmentation 分割法

split up into classes or segments and *linear regression* can be performed per segment



Unsupervised Machine Learning

无监督学习

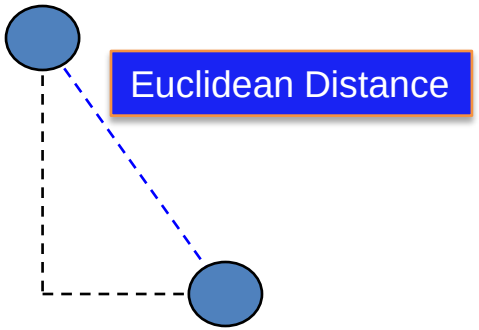
machine learning task of inferring a function to describe hidden structure from unlabeled data

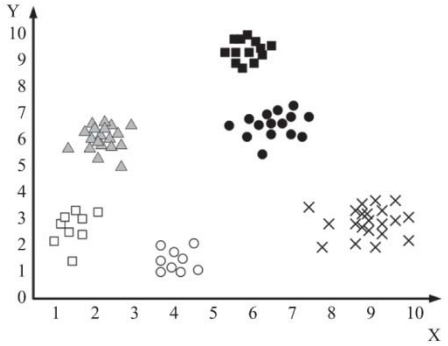
Clustering 聚类

- **Clustering Goal:** Group together similar items
- Clustering algorithms group together **similar items**
  - –The algorithm does not have examples showing how the samples should be grouped together (unlabeled data)

Similarity Computing (1) 相似度计算

- –The most popular (dis)similarity measure for continuous features are **Euclidean Distance** and **Pearson Linear Correlation**





$$d(X,Y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2} = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

Similarity Computing (2)相似度计算

X and Y are n Dimensional Vectors

$$X = (x_1, x_2, \ldots, x_n)$$

$$Y = (y_1, y_2, \ldots, y_n)$$

Measure Name	Formula	Description
Mahalanobis	$d(X,Y) = \sqrt{(X-Y)^T \Sigma^{-1} (X-Y)}$	X, Y are features vectors and Σ is the covariance matrix of the dataset
Manhattan (L_1 norm)	$d(X,Y) = \sum_i \lvert x_i - y_i \rvert$	X, Y are features vectors
L_p-norm	$d(X,Y) = (\sum_i \lvert x_i - y_i \rvert^n)^{\frac{1}{n}}$	X, Y are features vectors

Once a distance measure is selected, instances are grouped using it.

Pearson Linear Correlation 皮尔逊线性相关

Correlation Coefficient 相关系数

$$\rho_{X,Y} = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y}$$

Where, cov **is the covariance**

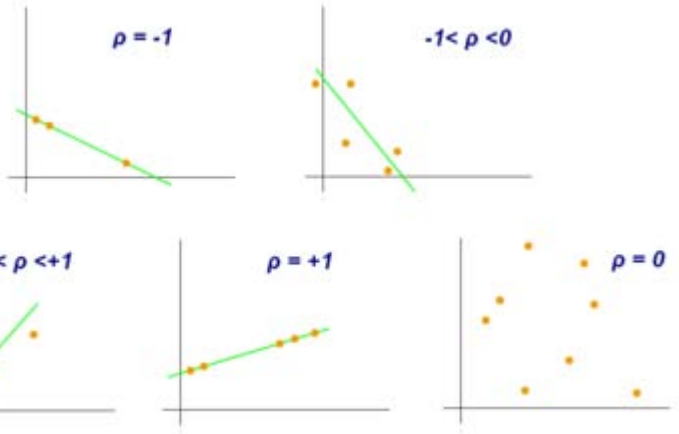
σ **is the standard deviation**

$$\operatorname{cov}(X,Y) = \mathrm{E}[(X-\mu_X)(Y-\mu_Y)]$$

$$\rho_{X,Y} = \frac{\mathrm{E}[XY] - \mathrm{E}[X]\,\mathrm{E}[Y]}{\sqrt{\mathrm{E}[X^2] - [E[X]]^2}\,\sqrt{\mathrm{E}[Y^2] - [E[Y]]^2}}.$$

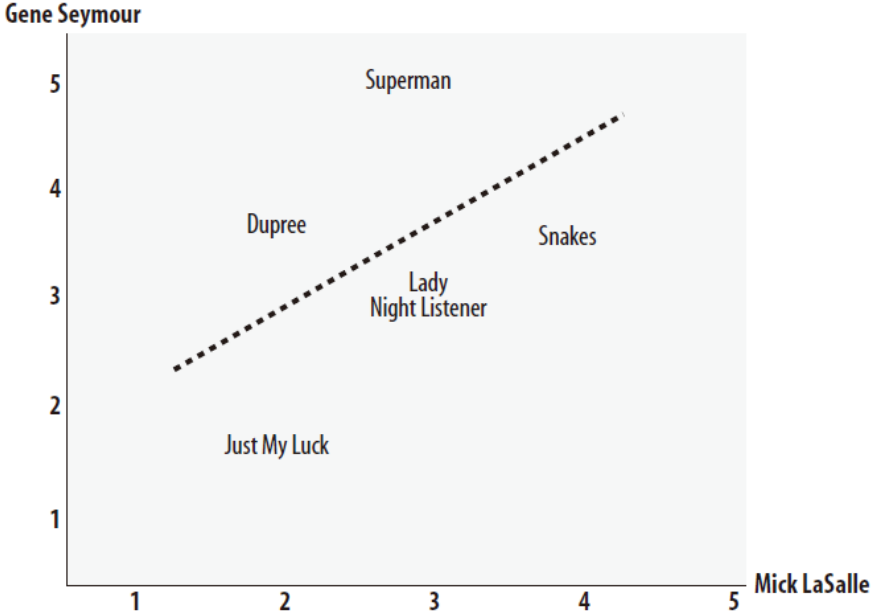
Relations between Variance and Covariance

$$\mu_X = \mathrm{E}[X]$$
$$\mu_Y = \mathrm{E}[Y]$$
$$\sigma_X^2 = \mathrm{E}[(X - \mathrm{E}[X])^2] = \mathrm{E}[X^2] - [E[X]]^2$$
$$\sigma_Y^2 = \mathrm{E}[(Y - \mathrm{E}[Y])^2] = \mathrm{E}[Y^2] - [E[Y]]^2$$
$$\mathrm{E}[(X-\mu_X)(Y-\mu_Y)] = \mathrm{E}[(X - \mathrm{E}[X])(Y - \mathrm{E}[Y])] = \mathrm{E}[XY] - \mathrm{E}[X]\,\mathrm{E}[Y]$$

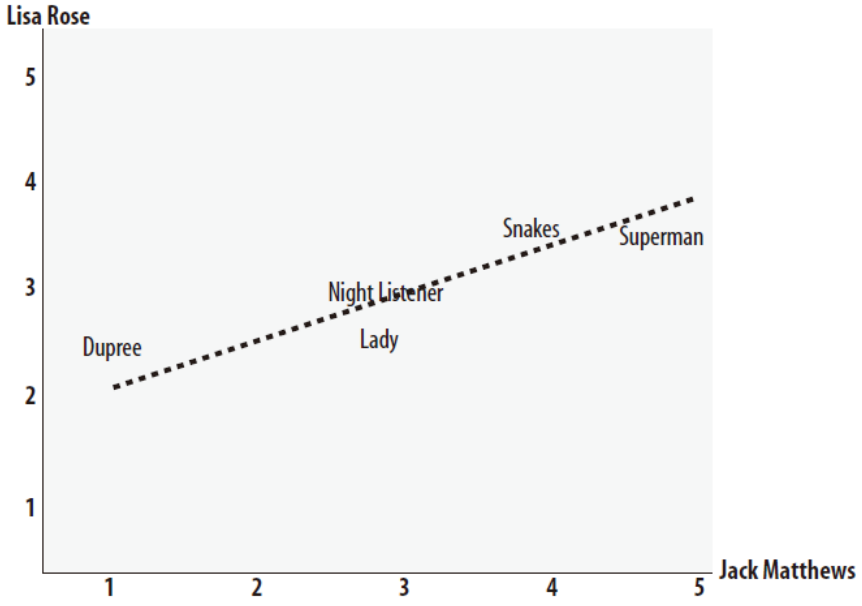


Film Ranking Correlation

***Superman** was rated 3 by Mick LaSalle and 5 by Gene Seymour,* so it is placed at (3,5) on the chart.



$\rho_{X,Y} = 0.4$



$\rho_{X,Y} = 0.75$

Conclusion: Films recommended to Lisa, also can be recommended to Jack.

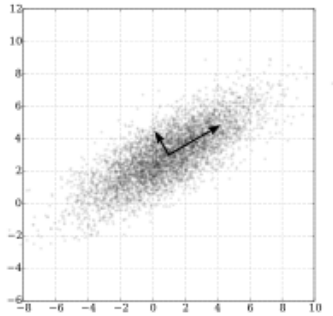
Dimensional Reduction 降维

Principal Component Analysis (PCA) 主成份分析

1. PCA is a statistical procedure ***converts*** a set of observations of possibly **correlated variables *into*** a set of values of linearly **uncorrelated variables** called principal components.

2. The number of principal components is **less than or equal to** the number of original variables.

3. This transformation is defined in such a way that **the first principal component** has ***the largest possible variance***, and each **succeeding component** in turn has **the highest variance possible under the constraint** that it is orthogonal to the preceding components.



Ref. http://www.cnblogs.com/SCUJIN/p/5965946.html

Reference

Reference

Books and Chapters (1)

https://item.jd.com/11983227.html

Chapter 1-2

Machine Learning Package Installation

Machine Learning Theory Foundations

Reference

Books and Chapters (2)

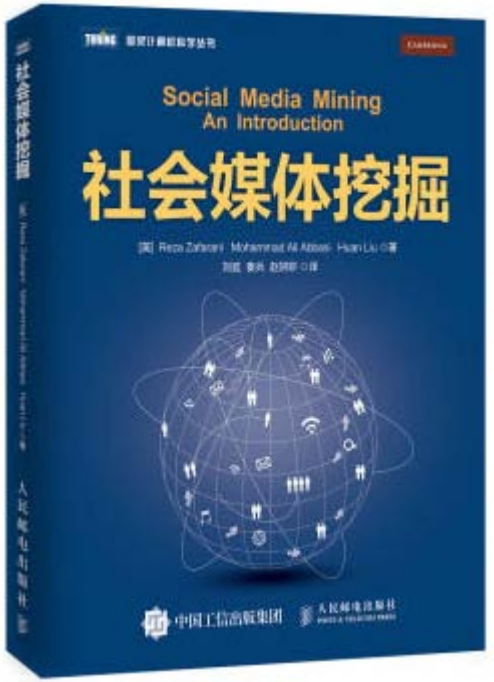
https://item.jd.com/11803260.html

Chapter 5

Data Mining Essentials

Online Reference:

http://www.public.asu.edu/~huanliu/

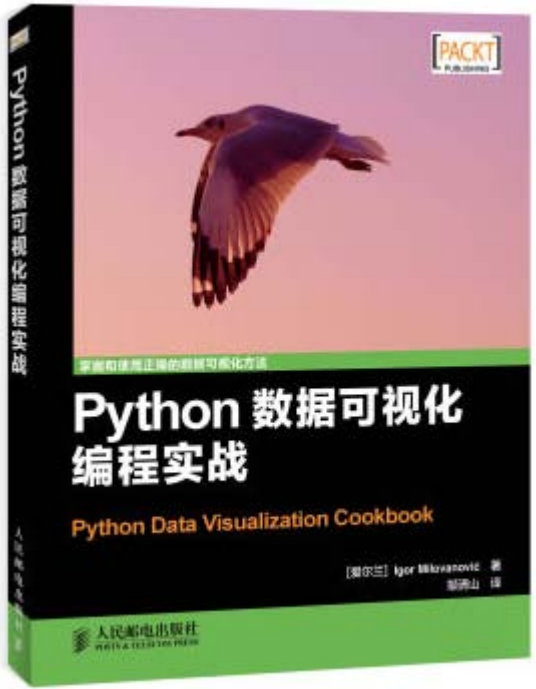


Reference

Books and Chapters (3)

https://item.jd.com/11676691.html

Python Data Visualization

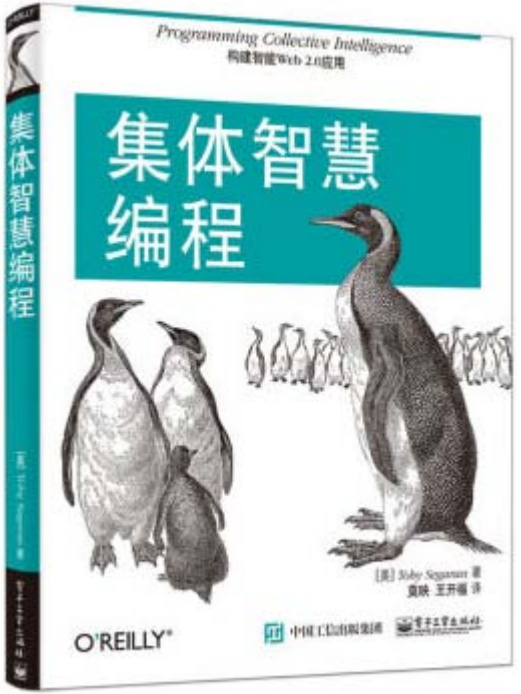


Reference

Books and Chapters (4)

https://item.jd.com/11667512.html

Programming Collective Intelligence



Python Extension Packages

http://www.lfd.uci.edu/~gohlke/pythonlibs/

Reference

Data Visualization in Python

- http://it.sohu.com/20151119/n427117609.shtml
- http://www.oschina.net/translate/python-data-visualization-libraries

Reference

Using WordCloud

- http://blog.csdn.net/tanzuozhev/article/details/50789226
- https://www.oschina.net/code/snippet_2294527_56155

Chinese Display

- http://blog.csdn.net/u012705410/article/details/47379957

Reference

Provided Repositories for Social Mining

- http://socialcomputing.asu.edu
- http://snap.Stanford.edu
- https://github.com/caesar0301/awesome-public-datasets

Homework

Homework

1. Finish Data Collection of your group
2. Try to use FMM/BMM/N-gram to segment words for a news report, and get tags from the segmentation result
3. Try to draw some conclusions based on the tags extracted from the segmentation results and some statistical computing
4. Write the final report of your group and hand it before Jan 6.

The End of Lecture 10

Thank You

http://www.wangting.ac.cn