本课件仅用于教学使用。未经许可，任何单位、组织和个人不得将课件用于该课程教学之外的用途(包括但不限于盈利等)，也不得上传至可公开访问的网络环境

数据科学导论

Introduction to Data Science

数据入门：特征工程

陈恩红，黄振亚
Email: cheneh@ustc.edu.cn, huangzhy@ustc.edu.cn

课程主页：
http://staff.ustc.edu.cn/~huangzhy/Course/DS2026.html

回顾：数据分析基础

- 数据采集　　Data Collection
- 数据预处理　　Data Preprocessing
- 特征工程　　Feature Engineering

特征工程

- 什么是数据的特征？
  - 用于表示数据
- 例：电子商务中的商品

CVPR 2021 AliProducts Challenge: Large-scale Product Recognition

- 背景：电商企业面临的大规模、细粒度商品图像识别问题
- 数据量：**300万张图片**，涵盖了**5万个SKU级商品类别**



特征工程

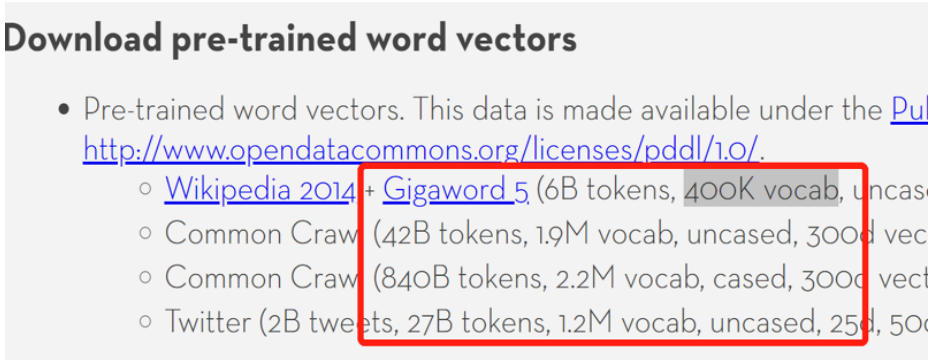
- **什么是数据的特征？**
  - 用于表示数据
  - 大数据应用包含几万至几百万的属性，其中大部分属性与挖掘任务不相关，是冗余的

个人借贷数据

loan_id	119262	贷款记录唯一标识
user_id	0	用户唯一标识
total_loan	12000.0	贷款金额
year_of_loan	5	贷款期限（year）
interest	11.53	贷款利率
…	…	…

NLP中的Glove词表

https://nlp.stanford.edu/projects/glove/



特征工程

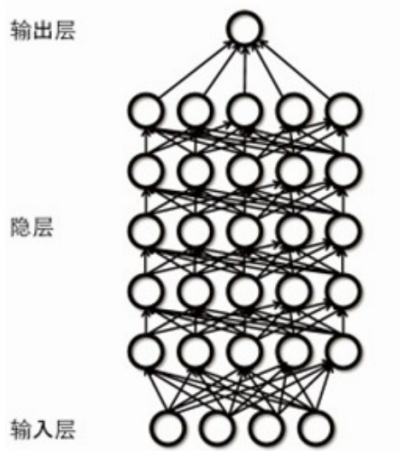
例子：主成分分析

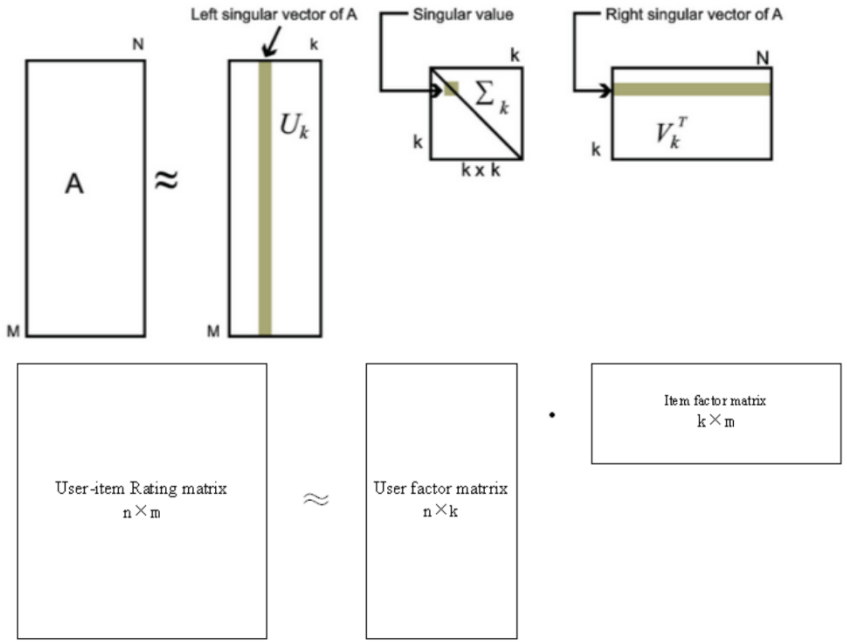
先把大图像分成16*16（256）的小图像块，把小图像块当成一个256维的向量，所有256维向量拼接成新的数据矩阵，对其进行归一化和PCA压缩（取前四个特征值，取前八个，前16个，一直到前256个特征值），压缩完以后需要重构图像就会得到以上的效果

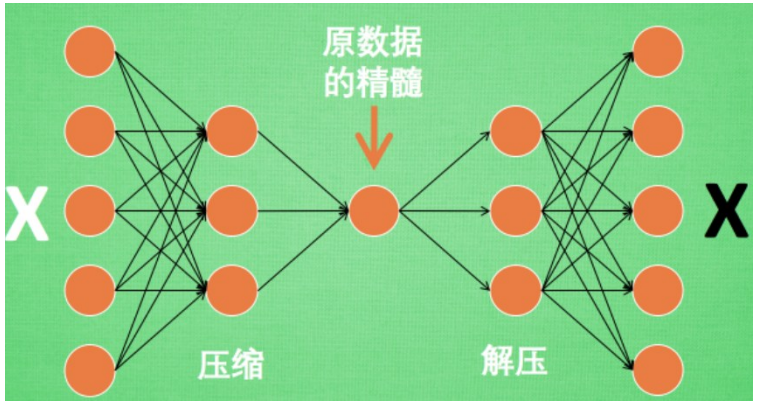


特征工程

- 特征学习方法
  - 奇异值分解(SVD)
  - 概率矩阵分解(PMF)
  - 深度学习(Deep Learning)
    - DNN，AutoEncoder, etc









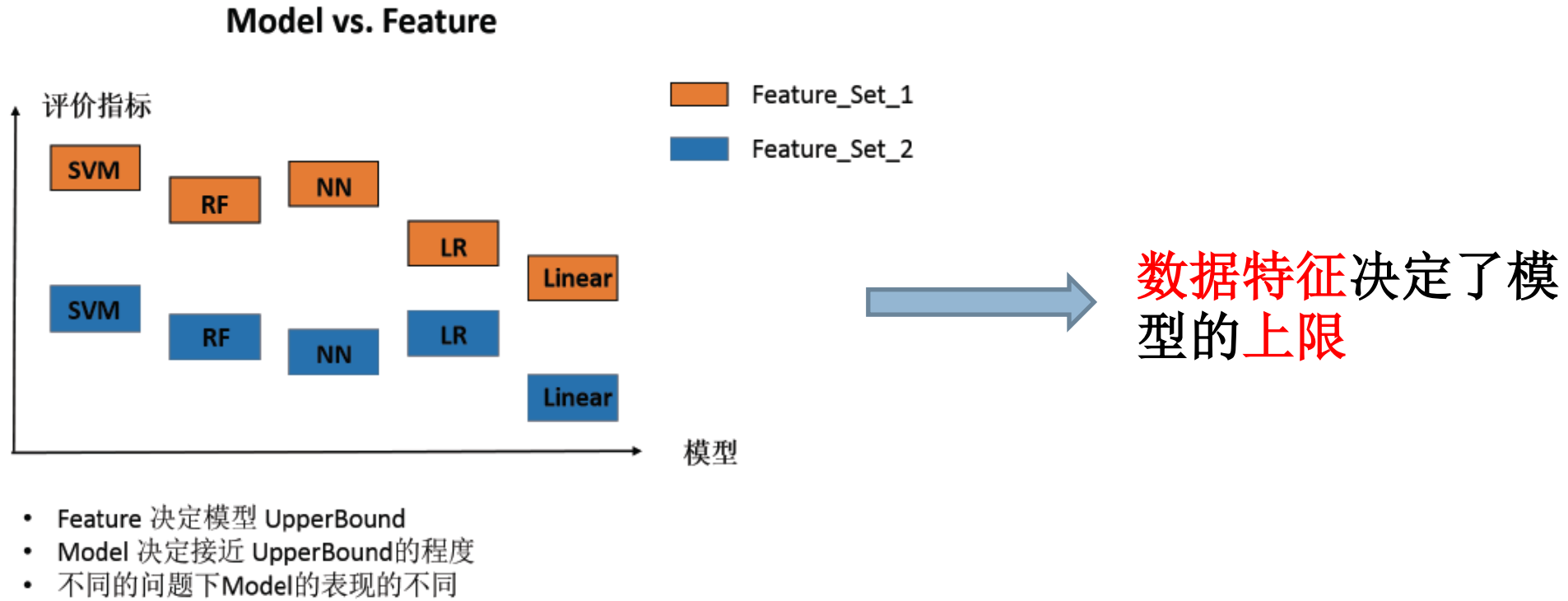
特征工程

- 特征工程的定义
- 特征工程的流程
- 特征学习
- 案例学习
- 参考文献

特征工程

- 特征工程是什么？ (Feature Engineering)

在数据预处理以后（或者数据预处理过程中），如何从数据中提取有效的特征，使这些特征能够尽可能的表达原始数据中的信息，使得后续建立的数据模型能达到更好的效果，就是特征工程所要做的工作。



特征工程

特征工程的意义

著名数据科学家Andrew Ng 对特征工程这样描述的：“虽然提取数据特征是非常困难、耗时并且需要相关领域的专家知识，但是机器学习应用的基础就是特征工程”

特征越好，灵活性越强

好的特征能使一般的模型也能获得很好的性能，在不复杂的模型上运行速度很快，并且容易理解和维护。

特征越好，构建的模型越简单

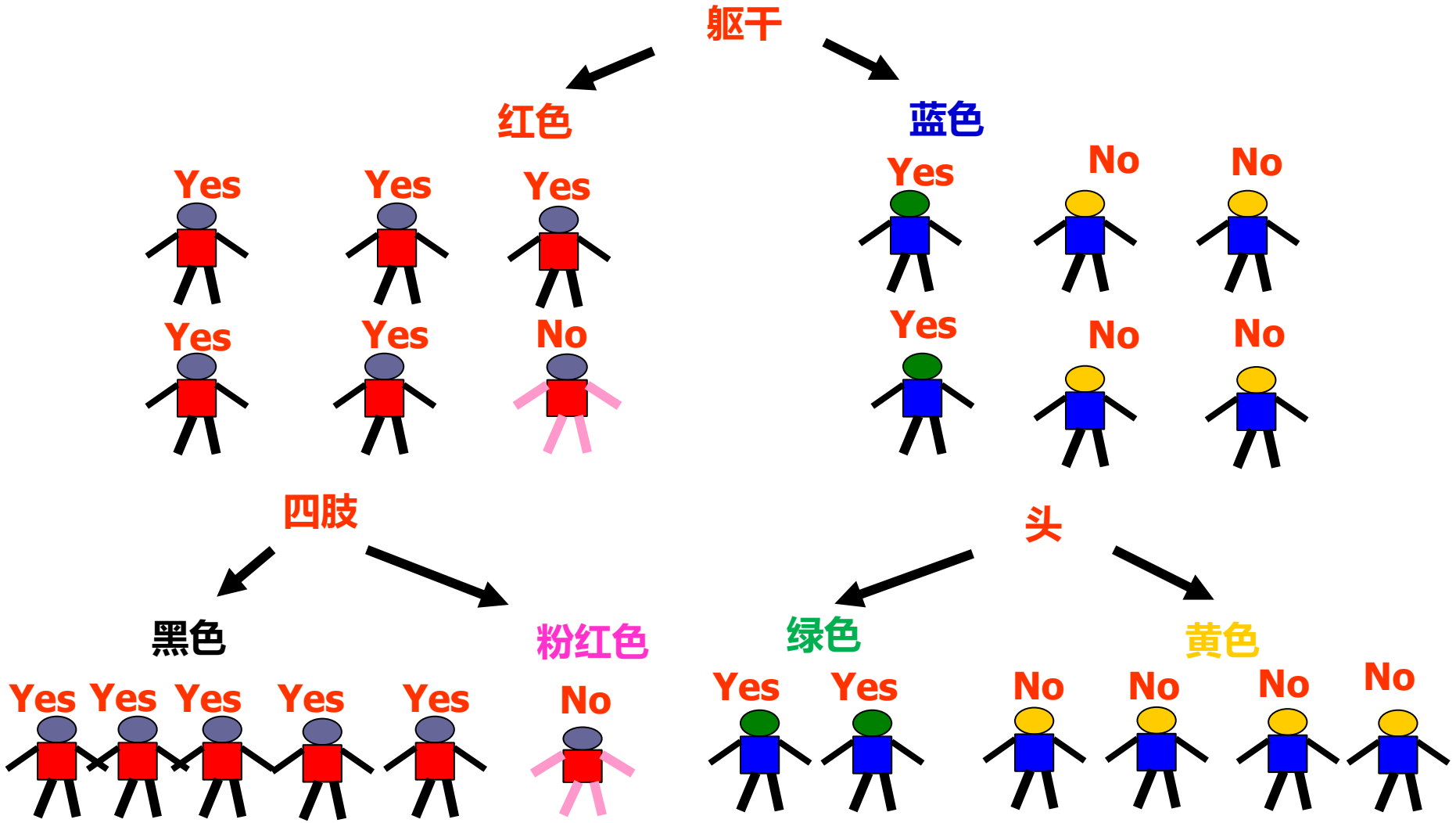
好的特征不需要花太多的时间去寻找最优参数，降低了模型的复杂度，使模型趋于简单。

特征越好，模型的性能越出色

好的特征能够使模型表现越出色是毫无疑问的，而特征工程的[illegible]性能



特征工程


躯干
红色
蓝色
Yes
Yes
Yes
Yes
Yes
No
Yes
No
No
Yes
No
No
四肢
头
黑色
粉红色
绿色
黄色
Yes
Yes
Yes
Yes
Yes
No
Yes
Yes
No
No
No
No

特征工程的流程

特征工程（重复迭代）的流程

1. 对特征进行头脑风暴

深入分析问题，观察数据的基本统计信息，结合问题的相关领域知识和参考其他问题的相关特征工程的方法并应用到自身的问题中来。

2. 特征的设计—基础且重要的步骤

人工设计特征、自动提取特征，或两者结合，得到模型使用的特征。

3. 特征的选择

使用不同的特征重要性评分方法或者特征选择方法，对特征的有效性进行分析，选出有效的特征。

4. 评估模型

利用所选择的特征对测试数据进行预测，评估模型的性能。

5. 上线测试

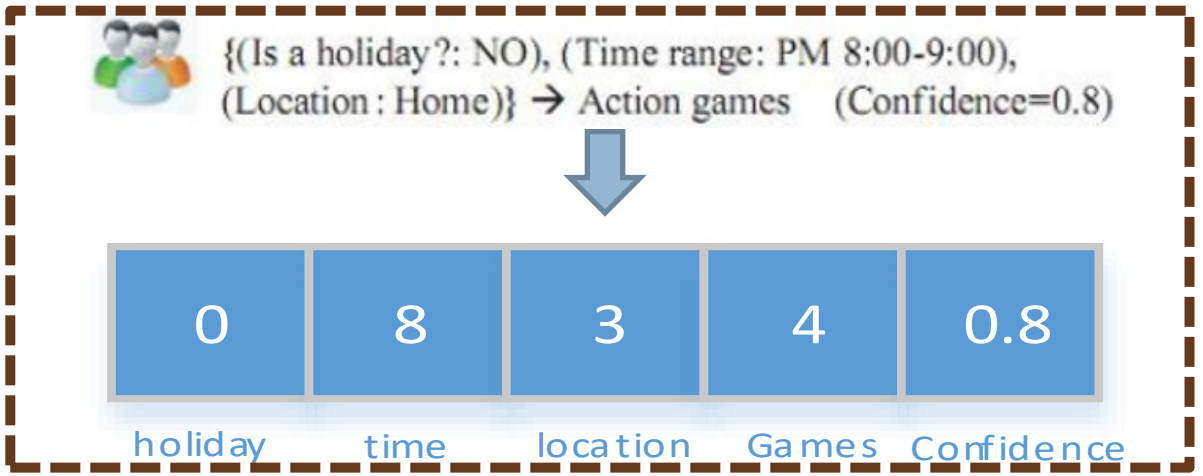
通过在线测试的效果判断特征是否有效，若不能达到要求，则重复2-5 步骤，直到模型的性能达到要求。

特征的设计

- 从原始数据中如何设计特征？
  - **基本特征的提取**

    基本特征的提取过程就是对原始数据进行预处理，将其转化成可以使用的数值特征。常见的方法有：数据的**归一化**、**离散化**、**缺失值补全**和**数据变换**等。
  - **创建新的特征**

    根据对应的领域知识，在基本特征的基础上进行特征之间的比值和交叉变化来构建新的特征。



特征的设计

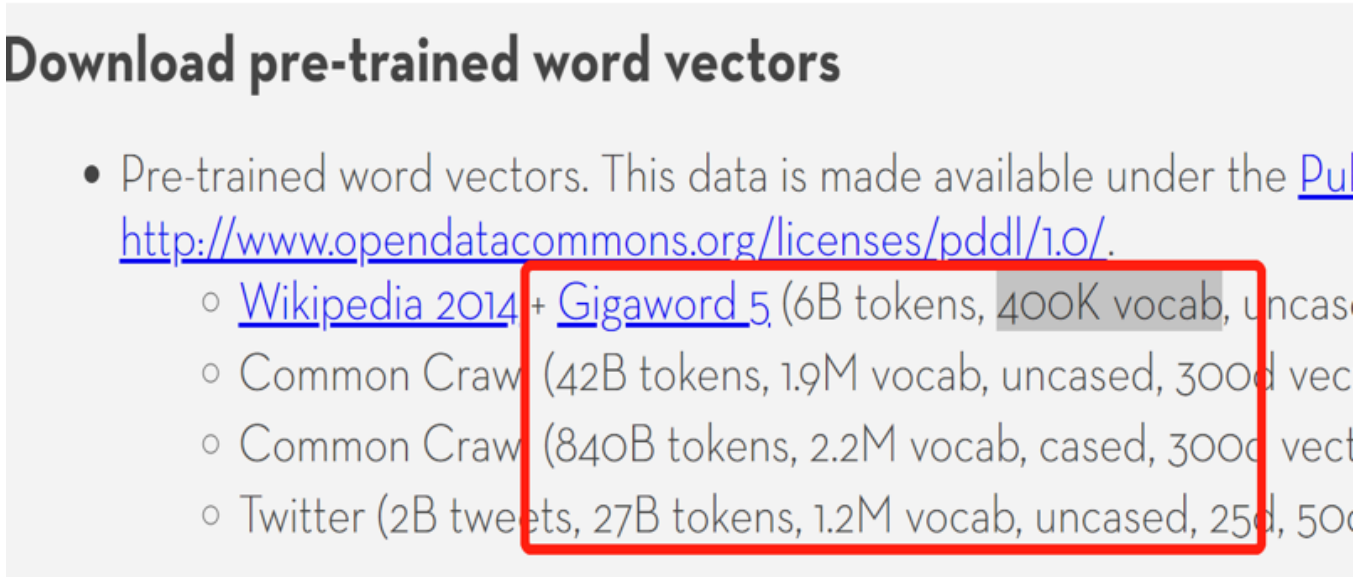
推荐系统中的“用户”和“商品”

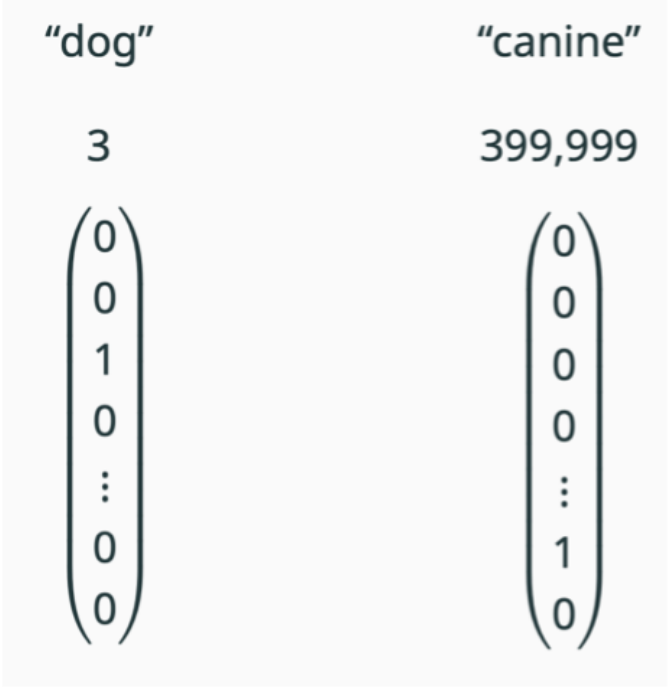


- 从原始数据中如何设计特征？
  - 独热特征表示 One-hot Representation
  - 将每个属性表示成一个很长的向量（每维代表一个属性值，如词语）
    - **函数：**[0, 0, 1, 0, 0, …, 0, 0, 0, 0]
    - **图像：**[0, 0, 0, 0, 0, …, 0, 0, 0, 1]
  - **优点：**直观，简洁
  - **缺陷**：
    - **“维度灾难”问题：**尤其是我们所构建的语料库包含的词语数据非常多的时候，独热表征在空间和时间上的开销都是十分巨大的
    - **“语义鸿沟”现象：**任意两个词之间都是完全孤立的，是无法刻画句子中词语的语序信息的（之前提到的词袋模型也是如此）。例如，我们是无法通过独热表征来判断“函数”与“偶函数”之间的联系的（但实际上这两个词语是非常相关的）。

特征的设计

- 从原始数据中如何设计特征？
  - 独热特征表示 One-hot Representation
    - **“维度灾难”问题**
    - **“语义鸿沟”现象**







特征的设计

- 从原始数据中如何设计特征？
  - **数据的统计特征，**如：文档中的词频统计

John likes to watch movies. Mary likes too.

John also likes to watch football games.

  - 字典

{"John": 1, "likes": 2, "to": 3, "watch": 4, "movies": 5, "also": 6, "football": 7, "games": 8, "Mary": 9, "too": 10}

  - 文档词频特征

[1, 2, 1, 1, 1, 0, 0, 0, 1, 1]

[1, 1, 1, 1, 0, 1, 1, 1, 0, 0]

特征的设计

- 从原始数据中如何设计特征？
  - 关键特征提取： **TF-IDF（词频-逆文档率）**
    - 算法简单高效,工业界用于最开始的**数据预处理**
  - 例如：找到能代表该文档中的**“关键词”**
  - 词频 （TF, Term Frequency）
    - TF = 某个词(特征值)在句子(数据)中出现的频率
  - 逆文档率 （IDF, Inverse Document Frequency）
    - IDF = log(语料库(数据库)的句子(数据)总数 / 包含该词(特征值)的句子(数据)总数)
  - 每个特征值（词）的重要性
    - w_{ij}= tf*idf = TF_{ij}*log(N/DF_i)

$$TF_{w,D_i} = \frac{count(w)}{|D_i|}$$

$$IDF_w = \log \frac{N}{\sum_{i=1}^{N} I(w, D_i)}$$

会有变型

特征的设计

- 从原始数据中如何设计特征？
  - 关键特征提取： **TF-IDF（词频-逆文档率）**
    - 每个特征值（词）的重要性
  - W_{ij} = tf*idf = TF_{ij}*log(N/DF_i)
- 如何找到关键特征（词）？

①根据 TF 可以找到一个句子中的高频词（特征值）（删去无意义的词，如停用词“的”、“是”、“了”等)

②根据 IDF 继续对句子中剩下的词进行权重赋值并排序，在数据库中越常见的词（特征值）权重越小

③根据 TF-IDF 可以得到一个句子（数据）中所有词（特征值）的 TF-IDF 值，进而排序筛选得到每个句子最有代表性的特征（“关键词”）

特征的设计

$$TF_{w,D_i} = \frac{count(w)}{|D_i|} \qquad IDF_w = \log \frac{N}{\sum_{i=1}^{N} I(w, D_i)}$$

- 从原始数据中如何设计特征？—计算 TF-IDF
  - d_1 (A, B, C, C, S, D, A, B, T, S, S, S, T, W, W, … ,)
  - d_2 (C, S, S, T, W, W, A, B, S, B , … ,)
  - d_3(不含ABCDSTW)

文档中词总数=25

TF

	d_1	d_2
A	0.08	0.04
B	0.08	0.08
C	0.08	0.04
D	0.04	0.00
S	0.16	0.12
T	0.08	0.04
W	0.08	0.08

IDF

	IDF
A	0.4
B	0.4
C	0.4
D	1.1
S	0.4
T	0.4
W	0.4

TF-IDF

	d_1	d_2
A	0.032	0.016
B	0.032	0.032
C	0.032	0.016
D	0.044	0.000
S	0.064	0.048
T	0.032	0.016
W	0.032	0.032

特征的设计

- 从原始数据中如何设计特征？
  - **TF-IDF（词频-逆文档率）** w_{ij}= tf*idf = TF_{ij}*log(N/DF_i)
- 优点
  - 简单快速的词（特征）重要性表示方法，结果比较符合实际情况
  - 应用广泛：不仅限于文本数据
- 缺点
  - 单纯以“词频”衡量一个词的重要性，不够全面，有时重要的词可能出现次数并不多
  - 无法体现词的**位置信息、顺序信息，**出现位置靠前的词与出现位置靠后的词，都被视为重要性相同
  - 无法发现词（特征）的**隐含联系，语义关系**，如同义词等

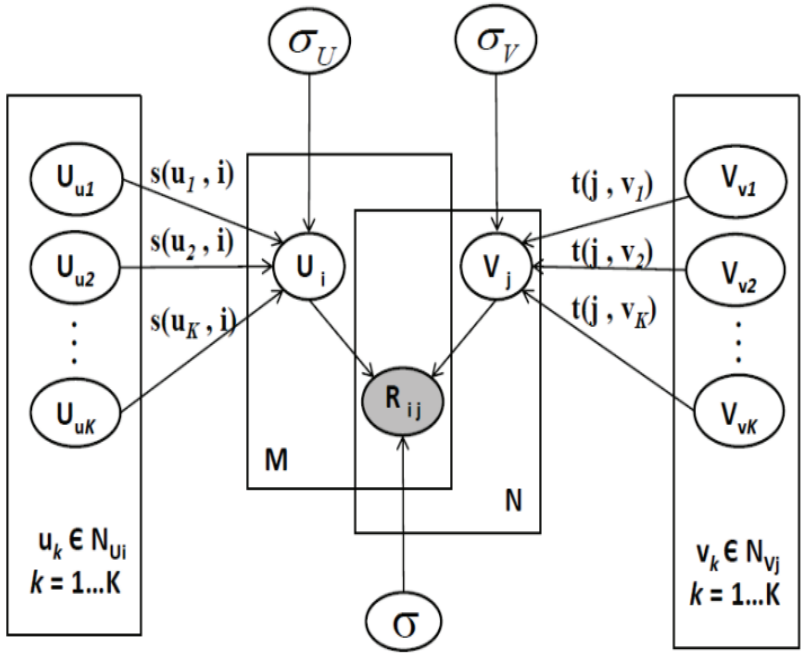
特征的设计

- 从原始数据中如何设计特征？
  - **TF-IDF（词频-逆文档率）—应用**
    - 搜索引擎；关键词提取；文本相似性；文本摘要
  - 推荐系统
    - 可以计算“用户-标签-商品”的特征
    - 用户-标签的TF-IDF

$$P_{il} = tf(i,l) \times \ln(\frac{M}{df(l)})$$

- 用户：i。标签：l。用户总数：M。

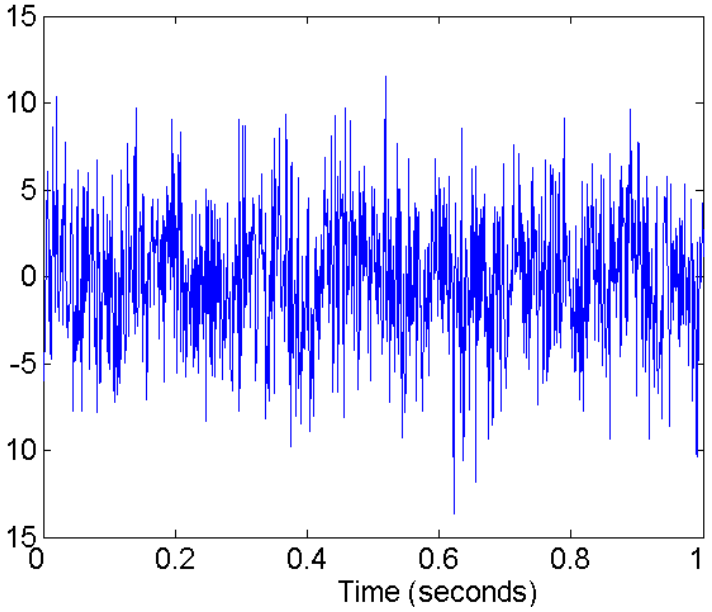
$$s(i,j) = \cos(\vec{i},\vec{j}) = \frac{\vec{i}\cdot\vec{j}}{\|\vec{i}\|_F * \|\vec{j}\|_F}$$



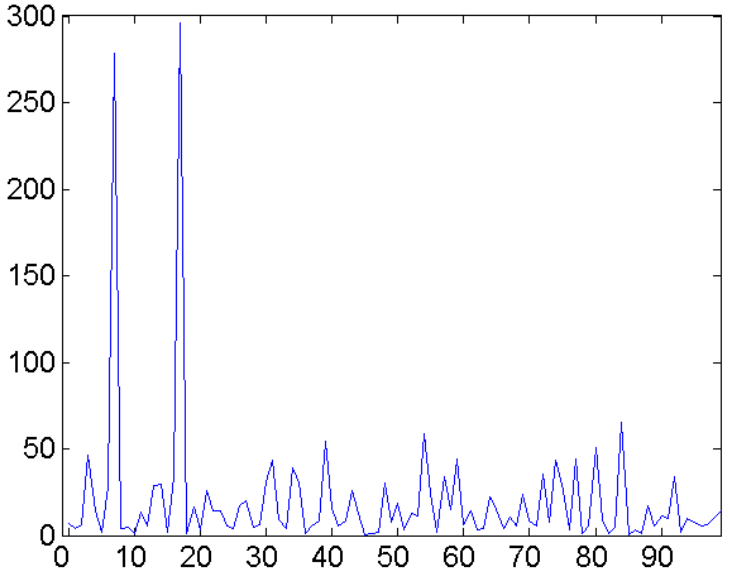
Le Wu, Enhong Chen, Qi Liu, Leveraging Tagging for Neighborhood-aware Probabilistic Matrix Factorization. CIKM'2012

特征的设计

- 从原始数据中如何设计特征？
  - 函数变换特征
    - 左图是根据两个Sin函数（分别是每秒7个和17个周期），以及一些噪声数据得到的序列图；
    - 右图是由**傅立叶变换**得到了频率图，可以看出变换后成功得到了两个概率最大的频率7和17（其中纵坐标是振幅，即概率值）



Two Sine Waves(正弦波) + Noise



Frequency

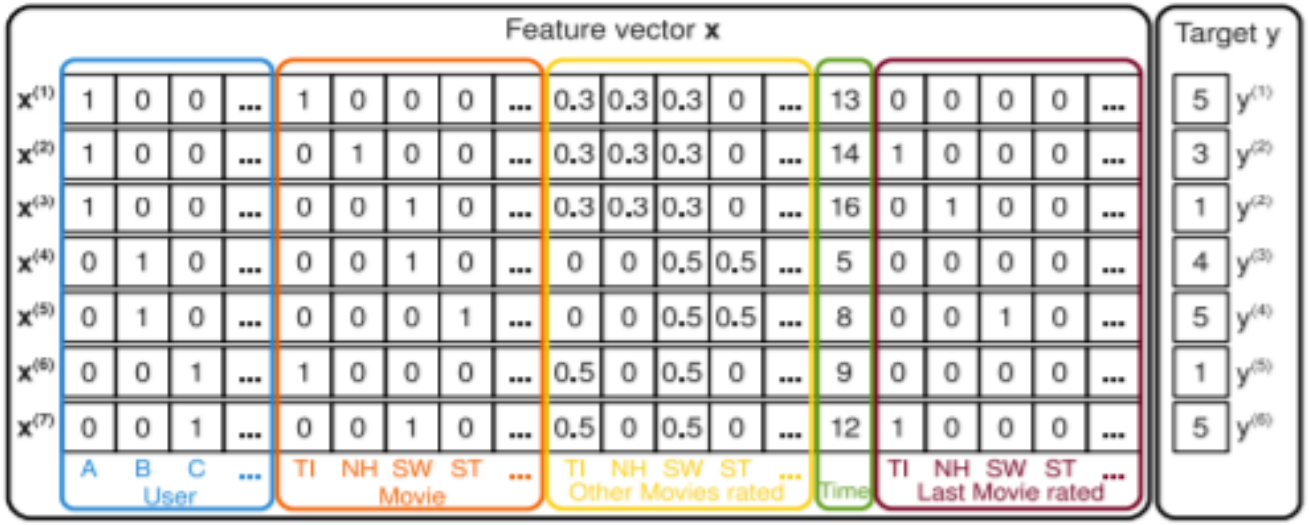
特征的设计

*ratings.csv - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
userId,movieId,rating,timestamp
1,1,4.0,964982703
1,3,4.0,964981247
1,6,4.0,964982224
1,47,5.0,964983815
1,50,5.0,964982931
1,70,3.0,964982400
1,101,5.0,964980868
```

- 从原始数据中如何设计特征？
  - **特征组合：构造高阶特征**
  - 上述所有构造的特征均可以：两两、三三、… 进行组合
    - Factorization Machine (2012)



$$S = \{(\text{A}, \text{TI}, 2010\text{-}1, 5), (\text{A}, \text{NH}, 2010\text{-}2, 3), (\text{A}, \text{SW}, 2010\text{-}4, 1), (\text{B}, \text{SW}, 2009\text{-}5, 4), (\text{B}, \text{ST}, 2009\text{-}8, 5), (\text{C}, \text{TI}, 2009\text{-}9, 1), (\text{C}, \text{SW}, 2009\text{-}12, 5)\}$$

$$\tilde{y}(x) = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} w_{ij} x_i x_j$$

特征的设计

- 举例：第二届“中国高校计算机大赛-大数据挑战赛”
- 赛题描述/数据：http://bdc.saikr.com/vse/bdc/2017
- 该赛题的求解目标是利用数据分析将人工的鼠标轨迹和代码生成的鼠标轨迹区分开来。这里的鼠标轨迹是指一种完成一种验证手段——拖动滑块到指定区域时鼠标的轨迹。



- 原始数据格式：一系列连续点的

例如：(2,3,4),(2,5,6)(4,3,7) (4,3)，该轨迹中含有三个点的坐标，以(x,y, time)的时间表示，终点坐标为(4,3)

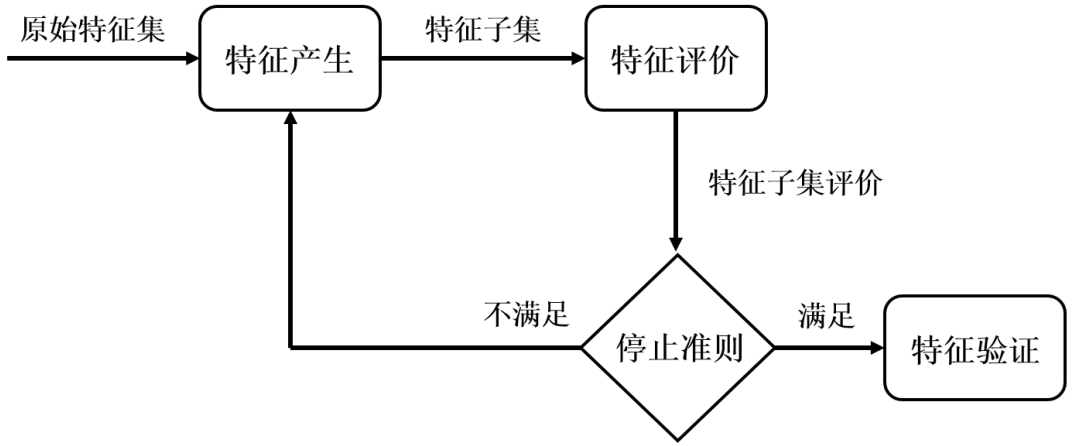


特征的设计

- 从原始数据中如何设计特征？
  - **基本特征的提取**
    - 轨迹运动数据的统计值：运动速度/加速度/角加速度/角速度的均值/极值/最值/中位数 等
    - 轨迹的描述：运动在x轴方向是否为单向，曲线平滑程度， 等
  - **创建新的特征**
    - 基本特征的简单二元运算， 加/减/乘/除/平方和/和平方/倒数和
    - 运动数据在某一维上的偏导
    - 领域专家知识

特征的选择

- 如何挑选有效的特征（**Subset Selection问题**）？
  - 在实际应用中，特征的数量往往比较多，可能会存在不相关的特征
  - 特征数量越多，分析特征、训练模型所需要的时间就越长，同时容易引起**“维度灾难”**，使得模型更加复杂
  - **特征选择**通过剔除不相关的特征或冗余的特征来**减少特征数量**，从而简化了模型并且提升了模型的泛化能力



特征选择的过程



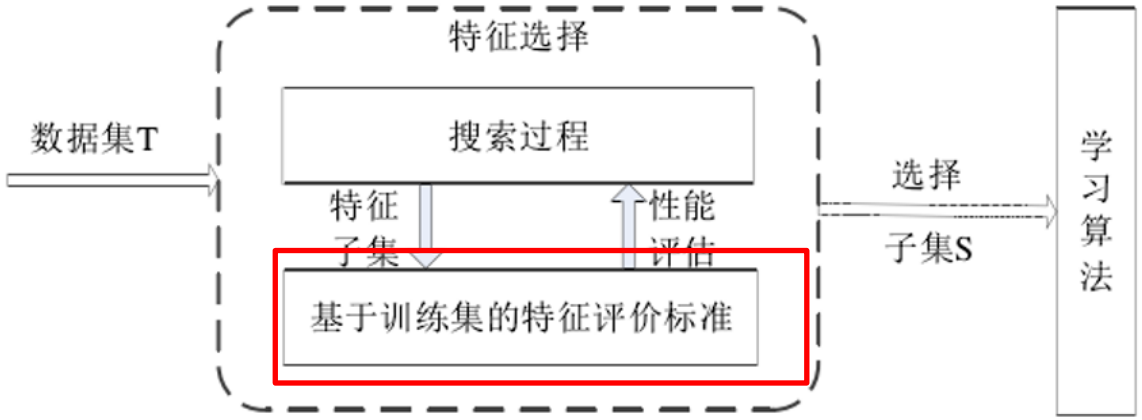
特征子集评价

如何评价特征子集？

不同的特征选择算法不仅对特征子集评价标准不同，有的还需要结合后续的学习算法模型。因此根据特征选择中子集评价标准和后续算法的结合方式主要分为**过滤式(Filter)**、**封装式(Wrapper)** 和**嵌入式(Embedded)**三种

1. 过滤式(Filter)评价策略方法

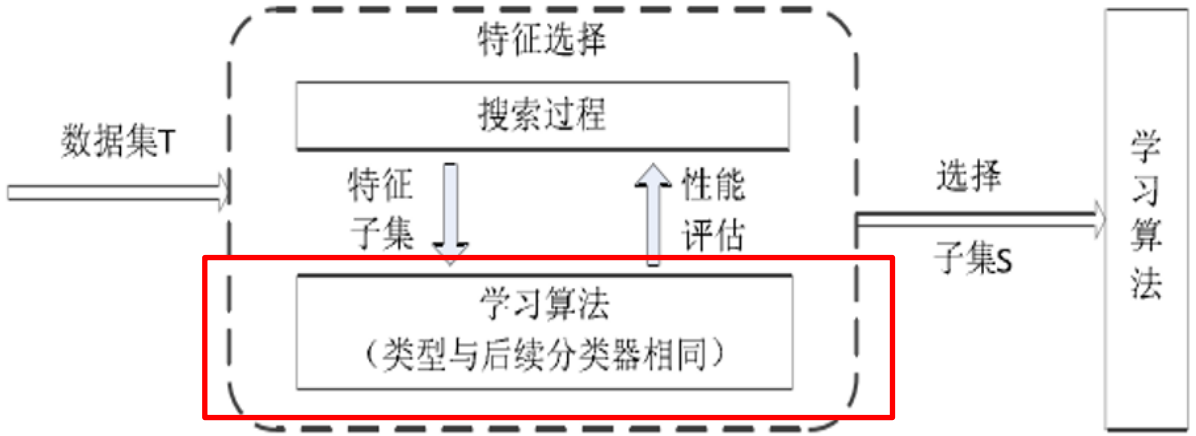
- 独立于后续的学习算法模型来分析数据集的固有的属性
- 采用一些基于信息统计的启发式准则来评价特征子集
- 启发式的评价函数：距离度量、信息度量、依赖性度量、一致性度量



特征子集评价

- 如何评价特征子集？
  - **2. 封装式(Wrapper)评价策略方法**
    - 将特征选择作为学习算法一个组成部分，需要结合后续的学习算法，并直接将学习算法的分类性能作为特征重要性的评价标准
    - 直接使用**分类器的性能作为评价的标准**，选出来的特征子集对分类一定有最好的性能

相对于Filter 选择方法，Wrapper 方法所选择的特征子集的规模要小得多，有利于关键特征的辨识，模型的分类性能更好。但Wrapper 方法泛化能力较差，当改变学习算法时，需要针对该学习算法重新进行特征选择，算法的计算**复杂度高**.



特征子集评价

- 如何评价特征子集？
  - **3. 嵌入式(Embedded)评价策略方法**
    - 特征选择算法嵌入到学习和分类算法中，**特征选择是算法模型中的一部分**
    - 算法模型训练和特征选择同时进行，互相结合。即，算法具有自动进行特征选择的功能

