

# 文本数据处理基础 与 $k$ -近邻算法

雷至祺

2022.05.05



# 目录

- 1 文本数据处理基础
- 2  $k$ -近邻 ( $k$ -NN) 算法
- 3 实验任务与要求



# 目录

- **1 文本数据处理基础**
  - 单词表示: One-hot编码
  - 文档表示: One-hot矩阵、Bag-of-Words模型、TF-IDF矩阵
- 2  $k$ -近邻 ( $k$ -NN) 算法
- 3 实验任务与要求



# 文本数据处理：编码

- 为什么需要对文本进行编码？
  - 图像由多个像素点构成，像素值之间是可计算的。
  - 与图像不同，文本一般很难直接被进行计算，所以我们需要对文本进行编码。

“Lion is the king of the jungle.”



“The tiger hunts in this forest.”

“Everybody loves New York.”



# 目录

- 1 文本数据处理基础
  - 单词表示: One-hot编码
  - 文档表示: One-hot矩阵、Bag-of-Words模型、TF-IDF矩阵
- 2  $k$ -近邻 ( $k$ -NN) 算法
- 3 实验任务与要求



# 单词的one-hot编码

## One-hot编码:

- 文档中每一个词都是一个V维的向量（V是词表大小），其中向量中只有对应词表的位置是1，其余都是0。

例如，给定文本数据集如下：

- 文档1： 不错 酒店 舒服 服务 态度 很好
- 文档2： 酒店 服务 热情 希望 服务
- 文档3： 苹果 手机 不错

按词的出现顺序构造词表：

- 不错 酒店 舒服 服务 态度 很好  
热情 希望 苹果 手机

则每个词的one-hot编码如下：

- 不错： [1, 0, 0, 0, 0, 0, 0, 0, 0, 0]
- 酒店： [0, 1, 0, 0, 0, 0, 0, 0, 0, 0]
- 舒服： [0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
- 服务： [0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
- .....



# 目录

- **1 文本数据处理基础**
  - 单词表示: One-hot编码
  - **文档表示: One-hot表示、Bag-of-Words模型、TF-IDF矩阵**
- 2  $k$ -近邻 ( $k$ -NN) 算法
- 3 实验任务与要求













# TF-IDF

- TF-IDF(term frequency - inverse document frequency)
- $TF\text{-}IDF = TF * IDF$ , 可以把IDF理解为TF的一个权重值

$$\text{tfidf}_{i,j} = \text{tf}_{i,j} \times \text{idf}_i$$

## 思考题2：IDF数值有什么含义？TF-IDF数值有什么含义？

[illegible][illegible]



# 目录

- 1 文本数据处理基础
- 2  $k$ -近邻 ( $k$ -NN) 算法
  - 有监督学习
    - $k$ -NN处理分类问题
    - $k$ -NN处理回归问题
    - $k$ -NN参数设置
- 3 实验任务与要求



# $k$ -NN与有监督学习

- $k$ -NN是有监督的机器学习模型
- 有监督学习的基本步骤：上课—考试
  - 给出带标签的训练数据
  - 用训练数据训练模型至一定程度
  - 用训练好的模型预测不带标签的数据的标签
- 常见的有监督学习问题：
  - 分类问题：预测离散值的问题（如预测明天是否会下雨）
  - 回归问题：预测连续值的问题（如预测明天气温是多少度）

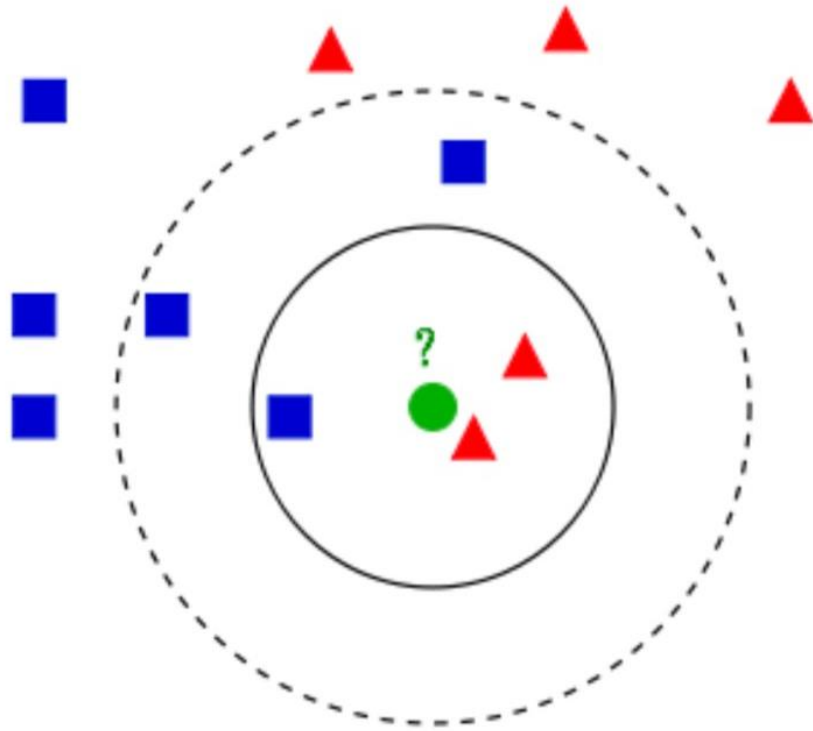


# 目录

- 1 文本数据处理基础
- **2  $k$ -近邻 ( $k$ -NN) 算法**
  - 有监督学习
  - **$k$ -NN处理分类问题**
  - $k$ -NN处理回归问题
  - $k$ -NN参数设置
- 3 实验任务与要求



# $k$ -NN处理分类问题



半径大小 表示 K值大小

- $k$ -nearest neighbours **classifier**:

$$f(q) = \text{maj} \left( g \left( \Phi_{X,k}(q) \right) \right)$$

- 其中:
  - $\Phi_{X,k}(q)$ : 返回训练集 $X$ 中距离 $q$ 最近的 $k$ 个样本
  - $g(\cdot)$ : 返回 (训练) 样本的标签
  - $\text{maj}(\cdot)$ : 返回众数





# $k$ -NN处理分类问题： 例子

- 给定文本的情感分类任务：
  - 输入： 文本
  - 输出： 类标签
  - 分类： 多数投票原则

Document number	The sentence words	emotion
train 1	I buy an apple phone	happy
train 2	I eat the big apple	happy
train 3	The apple products are too expensive	sadnesss
test 1	My friend has an apple	?



# $k$ -NN处理分类问题： 步骤

Document number	The sentence words	emotion
train 1	I buy an apple phone	happy
train 2	I eat the big apple	happy
train 3	The apple products are too expensive	sadnesss
test 1	My friend has an apple	?

## 1. 处理成one-hot矩阵

Document number	I	buy	an	apple	...	friend	has	emotion
train 1	1	1	1	1	...	0	0	happy
train 2	1	0	0	1	...	0	0	happy
train 3	0	0	0	1	...	0	0	sadness
test 1	0	0	1	1	...	1	1	?



# $k$ -NN处理分类问题： 步骤

Document number	I	buy	an	apple	...	friend	has	emotion
train 1	1	1	1	1	...	0	0	happy
train 2	1	0	0	1	...	0	0	happy
train 3	0	0	0	1	...	0	0	sadness
test 1	0	0	1	1	...	1	1	?

2. 相似度计算： 计算test1与每个train的距离

• 欧氏距离：  $d(train1, test1) = \sqrt{(1-0)^2 + (1-0)^2 + \dots + (0-1)^2} = \sqrt{6}$ ;

$$d(train2, test1) = \sqrt{(1-0)^2 + (1-0)^2 + \dots + (0-1)^2} = \sqrt{8}$$

$$d(train3, test1) = \sqrt{(0-0)^2 + (0-0)^2 + \dots + (0-1)^2} = \sqrt{9}$$

(也可以使用其他距离度量方式)

3. 类别计算： 最相似的k个样本之标签的众数

- 若k=1, test1的标签即为train1的标签happy;
- 若k=3, test1的标签为train1,train2,train3的标签中数量较多的, 即为happy。



# 目录

- 1 文本数据处理基础
- **2  $k$ -近邻 ( $k$ -NN) 算法**
  - 有监督学习
  - $k$ -NN处理分类问题
  - **$k$ -NN处理回归问题**
  - $k$ -NN参数设置
- 3 实验任务与要求



# $k$ -NN处理回归问题： 例子

- 输入： 文本
- 输出： 属于某一类的**概率**（连续值）

Document number	The sentence words	the probability of happy
train 1	I buy an apple phone	0.8
train 2	I eat the big apple	0.6
train 3	The apple products are too expensive	0.1
test 1	My friend has an apple	?



# $k$ -NN处理回归问题： 步骤

Document number	The sentence words	the probability of happy
train 1	I buy an apple phone	0.8
train 2	I eat the big apple	0.6
train 3	The apple products are too expensive	0.1
test 1	My friend has an apple	?

- 1. 处理成one-hot矩阵

Document number	I	buy	an	apple	...	friend	has	probability
train 1	1	1	1	1	...	0	0	0.8
train 2	1	0	0	1	...	0	0	0.6
train 3	0	0	0	1	...	0	0	0.1
test 1	0	0	1	1	...	1	1	?



# $k$ -NN处理回归问题：步骤

Document number	1	buy	an	apple	...	friend	has	probability
train 1	1	1	1	1	...	0	0	0.8
train 2	1	0	0	1	...	0	0	0.6
train 3	0	0	0	1	...	0	0	0.1
test 1	0	0	1	1	...	1	1	?

- 2. 相似度计算：计算test1与每个train的距离
- 3. 根据相似度加权：选取TopK个训练数据把距离的倒数作为权重，计算test1属于该标签的概率

$$P(\text{test1 is happy}) = \frac{\text{train1 probability}}{d(\text{train1}, \text{test1})} + \frac{\text{train2 probability}}{d(\text{train2}, \text{test1})} + \frac{\text{train3 probability}}{d(\text{train3}, \text{test1})}$$
$$= 0.47$$

- 思考题3：为什么是倒数？如果要求同一测试样本的各个情感概率总和为1，应该如何处理？



# 目录

- 1 文本数据处理基础
- **2  $k$ -近邻 ( $k$ -NN) 算法**
  - 有监督学习
  - $k$ -NN处理分类问题
  - $k$ -NN处理回归问题
  - **$k$ -NN参数设置**
- 3 实验任务与要求





# $k$ -NN参数设置

- 采用不同的距离度量方式（见下一页）
- 通过验证集对参数（ $k$ 值）进行调优
  - 如果 $k$ 值取的过大，学习的参考样本更多，会引入更多的噪音，所以可能存在欠拟合的情况；
  - 如果 $k$ 值取的过小，参考样本少，容易出现过拟合的情况
  - 关于 $k$ 的经验公式：一般取 $k = \sqrt{N}$ ， $N$ 为训练集实例个数，大家可以尝试一下
- 归一化

Name	Formula	Explain
Standard score	$X' = \frac{X - \mu}{\sigma}$	$\mu$ is the mean and $\sigma$ is the standard deviation
Feature scaling	$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$	$X_{min}$ is the min value and $X_{max}$ is the max value



# 不同距离度量方式

- 距离公式:

$L_p$  距离(所有距离的总公式):

- $$L_p(x_i, x_j) = \left\{ \sum_{l=1}^n |x_i^{(l)} - x_j^{(l)}|^p \right\}^{\frac{1}{p}}$$

- $p = 1$ : 曼哈顿距离;
- $p = 2$ : 欧氏距离, 最常见。

**例 3.1** 已知二维空间的 3 个点  $x_1 = (1, 1)^T$ ,  $x_2 = (5, 1)^T$ ,  $x_3 = (4, 4)^T$ , 试求在  $p$  取不同值时,  $L_p$  距离下  $x_1$  的最近邻点。

**解** 因为  $x_1$  和  $x_2$  只有第一维的值不同, 所以  $p$  为任何值时,  $L_p(x_1, x_2) = 4$ 。而

$$L_1(x_1, x_3) = 6, \quad L_2(x_1, x_3) = 4.24, \quad L_3(x_1, x_3) = 3.78, \quad L_4(x_1, x_3) = 3.57$$

于是得到:  $p$  等于 1 或 2 时,  $x_2$  是  $x_1$  的最近邻点;  $p$  大于等于 3 时,  $x_3$  是  $x_1$  的最近邻点。 ■

- 余弦相似度:

$$\cos \left( \vec{A}, \vec{B} \right) = \frac{\vec{A} \cdot \vec{B}}{|\vec{A}| |\vec{B}|}, \text{ 其中 } \vec{A} \text{ 和 } \vec{B} \text{ 表示两个文本特征向量;}$$

距离 = 1 - 相似度

- 余弦值作为衡量两个个体间差异的大小的度量
- 数值越大, 表示两个文本差距越小, 请大家自行脑补两个向量余弦值



# 不同距离度量方式

- Jaccard相似度

- 度量两个集合的相似性，给定集合 $A$ 和 $B$ ，则：

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

- 测试文档的单词集合为 $A$ ，训练文档的单词集合为 $B$ 。
    - 两篇文档共享的单词越多，Jaccard相似度越大。
    - 可以基于文档的one-hot表示计算，也可以直接基于文档的单词集合计算。



# 有监督学习：数据集划分

训练集

测试集

训练集

验证集

测试集





# 总结

- 第一步：文本处理
  - One-hot、BoW、TF-IDF
- 第二步：kNN分类与回归
  - 在训练集上训练，在验证集上调参
    - k值
    - 距离度量（欧氏距离、余弦距离、Jaccard距离等）
  - 上述过程反复进行，直到得到较优参数
  - 在测试集上得到预测的结果



# $k$ -NN算法的效率

- 对每个测试样本的预测，都需要访问并检索训练集中所有的样本。
- 假设训练集有 $N$ 个样本，测试集有 $M$ 个样本，每个样本是一个 $V$ 维的向量。
- 如果使用线性搜索的话，那么 $k$ -NN的时间花销就是 $O(N * M * V)$ 。
- 效率优化：kd-树。



# 目录

- 1 文本数据处理基础
- 2  $k$ -近邻 ( $k$ -NN) 算法
- **3 实验任务与要求**
  - 任务1:  $k$ -NN分类
  - 任务2:  $k$ -NN回归
  - 实验提交与验收



# 目录

- 1 文本数据处理基础
- 2  $k$ -近邻 ( $k$ -NN) 算法
- **3 实验任务与要求**
  - **任务1:  $k$ -NN分类**
  - **任务2:  $k$ -NN回归**
  - 实验提交与验收





# 任务1: $k$ -NN分类

- 使用 $k$ -NN进行分类任务
- 数据目录为classification\_dataset, 其中train\_set用于训练, validation\_set是验证集
- 通过调节 $k$ 值、不同距离度量等参数来筛选**准确率**最好的一组参数, 并将该过程记录在实验报告中
- 在测试集test\_set上应用该参数做预测, 输出结果保存为“**学号\_姓名拼音\_KNN\_classification.csv**”
  - 文件内部格式参考“19881234\_Sample\_KNN\_classification.csv”



# 分类实验数据介绍

```
Words (split by space),label  
europe retain trophy with big win,joy  
senate votes to revoke pensions,sad
```

- 数据一共有两列，其中每一列用英文逗号隔开。
- 第一列为文档，词之间用空格隔开；
- 第二列是标签。



# 回归评测指标：相关系数

- 相关系数是研究变量之间线性相关程度的量。在回归问题的应用场景下，用于计算实际概率向量以及预测概率向量之间的相似性

$$COR(X, Y) = \frac{cov(X, Y)}{\sigma_X \sigma_Y} = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2 \sum_{i=1}^n (Y_i - \bar{Y})^2}}$$

- 在情感分布预测问题中，我们在验证集上有所有文档预测得到的概率值，也有真实的概率值。先分别计算六个维度上的真实概率值和预测概率值的相关系数，然后对六个维度取平均计算得到最终相关系数



## 任务2: $k$ -NN回归

- 使用 $k$ -NN进行回归任务
- 数据目录为regression\_dataset, 其中train\_set用于训练, validation\_set是验证集
- 通过调节 $k$ 值、不同距离度量等参数来筛选**相关系数**最好的一组参数, 并将该过程记录在实验报告中
  - 这一步可以通过使用“**validation相关度评估.xlsx**”文件辅助验证, 也可自己写代码。
    - validation相关度评估.xlsx文件用于在验证集上评估结果, 使用相关系数, 大家把验证集上的预测结果, 粘贴在Predict工作表中, 右边会产生结果。Standard工作表不要修改内容。
- 在测试集test\_set上应用该参数做预测, 输出结果保存为“**学号\_姓名拼音\_KNN\_regression.csv**”
  - 文件内部格式参考“19881234\_Sample\_KNN\_regression.csv”
- **提示: 注意检查6种概率相加要等于1。**



# 回归实验数据介绍

```
Words (split by space),anger,disgust,fear,joy,sad,surprise  
europe retain trophy with big win,0,0,0,0.8721,0,0.1279  
senate votes to revoke pensions,0.1625,0,0.225,0,0.4375,0.175
```

- 数据一共有七列，其中每一列用英文逗号隔开。
- 第一列为文档，词之间用空格隔开；
- 第二到七列是标签对应的概率。



# 目录

- 1 文本数据处理基础
- 2  $k$ -近邻 ( $k$ -NN) 算法
- **3 实验任务与要求**
  - 任务1:  $k$ -NN分类
  - 任务2:  $k$ -NN回归
  - **实验提交与验收**



# 思考题

- 1. 朴素贝叶斯中，伯努利模型和多项式模型分别有什么优缺点？
  - 2. IDF数值有什么含义？TF-IDF数值有什么含义？
  - 3. kNN中为什么是距离的倒数？如果要求同一测试样本的各个情感概率总和为1，应该如何处理？
- 
- 不管选做哪些任务，上面三题都需要回答。



# 实验5

邹树和曹利刚

- 第11周：朴素贝叶斯
- 第12周：文本处理与 $k$ -NN
- 实验任务：文本情感分析
  - 在朴素贝叶斯分类、 $k$ -NN分类与 $k$ -NN回归中，三者至少完成一项；
  - 鼓励尝试多种算法及算法中的不同策略/参数，并进行结果对比分析；
  - 完成一份实验报告，注意实验报告要求。





# 实验提交

- 作业名称：实验5
- 截止时间：5月11日 23:00
- 本次实验提交样例：压缩包20\*\*\*\*\*\_wangxiaoming.zip, 内含：
  - 20\*\*\*\*\*\_wangxiaoming.pdf
  - /code: 文件夹, 内含所有实验代码并附上readme
  - /result: 文件夹, 内含实验结果 (根据完成情况, 至少包含一个)
    - 20\*\*\*\*\*\_wangxiaoming\_NB\_classification.csv
    - 20\*\*\*\*\*\_wangxiaoming\_KNN\_classification.csv
    - 20\*\*\*\*\*\_wangxiaoming\_KNN\_regression.csv



# 实验评价（从实验5开始）

- 实验质量（30'）
    - 代码正确性
    - 工作量（数量、质量）
  - 实验报告（70'）
    - 原理与关键代码（40%）
    - 实验结果展示（40%）
    - 思考题（20%）
  - 创新点与优化（+10'）
- 每次评选2~4份优秀报告，直接加实验课总平时成绩1~2分



# 实验报告要求

- 实验报告可使用Word/Markdown/Latex等撰写，以pdf格式提交，可参考课程网站（超算习堂）中的模板与实验报告编写建议，应包含如下内容：
  - (1) 算法原理：用**自己的话**解释一下自己对算法/模型的理解（不可复制PPT和网上文档内容），鼓励附上伪代码或者流程图（注意简洁规范清晰，包含关键步骤）
  - (2) **关键**代码展示：可截图或贴文本并对每个模块进行解释，包括代码+**注释**
  - (3) 创新点&优化：如果有的话，**分点**列出自己的创新点（加分项）
  - (4) 实验结果展示：用数据测试自己的模型是否**准确**
  - (5) 评测指标展示：基础模型的**指标**&(4)中对应分点优化后的模型指标+**分析**
  - (6) 思考题：PPT上写的思考题（如有）一般需要在报告最后写出解答
  - (7) 参考资料：参考的文献、博客、网上资源等需规范引用，否则涉嫌抄袭



# 实验验收

- 验收日期：5月5日/5月12日实验课
- 验收形式：助教上传一个小数据集到Q群中，下载好然后课上验收时当场跑程序，TA会根据结果判断算法是否正确。

Q & A

# 附录



# 文件读写

C++:

<http://blog.csdn.net/kingstar158/article/details/6859379/>

Java:

<http://blog.csdn.net/jiangxinyu/article/details/7885518/>

Python:

<http://www.cnblogs.com/allenblogs/archive/2010/09/13/1824842.html>



# 字符串分割

C++:

<http://blog.csdn.net/glt3953/article/details/11115485>

Java:

[http://blog.sina.com.cn/s/blog\\_b7c09bc00101d3my.html](http://blog.sina.com.cn/s/blog_b7c09bc00101d3my.html)

Python:

[http://blog.sina.com.cn/s/blog\\_81e6c30b01019wro.html](http://blog.sina.com.cn/s/blog_81e6c30b01019wro.html)