



文本数据集简单处理 & KNN (K最近邻)

丁诚





Outline

1/ 实验课程要求

2/ 文本数据集简单处理：TF-IDF矩阵

3/ KNN最近邻算法

4/ 附录：相关资料以及提交方式



实验课程内容：

- 由助教讲解实验内容
- 验收前一次的实验内容（包括公式推导、代码解释、现场运行代码产生结果）

实验课程要求：

- 实验需要一定的数学基础以及编程基础（公式的推导以及代码的实现）
- 禁止抄袭（代码和实验报告都禁止抄袭，若被发现后果会较为严重）



Contents

1/ 文本数据集简单处理：TF-IDF矩阵

2/ KNN最近邻

3/ 相关资料以及提交方式



简单的文本数据集处理-编码

为什么我们需要文本编码？ - 可计算性

“Lion is the king of the jungle.”



“The tiger hunts in this forest.”

“Everybody loves New York.”

简单的文本数据集处理-编码

文本编码的几个基本步骤:

- 1 将文本划分成独立的词汇
- 2 建立不重复词汇表
- 3 根据文本以及词汇表建立文本的向量表达

文本编号	词汇表					
训练文本1	苹果	手机	好用	销售		
训练文本2	市民	买	手机	手机		
训练文本3	市民	觉得	苹果	手机	贵	好用

不重复词汇表

贵	好用	觉得	买	苹果	市民	手机	销售
---	----	----	---	----	----	----	----



简单的文本数据集处理-OneHot矩阵

使用一个向量表示一篇文章，向量的长度为词汇表的大小，1表示**存在**对应的单词，0表示**不存在**

文本编号	词汇表					
训练文本1	苹果	手机	好用	销售		
训练文本2	市民	买	手机	手机		
训练文本3	市民	觉得	苹果	手机	贵	好用



贵	好用	觉得	买	苹果	市民	手机	销售
---	----	----	---	----	----	----	----



	贵	好用	觉得	买	苹果	市民	手机	销售
训练文本1	0	1	0	0	1	0	1	1
训练文本2	0	0	0	1	0	1	1	0
训练文本3	1	1	1	0	1	1	1	0



简单的文本数据集处理-TF(Term Frequency)

TF (Term Frequency)：向量的每一个值标志对应的词语出现的次数归一化后的频率。

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

TF矩阵

	贵	好用	觉得	买	苹果	市民	手机	销售
训练文本1	0	1/4	0	0	1/4	0	1/4	1/4
训练文本2	0	0	0	1/4	0	1/4	2/4	0
训练文本3	1/6	1/6	1/6	0	1/6	1/6	1/6	0



简单的文本数据集处理：TF-IDF 矩阵

IDF: 逆向文件频率; 假设总共有 $|D|$ 篇文章 $\{j: t_i \in d_j\}$ 表示出现了该单词的文章总数, IDF值的计算公式如下:

$$\text{idf}_i = \log \frac{|D|}{|\{j: t_i \in d_j\}|}$$

$$\text{idf}_i = \log \frac{|D|}{1 + |\{j: t_i \in d_j\}|}$$

IDF矩阵

	贵	好用	觉得	买	苹果	市民	手机	销售
IDF	$\log(3/1)$	$\log(3/2)$	$\log(3/1)$	$\log(3/1)$	$\log(3/2)$	$\log(3/2)$	$\log(3/3)$	$\log(3/1)$

思考: IDF 的第二个计算公式中分母多了个1是为什么?



简单的文本数据集处理：TF-IDF 矩阵

TF-IDF：每个元素都是TF与IDF的乘积

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

TF-IDF矩阵

	贵	好用	觉得	买	苹果	市民	手机	销售
训练文本1	0	$(1/4) \times \log(3/2)$	0	0	$(1/4) \times \log(3/2)$	0	$(1/4) \times \log(3/3)$	$(1/4) \times \log(3/1)$
训练文本2	0	0	0	$(1/4) \times \log(3/1)$	0	$(1/4) \times \log(3/2)$	$(2/4) \times \log(3/3)$	0
训练文本3	$(1/6) \times \log(3/1)$	$(1/6) \times \log(3/2)$	$(1/6) \times \log(3/1)$	0	$(1/6) \times \log(3/2)$	$(1/6) \times \log(3/2)$	$(1/6) \times \log(3/3)$	0

思考：IDF数值有什么含义？ TF-IDF数值有什么含义？



Contents

1/ 文本数据集简单处理: TF-IDF

2/ **KNN最近邻算法**

3/ 附录: 相关资料及提交方式



KNN最近邻算法

KNN是有监督的机器学习模型

有监督训练的步骤:

- 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



KNN最近邻算法

- 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**）



KNN最近邻算法

k-NN处理分类问题

- 输入：原始文本
- 输出：类标签（happy, sadness...）
- 分类原则：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	?



KNN最近邻算法

步骤1：数据集的特征表示

数据集

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	?

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	?



KNN最近邻算法

步骤2：相似度计算

计算test1与每个train的欧氏距离
(也可以使用其他距离度量方式)

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

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

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

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。



KNN最近邻算法

k-NN处理回归问题

- 输入：原始文本
- 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	?



KNN最近邻算法

步骤1：数据集的特征表示

数据集

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	?

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	?



KNN最近邻算法

步骤2：根据相似度加权

计算test1与每个train的距离，选取TopK个训练数据
把该距离的倒数作为权重，计算test1属于该标签的概率：

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

思考：为什么是倒数呢？

注意：同一测试样本的各个情感概率总和应该为1 如何处理？



KNN最近邻算法

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;
- $p = 2$: 欧式距离, 最常见。



KNN最近邻算法

不同距离度量方式

余弦相似度：

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

余弦距离：

$$\text{dis}(\vec{A}, \vec{B}) = 1 - \cos(\vec{A}, \vec{B}) \geq 0$$

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



KNN最近邻算法

更多实验方法提高准确率

- 采用不同的距离度量方式
- 通过验证集对参数（K值）进行调优
- 对权值进行归一化

Name	Formula	Explain
Standard score	$X' = \frac{X - \mu}{\sigma}$	μ is the mean and σ 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

PS: 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



补充介绍评测指标：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。

$$r(X, Y) = \frac{Cov(X, Y)}{\sqrt{Var[X] Var[Y]}}$$

在情感分类问题中：对于所有文档，计算其在六个维度上的真实概率值和预测概率值的向量相关系数，再对六个维度取平均计算得到最终相关系数。



KNN最近邻算法

训练集 验证集 测试集的区别

数据类型	有无标签	作用
训练集(training set)	有	用来训练模型或确定模型参数的，如k-NN中权值的确定等。 相当于平时练习。
验证集(validation set)	有	用来确定网络结构或者控制模型复杂程度的参数，修正模型。 相当于模拟考试。
测试集(test set)	无	用于检验最终选择最优的模型的性能如何。 相当于期末考试。



KNN最近邻算法

训练集 验证集 测试集的使用

- 一个典型的划分是训练集占总样本的50%，而其它各占25%，三部分都是从样本中随机抽取。
- 本次实验分类任务和回归任务都出了训练集，验证集和测试集。
- validation.xlsx文件用于在验证集上进行结果的评估，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。



Contents

1/ 文本数据集简单处理: TF-IDF

2/ KNN最近邻算法

3/ 附录: 相关资料及提交方式


TF-IDF实验任务

1、将数据集“semeval”的数据表示成TF-IDF矩阵，并保存为“学号_姓名拼音_TFIDF.txt”文件。

标准输出：

(不重复词向量

按照出现顺序构成)



TFIDF.txt - 记事本

	文件(F)	编辑(E)	格式(O)	查看(V)	帮助(H)		
0	-0.0719205	0	0.101366	0	0	0	0
0	-0.143841	0	0	0.101366	0	0	
0	-0.047947	0	0	0	0.0675775	0.0675775	



SemEval 实验数据分析

- 每一行即一篇文本，每一行的组成成分示例：
- 文本编号，与下一项以tab隔开
- 总情感权重、各情感权重，各项之间以空格隔开，与下一项以tab隔开
- 文本内容，单词之间以空格隔开



KNN实验任务

• 分类（使用准确率进行衡量结果）

1. 使用KNN处理分类问题。在验证集上，通过调节K值、选择不同距离等方式得到一个准确率最优的模型参数，并将该过程记录在实验报告中。
2. 在测试集上应用步骤1中得到的模型参数（K，距离类型等），将输出结果保存为“学号_姓名拼音_KNN_classification.csv”，
文件内部格式参考'16351234_Sample_KNN_classification.csv"

• 回归（使用相关系数进行衡量结果）

1. 使用KNN处理回归问题，在验证集上，通过调节K值、选择不同距离等方式得到一个相关系数最优的模型参数，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（也可以自己写代码）。
2. 在测试集上应用步骤1中得到的模型参数（K，距离类型等），将输出结果保存为“学号_姓名拼音_KNN_regression.csv”，
文件内部格式参考'16351234_Sample_KNN_regression.csv"

提示：请记得检查你们6种情感概率相加是否为1



实验报告内容

- (1) 算法原理：用**自己的话**解释一下自己对模型和算法的理解（不可复制网上文档内容）
- (2) 伪代码：伪代码或者流程图（注意简洁清晰）
- (3) 关键代码截图：代码+**注释**
- (4) 创新点&优化：**分点**列出自己的创新点
- (5) 实验结果展示：用小数据测试自己的模型是否准确
- (6) 评测指标展示：模型基础的指标+ **(4) 中对应分点**优化后的模型指标
- (7) 编程语言只可以使用c++,java,python中的一种，并且python不能使用现有机器学习高级库，否则扣分。
- (8) 代码会进行查重，若相似度高于不可接受阈值，按抄袭处理不接受任何反驳。



实验提交声明

1. 提交作业FTP: <ftp://118.31.11.174> 账号及密码: student

2. 提交格式:

"lab1"文件夹下有:

"code"文件夹: 存放代码, 将tfidf和KNN代码打包成一个压缩包, 命名为"学号_姓名拼音_lab1.zip"

"report"文件夹: 存放实验报告 (PDF格式), 命名为"学号_姓名拼音_lab1.pdf",

3. 实验报告提交DDL: 9月5号23点前



验收声明

1. 验收日期：下一周验收
2. 验收形式：在每个时段上课前会上传一个小数据集到群上，提前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。验收结束可离开教室



文本读写

C++:

<http://www.cnblogs.com/ifeiyun/articles/1573134.html>

Java:

<https://www.cnblogs.com/zhuocheng/archive/2011/12/12/2285290.html>

Python:

<https://www.liaoxuefeng.com/wiki/0014316089557264a6b348958f449949df42a6d3a2e542c000/001431917715991ef1ebc19d15a4afdace1169a464eecc2000>



字符串分割

C++:

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

Java:

http://blog.sina.com.cn/s/blog_b7c09bc00101d3my.html

Python:

http://blog.sina.com.cn/s/blog_81e6c30b01019wro.html



THANKS

