



# 实验二：K近邻与朴素贝叶斯

## ——分类和回归

PPT制作及出题人：  
刘金杨（KNN）  
商家煜（NB）



## 几点说明

1. 本次实验报告DDL为10月18日，但是10月12，13日要验收KNN
2. 如果报告有新版本，在文件名后面加后缀，例如“15351234\_zhangsan\_v1.xx”
3. 代码一定会进行查重，重复率达到不可接受的阈值按抄袭处理，不接受任何反驳。
4. python库只能用numpy（矩阵的运算也可以用）



# 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的理解
2. 伪代码：伪代码或者流程图（注意清晰简洁）
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正确
6. 评测指标展示：基础模型的指标+与第4点中分点对应的优化后的指标（如果有）

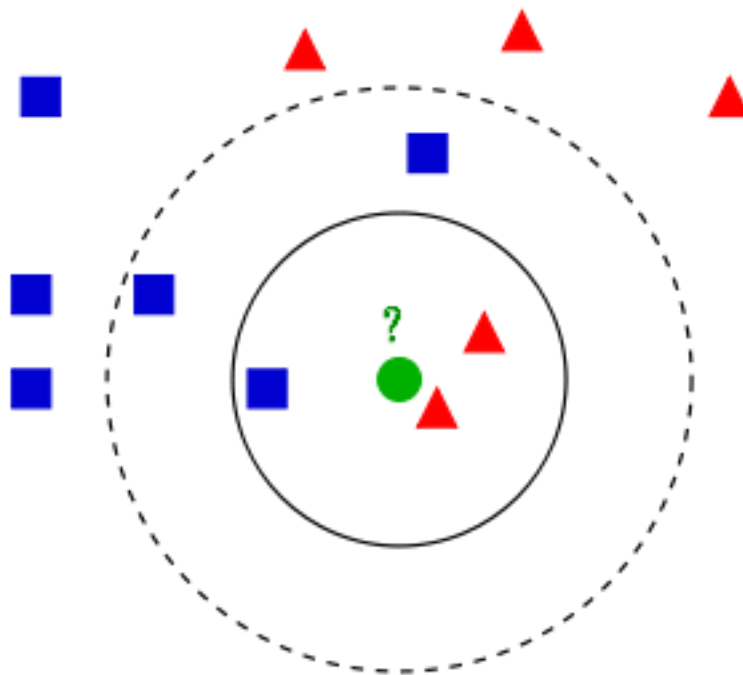


- KNN和NB都是有监督的机器学习模型
- 有监督训练的步骤：
  - 给出带标签的训练数据
  - 用训练数据训练模型至一定程度
  - 用训练好的模型预测不带标签的数据的标签



- 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**）

# k-NN处理分类问题



半径大小 表示 K值大小



# 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	?



# 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	?



## 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。



# 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	?

# 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	?



## 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？**



# 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。

（思考：在矩阵稀疏程度不同的时候，这两者表现有什么区别，为什么？）



# 不同距离度量方式

余弦相似度：

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

- 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，表示两个文本差距越小
- 为负代表差距越大，请大家自行脑补两个向量余弦值。



# 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	$X' = \frac{X - \mu}{\sigma}$	$\mu$ 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

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



# 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 <b>训练模型</b> 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 <b>确定</b> 网络结构或者控制模型复杂程度的 <b>参数</b> ， <b>修正模型</b> 。 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。





# 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。



# 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“15351234\_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“15351234\_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



# Naïve Bayes

BAYES RULE

HOLY GRAIL

A hand-drawn arrow pointing from the 'Holy Grail' oval towards the 'BAYES RULE' text.

REV THOMAS BAYES



# 实验课内容



Example:  $P_{(c)} = 0.01$

Sensitivity -> 真阳性

Test: 90% it is positive if you have cancer

$$P_{(Pos|c)} = 0.9$$

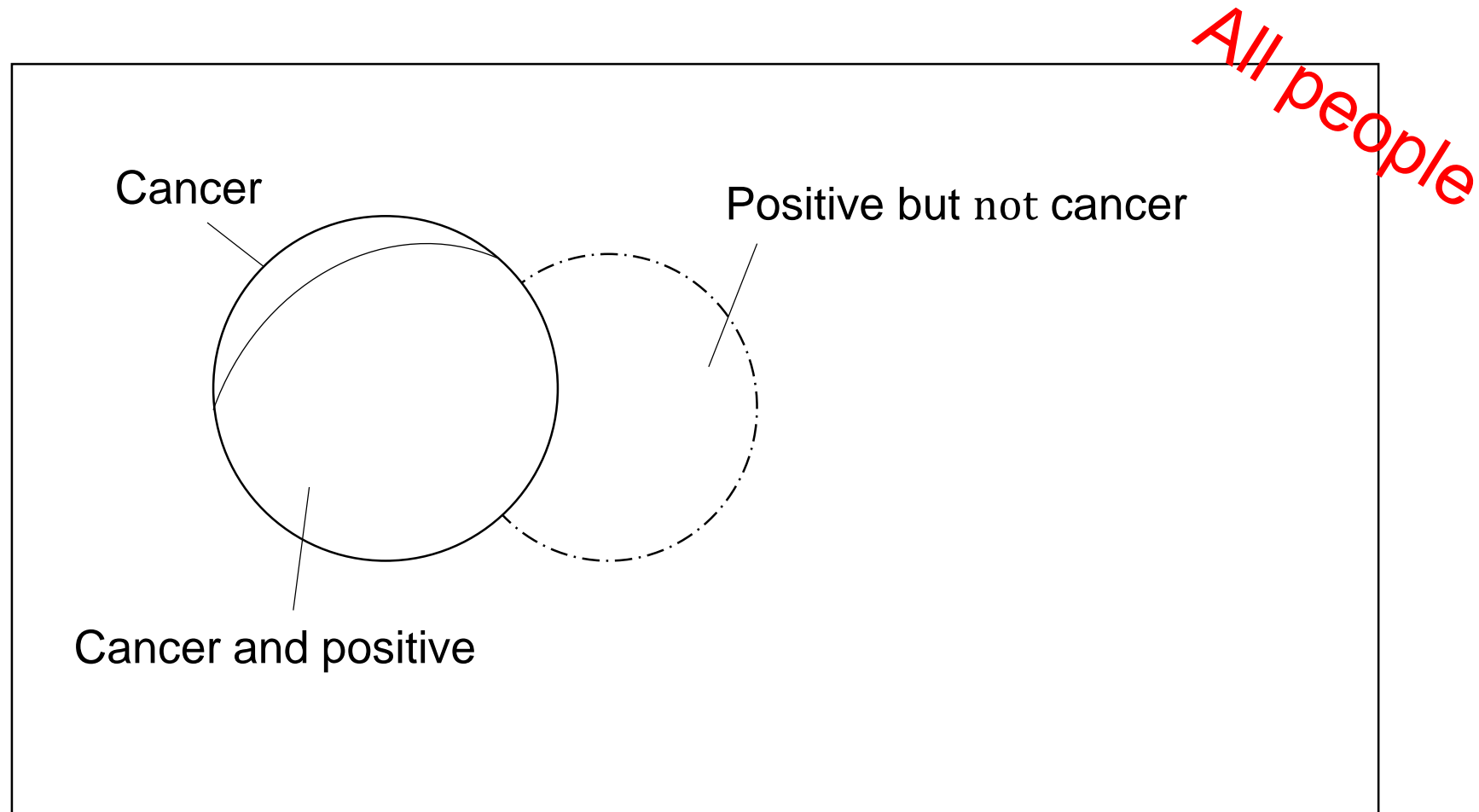
Specificity -> 假阴性

90% it is negative if you don't have cancer

$$P_{(Neg|\neg c)} = 0.9$$

Question: If Test = Positive

Probability of having cancer





Prior:

$$P_{(c)} = 0.01$$

$$P_{(\text{Pos}|c)} = 0.9$$

$$P_{(\text{Neg}|\neg c)} = 0.9$$



$$P_{(\neg c)} = 0.99$$

$$P_{(\text{Pos}|\neg c)} = 0.1$$

Posterior:

$$p(y|x) = \frac{p(x, y)}{p(x)} = \frac{p(x|y)p(y)}{p(x)}$$



Joint:  $P_{(c, pos)} = P_{(c)} * P_{(Pos|c)} = 0.009$

$$P_{(\neg c, pos)} = P_{(\neg c)} * P_{(Pos|\neg c)} = 0.099$$

normalizer:  $P_{(pos)} = P_{(c, pos)} + P_{(\neg c, pos)} = 0.108$

Posterior :  $P_{(c|pos)} = 0.0833$

$$P_{(\neg c|pos)} = 0.9167$$

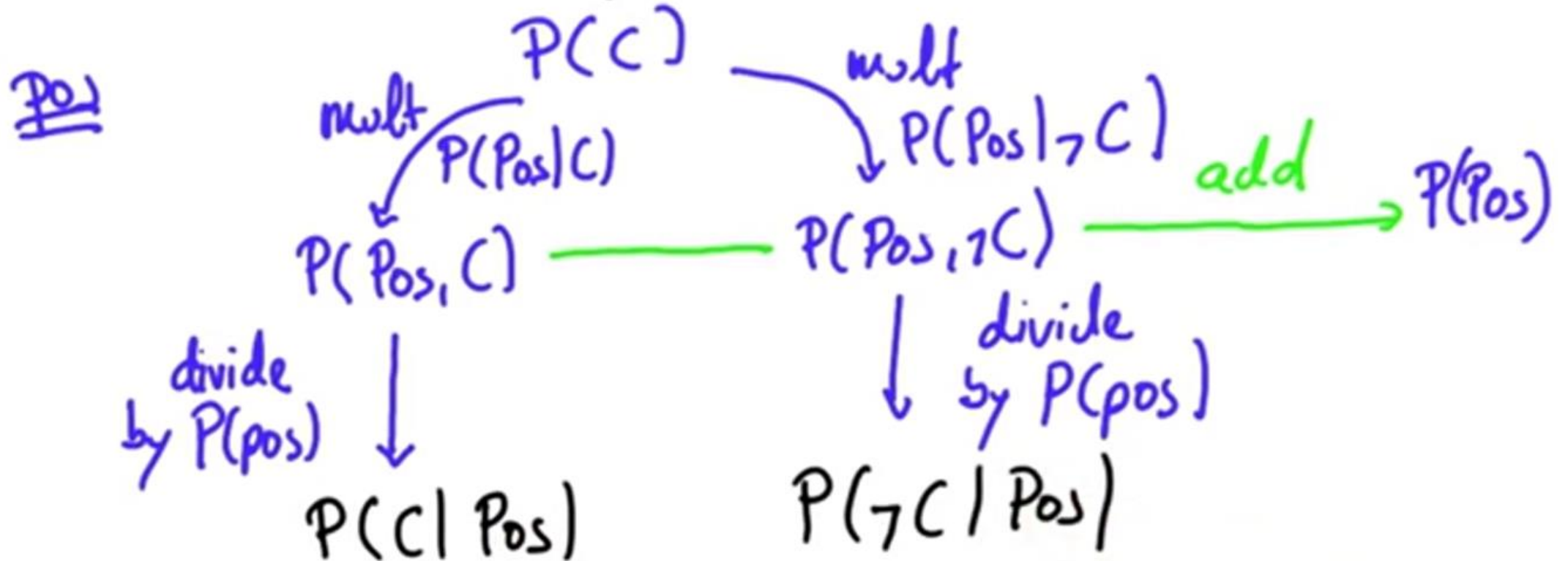




$P(C)$  prior

$P(Pos|C)$  sensitivity

$P(Neg|\neg C)$  specificity





# Bayes rule

$$p(y|x) = \frac{p(x, y)}{p(x)} = \frac{p(x|y)p(y)}{p(x)}$$

where  $p(x)$  is a **constant** for all classes. Take animal classification as example:  $p(x) = p(x|y=0)p(y=0) + p(x|y=1)p(y=1)$

Thus NB is to find

$$\begin{aligned} y &= \arg \max_y p(y|x) = \arg \max_y \frac{p(x, y)}{p(x)} \\ &= \arg \max_y p(x|y)p(y) \end{aligned}$$



# Classification for Naïve Bayes



# Classification for Naïve Bayes

**Bernoulli Model ( 伯努利模型 )** : a document is represented by a feature vector with **binary** elements taking value 1 if the corresponding word is present in the document and 0 if the word is not present.

$$p(x_k|e_i) = \frac{n_{e_i}(x_k)}{N_{e_i}} \quad p(e_i) = \frac{N_{e_i}}{N}$$

where  $n_{e_i}(x_k)$  is the number of documents of emotion  $e_i$  in which  $x_k$  is observed, and  $N_{e_i}$  and  $N$  is the number of documents with emotion  $e_i$  and total documents, respectively.



# Classification for Naïve Bayes

**Multinomial Model ( 多项式模型 )** : a document is represented by a feature vector with integer elements whose value is the **frequency** of that word in the document.

$$p(x_k|e_i) = \frac{nw_{e_i}(x_k)}{nw_{e_i}} \quad p(e_i) = \frac{N_{e_i}}{N}$$

where  $nw_{e_i}(x_k)$  is the number of times word  $x_k$  occurs in documents with emotion  $e_i$ , and  $nw_{e_i}$  is the total number of words occurs in documents with emotion  $e_i$ .



# Example

ID	text	class label
1	good,thanks	joy
2	No impressive, thanks	sad
3	Impressive good	joy
4	No, thanks	?



ID	goods	thanks	no	impressive	class label
1	1	1	0	0	joy
2	0	1	1	1	sad
3	1	0	0	1	joy
4	0	1	1	0	?

Bernoulli Model (伯努利模型) :

$$P_{(\text{thanks}|\text{joy})} = 1/2$$

Multinomial Model (多项式模型) :

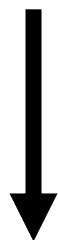
$$P_{(\text{thanks}|\text{joy})} = 1/4$$

思考题：这两个模型分别有什么优缺点



# Classification (多项式模型)

ID	text	class label
1	good,thanks	joy
2	No impressive, thanks	sad
3	Impressive good	joy
4	No, thanks	?



ID	goods	thanks	no	impressive	class label
1	1	1	0	0	joy
2	0	1	1	1	sad
3	1	0	0	1	joy
4	0	1	1	0	?

Target function:

$$p(\text{joy}|d_4) = p(\text{joy}) \cdot p(d_4|\text{joy})$$

$$p(\text{sad}|d_4) = p(\text{sad}) \cdot p(d_4|\text{sad})$$

Example:

$$\begin{aligned} p(\text{joy}|d_4) &= p(d_4|\text{joy}) \cdot p(\text{joy}) \\ &= p(\text{" thanks" , " no" }|\text{joy}) \cdot p(\text{joy}) \\ &= p(\text{" thank" }|\text{joy}) \cdot p(\text{" no" }|\text{joy}) \cdot p(\text{joy}) \\ &= \frac{1}{4} \times 0 \times \frac{2}{3} = 0 \end{aligned}$$



# Regression for Naïve Bayes





# Example

Documnt	sentence	joy	sad
train1 (d1)	Step by step, we will succeed.	0.9	0.1
train2 (d2)	We step on shit.	0.3	0.7
test1 (d3)	We succeed.	?	?

Figure: Example of documents

<b>X</b>	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	emotion	
Document	step	by	we	succeed	on	shit	will	joy	sad
train1 (d1)	0.33	0.17	0.17	0.17	0	0	0.17	0.9	0.1
train2 (d2)	0.25	0	0.25	0	0.25	0.25	0	0.3	0.7
test1 (d3)	0	0	0.5	0.5	0	0	0	?	?

Figure: TF features of documents



# Example

To predict emotion  $e_i$  of test document  $X_3 = (x_3, x_4)$ , we need to estimate:

$$\arg \max_{e_i} p(e_i | X) = \arg \max_{e_i} \sum_{j=1}^M \prod_{k=1}^{K'} p(x_k | e_i, d_j) p(d_j, e_i)$$

$$\begin{aligned} p(\text{joy} | X_3) &\propto p(x_3 | \text{joy}, d_1) p(x_4 | \text{joy}, d_1) p(d_1, \text{joy}) \\ &\quad + p(x_3 | \text{joy}, d_2) p(x_4 | \text{joy}, d_2) p(d_2, \text{joy}) \\ &= 0.17 \times 0.17 \times 0.9 + 0.25 \times 0 \times 0.3 = 0.02601 \end{aligned}$$

$$\begin{aligned} p(\text{sad} | X_3) &\propto p(x_3 | \text{sad}, d_1) p(x_4 | \text{sad}, d_1) p(d_1, \text{sad}) \\ &\quad + p(x_3 | \text{sad}, d_2) p(x_4 | \text{sad}, d_2) p(d_2, \text{sad}) \\ &= 0.17 \times 0.17 \times 0.1 + 0.25 \times 0 \times 0.7 = 0.00289 \end{aligned}$$



# Example

Normalize the posterior distribution:

$$p'(joy|X_3) = \frac{0.02601}{0.02601 + 0.00289} = 0.9$$

$$p'(sad|X_3) = \frac{0.00289}{0.02601 + 0.00289} = 0.1$$



# Laplace Smoothing

Notice that if the word  $x_k$  in test document does not occur in the training set,  $p(x_k|d_j, e_i)$  will be zero and thus cause the resulting value becoming 0.

Solution: **Laplace Smoothing!** ( 拉普拉斯平滑 )

- regression model:

$$p(x_k|d_j, e_i) = \frac{x_k + 1}{\sum_{k=1}^K x_k + K}$$

- classification model:

$$\text{Bernoulli: } p(x_k|e_i) = \frac{n_{e_i}(x_k) + 1}{N_{e_i} + 2}$$

$$\text{Multinomial: } p(x_k|e_i) = \frac{nw_{e_i}(x_k) + \alpha}{nw_{e_i} + V_{e_i}}$$

$0 < \alpha \leq 1$   
 $n$  为 训练集不重复单词  
总数

where  $nw_{e_i}$  is the total number of words in documents with emotion  $e_i$ , and  $V_{e_i}$  is the number of non-repetitive words with label  $e_i$ .



# Task

- 分类：使用准确率衡量结果，只要求实现多项式模型，可以在创新中实现其他模型。
- 回归：使用相关系数衡量结果，归一化后，使得6种情感概率相加为1
- 必须实现拉普拉斯平滑



# 数据说明

总共 **两个压缩包**：

**classification\_dataset**和**regression\_dataset**

里面分别有三个文件，分别用作**train**、**validation**、**test**

从外，**regression**另外提供了一个相似度评估文件。



# 提交文件

- 总共 **两个结果文件**：  
“**学号\_姓名拼音\_NB\_classification.csv**”，  
“**学号\_姓名拼音\_NB\_regression.csv**”，  
打包，正确命名后上交ftp。  
文件内部格式**参考**“15351234\_Sample\_NB\_classification.csv”  
文件内部格式**参考**“15351234\_Sample\_NB\_regression.csv”
- 代码文件 **尽量** 是写在一个代码文件里，，如果有多个代码文件，打包，正确命名后上交ftp。
- 报告中要有 **所有任务** 的 **结果展示**，报告提交PDF版本，请勿提交word文件，避免排版混乱。



# 两次实验共提交文件

-----FTP

-----班级

-----报告

-----一份报告

-----结果

-----包含两个实验4个结果的压缩包（采用zip格式，“发送到”压缩）

-----代码

-----包含两个实验2份代码的压缩包（采用zip格式，“发送到”压缩）

如果对此次实验题目有疑问，请联系刘金杨和商家煜。





# 注意事项

## 1、作业提交地址

FTP地址：<ftp://39.108.233.34>

登录用户名与密码均为 student

提交文件夹的名字是 labx\_yyyyddmmend, x为第几次实验, yyyyddmmend是指截止日期, 比如 20171018end

## 2、命名方式

查询“实验课须知”, 实验报告, 所有代码文件以及结果文件都需要上交。

3、编程语言可用 C++, python, matlab, java等, **不能使用现成库 (如 sklearn 等)**, 否则扣分

## 4、提交截止时间

**2017年10月18日23: 59: 59前**