

武 汉 纺 织 大 学

《机器学习》课程报告

题目： 基于朴素贝叶斯的邮件分类器

成 绩: _____

学 号: 1704210616

姓 名: PEAK

班 级: 计科 11700

指导教师: XXXX

报告日期: 2019 年 ** 月 ** 日

1 我是一级标题

在这里书写内容即可在这里书写内容即可在这里书写内容即可在这里书写内容即可在这里书写内容即可在这里书写内容即可在这里书写内容即可在这里书写内容即可在这里书写内容即可，这样写的段落是不会自用换行的，具体要像下面这样。

向上面一行，没有
是不能换行的，但是这样换行是不能够首行缩进的。
空一行也能完成换行

2 我是一级标题

这一节重点介绍小标题、符号、代码的使用。就像这个样子，就限定了字体的使用范围，如果要强调，就能够像我这个样子。如果想使用脚注就使用后坳面这个¹，我还想再脚注中添加公式²

2.1 我是二级标题

这里能够写二级子目录的东西。这里能够写二级子目录的东西。这里能够写二级子目录的东西。这里能够写二级子目录的东西。这里能够写二级子目录的东西。这里能够写二级子目录的东西。这里能够写二级子目录的东西。

2.2 我是二级标题

2.2.1 我是三级级标题

我是段落内容。我是段落内容。我是段落内容。我是段落内容。我是段落内容。我是段落内容。我是段落内容。我是段落内容。我是段落内容。我是段落内容。我是段落内容。我是段落内容。我是段落内容。我是段落内容。

2.2.2 我是三级级标题

我是段落内容。我是段落内容。我是段落内容。我是段落内容。我是段落内容。我是段落内容。我是段落内容。我是段落内容。我是段落内容。我是段落内容。我是段落内容。我是段落内容。我是段落内容。我是段落内容。

¹我是脚注的内容

²比如放置一个条件概率公式, $P(X|Y)$

$$P(X|Y) = \frac{P(XY)}{P(Y)}$$

我是段落内容。我是段落内容。我是段落内容。我是段落内容。我是段落内容。
我是段落内容。

2.3 我是二级标题

这里将介绍怎么编写公式:

我们能够简单的输入问本:

$$P(XY) = P(Y|X) * P(X) \quad (1)$$

使用 *equation* 这个就是输入公式的环境, 就能输入独占一行的公式, 使用 $$ 包括只能输入, 与文本在一行, $$ 也能独占, 但是不编号。

使用 *ref* 就能实现对公式的引用, 将 (1) 变换为条件概率公式, 如下:

$$P(Y|X) = \frac{P(XY)}{P(X)}$$

这样就完成了公式的输入。其中 *frac* 输入公式的命令, 在导航栏中就能找到所有的这样的命令, 就能完成所有公式的输入了. 如果问题还是蛮大就去看看别人的博客, 进行公式入门就能够非常好的显示公式了。

2.4 我是二级标题

添加代码的时候就只需要按照下面的方式添加代码就好了:

```
1 def viterbi(obs, states, start_p, trans_p, emit_p):
2     V = [{}] #tabular
3     path = {}
4     for y in states: #init
5         V[0][y] = start_p[y] * emit_p[y].get(obs[0],0)
6         path[y] = [y]
7     for t in range(1,len(obs)):
8         V.append({})
9         newpath = {}
10        for y in states:
11            (prob,state ) = max([(V[t-1][y0] * trans_p[y0
12                               ].get(y,0) * emit_p[y].get(obs[t],0) ,y0)
                               for y0 in states if V[t-1][y0]>0])
        V[t][y] =prob
```

Symbol	Command	Symbol	Command	Symbol	Command
α	<code>\alpha</code>	θ	<code>\theta</code>	o	<code>o</code>
β	<code>\beta</code>	ϑ	<code>\vartheta</code>	π	<code>\pi</code>
γ	<code>\gamma</code>	ι	<code>\iota</code>	ϖ	<code>\varpi</code>
δ	<code>\delta</code>	κ	<code>\kappa</code>	ρ	<code>\rho</code>
ϵ	<code>\epsilon</code>	λ	<code>\lambda</code>	ϱ	<code>\varrho</code>
ε	<code>\varepsilon</code>	μ	<code>\mu</code>	σ	<code>\sigma</code>
ζ	<code>\zeta</code>	ν	<code>\nu</code>	ς	<code>\varsigma</code>
η	<code>\eta</code>	ξ	<code>\xi</code>	τ	<code>\tau</code>
υ	<code>\upsilon</code>	ϕ	<code>\phi</code>	φ	<code>\varphi</code>
χ	<code>\chi</code>	ψ	<code>\psi</code>	ω	<code>\omega</code>

图 1: 小写希腊字母

```

13         newpath[y] = path[state] + [y]
14         path = newpath
15         (prob, state) = max([(V[len(obs) - 1][y], y) for y in
16                             states])
17         return (prob, path[state])

```

3 这一节重点讲解一下怎么插入图片与表格

3.1 插入图片

插入图片就像这个样子，使用命令 `\includegraphics[width=10cm]{image.png}` 设置的是显示的大小，`{}` 中显示的是显示图片的路径，文件名只能是英文字母组成！

同样的道理，使用 `label` 做标记，使用 `ref` 就能够引用，比如你看图 (1) 就是的。

图片默认显示在当前页面的最上方。

3.2 表格

插入表格就麻烦多了：`p{2cm} < {centering}` 表示居中，显示长度为 `2cm`，`caption{}` 显示的是表名，`&` 表示的是表中的竖线分割。`hline` 显示了横线分割这里故意显示了一个超大的表格，给大家看看到底是怎么做的。

表 1: NoSQL 和关系数据库的简单比较

比较标准	RDBMS	NoSQL	备注
数据库原理	完全支持	部分支持	RDBMS 有关系代数理论作为基础,NoSQL 没有统一的理论基础
数据规模	大	超大	RDBMS 很难实现横向扩展,纵向扩展的空间也比较有限,性能会随着数据规模的增大而降低,NoSQL 可以很容易通过添加更多设备来支持更大规模的数据
数据库模式	固定	灵活	RDBMS 需要定义数据库模式,严格遵守数据定义和相关约束条件,NoSQL 不存在数据库模式,可以自由灵活定义并存储各种不同类型的数据
查询效率	快	简单查询快,复杂查询慢 ³	RDBMS 借助于索引机制可以实现快速查询(包括记录查询和范围查询),很多 NoSQL 数据库没有面向复杂查询的索引,虽然 NoSQL 可以,使用 MapReduce 来加速查询,但是,在复杂查询方面的性能仍然不如 RDBMS
一致性	强一致性	弱一致性	RDBMS 严格遵守事务 ACID 模型,可以保证事务强一致性,很多 NoSQL 数据库放松了对事务 ACID 四性的要求,而是遵守 BASE 模型,只能保证最终一致性
数据完整性	容易实现	很难实现	任何一个 RDBMS 都可以很容易实现数据完整性,比如通过主键或者非空约束来实现实体完整性,通过主键、外键来实现参照完整性,通过约束或者触发器来实现用户自定义完整性,但是,在 NoSQL 数据库却无法实现
扩展性	一般	好	RDBMS 很难实现横向扩展,纵向扩展的空间也比较有限,NoSQL 在设计之初就充分考虑了横向扩展的需求,可以很容易通过添加廉价设备实现扩展

³可以实现高效的简单查询,但是不具备高度结构化查询等特性,复杂查询的性能不尽人意

可用性	好	很好	RDBMS 在任何时候都以保证数据一致性为优先目标，其次才是优化系统性能，随着数据规模的增大，RDBMS 为了保证严格的一致性，只能提供相对较弱的可用性，大多数 NoSQL 都能提供较高的可用性
标准化	是	否	RDBMS 已经标准化 (SQL) ,NoSQL 还没有行业标准，不同的 NoSQL 数据库都有自己的查询语言，很难规范应用程序接口,StoneBraker 认为：NoSQL 缺乏统一查询语言，将会拖慢 NoSQL 发展
技术支持	高	低	RDBMS 经过几十年的发展，已经非常成熟，Oracle 等大型厂商都可以提供很好的技术支持,NoSQL 在技术支持方面仍然处于起步阶段，还不成熟，缺乏有力的技术支持
可维护性	复杂	复杂	RDBMS 需要专门的数据库管理员 (DBA) 维护,NoSQL 数据库虽然没有 DBMS 复杂，也难以维护

4 小结

这里我简单总结下：我会使用列表显示的我的总结部分，你们可以学习下：

- 文章中显示了对文本、公式、标题、脚注进行了演示与说明，可以照葫芦画瓢，模仿编写这个
- 在公式上大家还是最好能够多看几篇博客，熟悉公式命令之后就就非常好编程了
- 图片、表格就按照我写的模仿，相关的说明在文中已经提到了

最后演示一下有序列表，可以在伪代码的部分使用：

1. 初始化随机变量
2. 计算均值
3. 计算方差
4. ...

这是最简单的子文档了，在我当前的这对标签之写文档之后就能够显示在主文档中了