

# 第14周学习记录

Allan

2020 年 12 月 13 日

## 目录

<b>1</b>	<b>12月7日</b>	<b>1</b>
1.1	LaTeX . . . . .	2
1.2	线性代数 . . . . .	2
<b>2</b>	<b>12月8日</b>	<b>2</b>
2.1	线性代数 . . . . .	2
2.2	python . . . . .	2
<b>3</b>	<b>12月9日</b>	<b>3</b>
3.1	python . . . . .	3
<b>4</b>	<b>12月10日</b>	<b>3</b>
4.1	python . . . . .	3
<b>5</b>	<b>12月11日</b>	<b>4</b>
5.1	LaTeX . . . . .	4

## 1 12月7日

其实全文都是从这个.md文件里面搬过来的，主要为了实践一下latex的用法

### 1.1 LaTeX

下载并安装了LaTeX，学习了部分基本格式。

### 1.2 线性代数

复习并做了两套综合卷，迎接明天的期末考试。

## 2 12月8日

### 2.1 线性代数

学习并复习了线性代数

通过观看MIT线性代数复习并巩固了以下知识：

矩阵乘法(四种方法)：常规、列方法、行方法、列乘行。

逆矩阵：行列式不为零(非奇异矩阵)才存在，行列式为零时(奇异矩阵)不存在逆矩阵。

LU分解：将一个矩阵分为上三角阵与下三角阵相乘。

矩阵转置，向量空间，列空间和零空间( $AX=0$ 时X的所有解的集合所形成的空间)。

今日考试把基础解系和通解概念弄反了(反思)

### 2.2 python

学习了NumPy模块(python数据分析手册)：

(引用时一般import numpy as np便于使用)

了解了python中数据的存储方式：是用c语言编写的利用指针和结构体进行存储的。

优点：是动态的，灵活性较高。(甚至可以在一个数组中存多种类型数据)

缺点：每个结构体都包含完整的信息，在很多时候是多余的。

而numpy中虽然数组类型是固定的(一般情况下只能是同类型数据)，但是更高效。

学习了NumPy中的数组：

了解了数组的属性：数组大小(所有元素的总长度)、形状(如 $3*4$ )，数据类型(int,float等)，字符长度等

对数组进行索引、切片(与python基础语法基本相同)

## 3 12月9日

### 3.1 python

NumPy中, 对多维数组进行切片, 以获取行和列, 此时的切片数组是原数组的视图, 若对其进行修改, 会使原数组相应位置也发生改变(非副本视图)。

数组的复制(numpy.copy)(副本)。

利用numpy.reshape将数组变形, 如:单行或单列数组变成矩阵。

利用numpy中concatenate, vstack和hstack等进行数组的拼接。(其实用concatenate都可以实现, 但后者更简洁):

```
numpy.vstack([array1,array2])
numpy.concatenate([array1,array2])
#以上两个等价
numpy.hstack([array1,array2])
numpy.concatenate([array1,array2],axis = 1)
#以上两个等价
```

以及数组的分裂(split), 分裂是以输入的值为下标作为另一个新数组的首元素,

如:

```
x = [1,2,3,4]
x1,x2 = np.split(x,[2])
此时x1 = [1,2], x2 = [3,4]
```

以上的对数组操作可以混合使用, 灵活使用以满足不同的需求。

## 4 12月10日

### 4.1 python

NumPy模块的学习:

python中一些循环耗时过长, 而用向量的方式进行计算可以有效地提高代码运行效率。

了解了numpy中的通用函数(包括四则运算、绝对值、三角函数以及指数对数等)

对数组进行运算，实际上就是分别对数组中每一个元素进行运算，再返回一个新的(运算后的)数组。

利用out可以将运算后数组指定输出到想要储存的位置(包括数组切片的视图，从而修改数组的部分元素)：

```
x = numpy.arange(4) #创建一个数组
y = numpy.empty(4) #再创建一个空的等长数组
#y = numpy.empty(len(x)) 也可以这样
np.multiply(x, 10, out=y) #指定输出到y
print(y)
#结果[ 0. 10. 20. 30. 40.]
```

聚合:用reduce(之前学过的高阶函数进行累积计算)返回一个最终计算结果。

外积:(outer)将两组数组所有数的两两组合进行运算，并把所有结果都返回。

如:

```
np.multiply.outer([1,2,3], [1,2,3])
```

会得到:

```
array([[1,2,3],
       [2,4,6],
       [3,6,9]])
```

过程实际上是:

```
array([[1*1,1*2,1*3],
       [2*1,2*2,2*3],
       [3*1,3*2,3*3]])
```

## 5 12月11日

### 5.1 LaTeX

学习了LaTeX的基础语法，了解了它的常用的控制序列的格式，可以掌握基本排版文章的水平。

基础排版格式：

```
\ documentclass[UTF8]ctexart %文档类别说明
\ usepackageamsmath %可以通过\ usepackage引入宏包，类似import
\ titlehello,world %标题
\ authorAllan %作者名
\ date2020 年 12 月 13 日 %日期(\ today可以生成当前日期)
\ begindocument %与\ end配对，中间为环境，此处环境为(document)
\ maketitle %生成导言区中的标题等
\ tableofcontents %(放在maketitle后)可生成目录
\ section这一级标题
内容1
\ subsection这是二级标题
内容2
\ paragraph段落
内容3
\ subparagraph这是二级段落
内容4
\ subsubsection这是三级标题
内容5
\ enddocument
```

上面通过`\ usepackage`导入(amsmath)宏包以引入数学功能，从而进行数学公式的插入与使用。

具体的数学公式用法待使用时再去学习。