

第一题

(1) 由题意可设 $Y_{i,j}$ 表示编号 i 和编号 j 的小球是否落入同一个盒子

则有

$$Y_{i,j} = \begin{cases} 1, & \text{编号 } i \text{ 和编号 } j \text{ 的小球落入同一个盒子} \\ 0, & \text{其他} \end{cases} \sim B\left(1, \frac{1}{n}\right)$$

$$\therefore X_{m,n} = \sum_{1 \leq i < j \leq m} Y_{i,j}, \text{ 其中共有 } \binom{m}{2} = \frac{m(m-1)}{2} \text{ 项 } Y_{i,j}$$

$$\therefore \mu_{m,n} = \mathbb{E}[X_{m,n}] = \sum_{1 \leq i < j \leq m} \mathbb{E}[Y_{i,j}] = \frac{m(m-1)}{2} \cdot \frac{1}{n} = \frac{m(m-1)}{2n}$$

由方差公式 $D(X + Y) = D(X) + D(Y) + 2\text{Cov}(X, Y)$

$$\therefore \sigma_{m,n}^2 = D(X_{m,n}) = \sum_{1 \leq i < j \leq m} D(Y_{i,j}) + \sum_{1 \leq i < j, 1 \leq k < l, (i,j) \neq (k,l)} \text{Cov}(Y_{i,j}, Y_{k,l})$$

前一项的计算结果为 $\frac{1}{n} \left(1 - \frac{1}{n}\right) \cdot \frac{m(m-1)}{2} = \frac{m(m-1)}{2n} \left(1 - \frac{1}{n}\right)$

对于后一项，分以下两种情况讨论：

- 若 $\{i, j\} \cap \{k, l\} = \emptyset$, 则 $Y_{i,j}$ 与 $Y_{k,l}$ 独立, $\text{Cov}(Y_{i,j}, Y_{k,l}) = 0$
- 若 $\{i, j\} \cap \{k, l\} \neq \emptyset$, 不妨设 $i = k$, 则

$$\begin{aligned} \text{Cov}(Y_{i,j}, Y_{i,l}) &= \mathbb{E}[Y_{i,j} Y_{i,l}] - \mathbb{E}[Y_{i,j}] \mathbb{E}[Y_{i,l}] \\ &= P(Y_{i,j} = 1, Y_{i,l} = 1) - \frac{1}{n} \cdot \frac{1}{n} \\ &= \frac{1}{n} \cdot \frac{1}{n} - \frac{1}{n^2} = 0 \end{aligned}$$

综上所述, 后一项各协方差均为 0, 故 $\sigma_{m,n}^2 = \frac{m(m-1)}{2n} \left(1 - \frac{1}{n}\right)$

(2) 由切比雪夫不等式, 有

$$P(|X_{m,n} - \mu_{m,n}| \geq k) \leq \frac{\sigma_{m,n}^2}{k^2}$$

令 $k = c\sqrt{\mu_{m,n}}$, 则

$$\begin{aligned} P(|X_{m,n} - \mu_{m,n}| \geq c\sqrt{\mu_{m,n}}) &\leq \frac{\sigma_{m,n}^2}{c^2 \mu_{m,n}} \\ \Rightarrow P(|X_{m,n} - \mu_{m,n}| < c\sqrt{\mu_{m,n}}) &\leq \frac{1}{c^2} \cdot \left(1 - \frac{1}{n}\right) \leq \frac{1}{c^2} \end{aligned}$$

证明完毕

第二题

$\because y = e^x$ 是单调递增函数

$$\therefore P(\lambda \geq \alpha) = P(e^\lambda \geq e^\alpha)$$

又 $\because y = a^x$ 在 $a > 0$ 时是不减函数

$$\therefore P(e^\lambda \geq e^\alpha) = P(e^{s\lambda} \geq e^{s\alpha})$$

\therefore 由马尔可夫不等式, 有

$$\begin{aligned} P(e^{s\lambda} \geq e^{s\alpha}) &\leq \frac{\mathbb{E}[e^{s\lambda}]}{e^{s\alpha}} \\ \Rightarrow P(\lambda \geq \alpha) &\leq e^{-s\alpha} \mathbb{E}[e^{s\lambda}] \quad (\text{一式证毕}) \end{aligned}$$

同理一式证明有 $P(\lambda \leq \alpha) = P(e^{s\lambda} \leq e^{s\alpha})$

$\because y = \frac{1}{x}$ 在 $x > 0$ 时是单调递减函数

$$\therefore P(e^{s\lambda} \leq e^{s\alpha}) = P\left(\frac{1}{e^{s\lambda}} \geq \frac{1}{e^{s\alpha}}\right)$$

\therefore 由马尔可夫不等式，有

$$\begin{aligned} P\left(\frac{1}{e^{s\lambda}} \geq \frac{1}{e^{s\alpha}}\right) &\leq \frac{\mathbb{E}\left[\frac{1}{e^{s\lambda}}\right]}{\frac{1}{e^{s\alpha}}} \\ \Rightarrow P(\lambda \leq \alpha) &\leq e^{s\alpha} \mathbb{E}[e^{-s\lambda}] \quad (\text{二式证毕}) \end{aligned}$$

第三题

设 d 维单位球的体积为 $V_d(1) = C_d \cdot 1^d = C_d$ ，其中 C_d 是与维度 d 相关的常数

半径为 $1 - \epsilon$ 的球体积为 $V_d(1 - \epsilon) = C_d(1 - \epsilon)^d$

球表面厚度为 ϵ 的壳层体积为 $V_{shell} = V_d(1) - V_d(1 - \epsilon) = C_d[1 - (1 - \epsilon)^d]$ 。

根据题意，壳层体积占总体积的 99%，即：

$$\frac{V_{shell}}{V_d(1)} = 1 - (1 - \epsilon)^d = 0.99$$

代入 $d = 1000$ ，解得：

$$(1 - \epsilon)^{1000} = 0.01 \Rightarrow 1 - \epsilon = (0.01)^{\frac{1}{1000}}$$

$$\epsilon = 1 - 10^{-\frac{2}{1000}} = 1 - 10^{-0.002} \approx 0.0046$$

故 ϵ 应约为 0.0046

第四题

我认为这道题的核心是让我们直观理解高维空间中的点分布特性，即任意点对距离几乎都集中在一个特定范围内，且它们与原点构成的向量之间的夹角趋近于直角（随机采样的近似正交性）

我写了一段 Python 代码（见附录）来进行采样与距离和夹角的计算，最后将结果用直方图形式进行可视化：

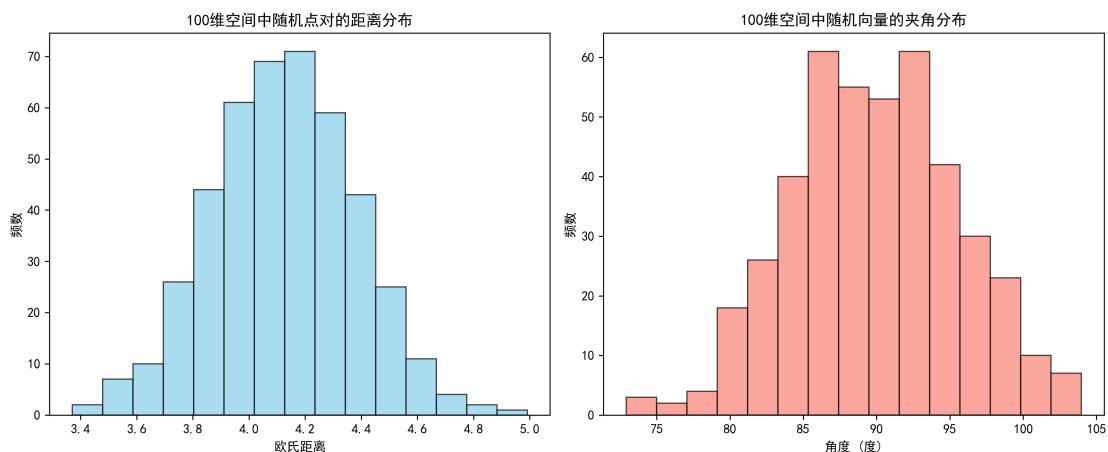


图 1：高维空间中点对距离与夹角分布直方图

从图中可以看出，在高维空间中：

- **距离集中**: 点对之间的距离分布呈现出明显的集中趋势(单峰分布),说明高维空间中任意两点的距离差异较小。
- **角度正交**: 向量之间的夹角高度集中在 90° 附近,说明高维空间中的随机向量倾向于相互正交。

第五题

将事件都写为其首字母用于简化表达

$$\begin{aligned}
 \tilde{P}(E = \text{True} \mid J = \text{True}, M = \text{False}) &= P(E) \sum_b P(B) \sum_a P(A \mid B, E) P(J \mid A) P(M \mid A) \\
 &= P(E) \left[P(B)(P(A \mid B, E) P(J \mid A) P(\neg M \mid A)) \right. \\
 &\quad \left. + P(\neg B)(P(A \mid \neg B, E) P(J \mid \neg A) P(\neg M \mid \neg A)) \right] \\
 &= 0.002 \cdot 0.20385 \\
 &= 0.0004077
 \end{aligned}$$

$$\begin{aligned}
 \tilde{P}(E = \text{False} \mid J = \text{True}, M = \text{False}) &= P(\neg E) \sum_b P(B) \sum_a P(A \mid B, \neg E) P(J \mid A) P(M \mid A) \\
 &= P(\neg E) \left[P(B)(P(A \mid B, \neg E) P(J \mid A) P(\neg M \mid A)) \right. \\
 &\quad \left. + P(\neg B)(P(A \mid \neg B, \neg E) P(J \mid \neg A) P(\neg M \mid \neg A)) \right] \\
 &= 0.998 \cdot 0.051705 \\
 &= 0.0516015
 \end{aligned}$$

$$\begin{aligned}
 \therefore P(E = \text{True} \mid J = \text{True}, M = \text{False}) &= \frac{\tilde{P}(E = \text{True} \mid J = \text{True}, M = \text{False})}{\tilde{P}(E = \text{True} \mid J = \text{True}, M = \text{False}) + \tilde{P}(E = \text{False} \mid J = \text{True}, M = \text{False})} \\
 &= \frac{0.0004077}{0.0004077 + 0.0516015} \\
 &\approx 0.00784
 \end{aligned}$$

$$\begin{aligned}
 P(E = \text{False} \mid J = \text{True}, M = \text{False}) &= \frac{\tilde{P}(E = \text{False} \mid J = \text{True}, M = \text{False})}{\tilde{P}(E = \text{True} \mid J = \text{True}, M = \text{False}) + \tilde{P}(E = \text{False} \mid J = \text{True}, M = \text{False})} \\
 &= \frac{0.0516015}{0.0004077 + 0.0516015} \\
 &\approx 0.99216
 \end{aligned}$$

综上所述,有

$$P(E \mid J, \neg M) \approx 0.00784$$

$$P(\neg E \mid J, \neg M) \approx 0.99216$$

第六题

- (1) 吐槽：知识面确实太广了，这也导致每一个新知识点其实我一开始都不知道它是干什么用的，就需要课后自己去找一些资料看；另外王老师的 PPT 疑似有些过于简略了，如果能再增加一些解释性文字我觉得效果会很好。
- (2) 王老师课上举的很多例子我觉得都很贴切，这一点比大模型给的生硬例子或者为了“直白易懂”而瞎举的例子好多了，但也正如我在吐槽部分所说的，大模型在介绍新知识的立体度上相对来说还是更好一些的，我认为学习这门课时将老师和大模型讲到的内容结合起来才能达到最好的效果。
- (3) 大模型在解答作业题上确实对我帮助很大，尤其是体现在解答一些比较困难的证明题上，它可以对我自己的证明过程进行补充，让逻辑更严密。但为了真正学好这门课，在考试中拿到相对好看的分数，我自己一定要把会的部分都写好后再借助大模型神力。
- (4) 我清晰记得作业 4 第一题我让 deepseek 给我检查的时候它信誓旦旦地把稳态向量计算时候的矩阵放到向量后面，变成求马尔可夫链的稳态向量了。甚至让我怀疑自己写的这个是不是错的，最后因为内鬼 ds 的答案看着更像对的，我毅然决然地把答案改错了，无语。这大模型还是挺会胡说八道的。

下面的矩阵给出了漠河的天气部分信息。天气类型只有三种：雨 (rain)、云 (cloud) 和雪 (snow)。

		Rain	Cloud	Snow
A =	Rain	0.1	0.2	0.8
	Cloud	0.3	0.6	0.1
	Snow	a	b	c

矩阵的列表示概率。以第一列为为例：如果今天下雨，那么明天下雨的概率为 0.1，多云的概率为 0.3。求 a, b, c 来补充矩阵。然后，通过计算 A 的稳态向量，确定长期情况下有雨、多云和下雪天气的概率。

图 2: 作业 4 第一题题目

已思考 (用时 234 秒) >

由矩阵每列概率之和为1, 得:

$$\text{第一列: } 0.1 + 0.3 + a = 1 \Rightarrow a = 0.6$$

$$\text{第二列: } 0.2 + 0.6 + b = 1 \Rightarrow b = 0.2$$

$$\text{第三列: } 0.8 + 0.1 + c = 1 \Rightarrow c = 0.1$$

完整矩阵为:

$$A = \begin{pmatrix} 0.1 & 0.2 & 0.8 \\ 0.3 & 0.6 & 0.1 \\ 0.6 & 0.2 & 0.1 \end{pmatrix}$$

求稳态向量 $\mathbf{v} = (v_R, v_C, v_S)$, 满足 $\mathbf{v}A = \mathbf{v}$ 且
 $v_R + v_C + v_S = 1$:

$$\begin{cases} v_R = 0.1v_R + 0.3v_C + 0.6v_S \\ v_C = 0.2v_R + 0.6v_C + 0.2v_S \\ v_S = 0.8v_R + 0.1v_C + 0.1v_S \end{cases}$$

化简得:

$$0.9v_R - 0.3v_C - 0.6v_S = 0 \quad (1)$$

$$-0.2v_R + 0.4v_C - 0.2v_S = 0 \quad (2)$$

$$-0.8v_R - 0.1v_C + 0.9v_S = 0 \quad (3)$$

由(2)得 $v_S = 2v_C - v_R$, 代入(1):

$$0.9v_R - 0.3v_C - 0.6(2v_C - v_R) = 0 \Rightarrow 1.5v_R - 1$$

则 $v_S = 2v_R - v_R = v_R$ 。

由归一化 $v_R + v_C + v_S = 3v_R = 1 \Rightarrow v_R = \frac{1}{3}$ 。

因此稳态向量为 $\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)$ 。

长期情况下, 雨、多云、下雪的概率均为 $\frac{1}{3}$ 。

答案: $a = 0.6$, $b = 0.2$, $c = 0.1$, 稳态向量为

$\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)$ 。

图 3: deepseek 给出的答案

附录

Listing 1: 高维空间点对距离与夹角计算

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.spatial.distance import pdist

# 设置中文字体
plt.rcParams["font.sans-serif"] = ["SimHei"]
plt.rcParams["axes.unicode_minus"] = False

# 设置参数
dim = 100
n_samples = 30

# 生成数据: [-0.5, 0.5]^100 中的均匀分布, 形状为 (30, 100)
X = np.random.uniform(-0.5, 0.5, (n_samples, dim))

# 计算所有点对之间的距离
distances = pdist(X, metric="euclidean")

# 计算所有点向量之间的夹角
norms = np.linalg.norm(X, axis=1, keepdims=True)
X_normalized = X / norms
cosine_similarity = np.dot(X_normalized, X_normalized.T)
indices = np.triu_indices(n_samples, k=1)
cos_theta = cosine_similarity[indices]
cos_theta = np.clip(cos_theta, -1.0, 1.0)
angles = np.arccos(cos_theta) * (180 / np.pi)

# 绘图
plt.figure(figsize=(12, 5))

# 距离直方图
plt.subplot(1, 2, 1)
plt.hist(distances, bins=15, color="skyblue", edgecolor="black", alpha=0.7)
plt.title(f"100维空间中随机点对的距离分布")
plt.xlabel("欧氏距离")
plt.ylabel("频数")

# 角度直方图
plt.subplot(1, 2, 2)
plt.hist(angles, bins=15, color="salmon", edgecolor="black", alpha=0.7)
plt.title(f"100维空间中随机向量的夹角分布")
plt.xlabel("角度 (度)")
plt.ylabel("频数")

plt.tight_layout()
plt.show()
```