大数据算法第一次作业

2025 年 4 月 15 日

Problem 1. 假设 $Z_i, \ldots, Z_n$ 为 n 个独立同分布的随机变量，并且满足 $\mathbb{E}[Z_i] = 0, Var(Z_i) < \infty$. 定义均值为 $\bar{Z} = \frac{1}{n}\sum_{i=1}^n Z_i$. 证明对于任意 $t > 0$ 有:

$$\mathbb{P}(|\bar{Z}| \geq t) \to 0$$

证明.

$$\text{Var}(\bar{Z}) = \mathbb{E}\left[\left(\frac{1}{n}\sum_{i=1}^n Z_i\right)^2\right] = \frac{1}{n^2}\sum_{i,j\leq n}\mathbb{E}[Z_iZ_j] = \frac{1}{n^2}\sum_{i=1}^n \mathbb{E}[Z_i^2] = \frac{\text{Var}(Z_1)}{n}.$$

特别地，对于任意 $t \geq 0$，我们有

$$\mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^n Z_i\right| \geq t\right) \leq \frac{\text{Var}(Z_1)}{nt^2},$$

因此

$$\mathbb{P}(|\bar{Z}| \geq t) \to 0 \quad \text{对于任意} t > 0.$$

□

Problem 2. 假设 $Z_i, \ldots, Z_n$ 为 n 个独立的有界随机变量，其中 $Z_i \in [a, b]$ 且 $-\infty < a \leq b < \infty$。证明

$$\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^n (Z_i - \mathbb{E}[Z_i]) \geq t\right) \leq \exp\left(-\frac{2nt^2}{(b-a)^2}\right),$$

$$\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^n (Z_i - \mathbb{E}[Z_i]) \leq -t\right) \leq \exp\left(-\frac{2nt^2}{(b-a)^2}\right)$$

对于任意 $t \geq 0$ 成立.

证明.

$$\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^n (Z_i - \mathbb{E}[Z_i]) \geq t\right) = \mathbb{P}\left(\sum_{i=1}^n (Z_i - \mathbb{E}[Z_i]) \geq nt\right)$$

$$\leq \mathbb{E}\left[\exp\left(\lambda\sum_{i=1}^n (Z_i - \mathbb{E}[Z_i])\right)\right] e^{-\lambda nt}$$

$$= \left(\prod_{i=1}^n \mathbb{E}[e^{\lambda(Z_i - \mathbb{E}[Z_i])}]\right) e^{-\lambda nt} \leq \left(\prod_{i=1}^n e^{\frac{\lambda^2(b-a)^2}{8}}\right) e^{-\lambda nt}$$

其中最后一个不等式是 $Hoeffding's Lemma$。在 $\lambda \geq 0$ 上最小化，我们得到

$$\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n}(Z_i - \mathbb{E}[Z_i]) \geq t\right) \leq \min_{\lambda \geq 0} \exp\left(\frac{n\lambda^2(b-a)^2}{8} - \lambda nt\right) = \exp\left(-\frac{2nt^2}{(b-a)^2}\right).$$

另一个不等式证明类似。 □

Problem 3. 假设我们有一个二维数据集 X $= (1,1),(1,2),(2,1),(2,2),(10,10),(10,11),(11,10),(11,11)$，我们希望将其分为 k $= 2$ 类

(1) 如果初始类中心选择为 $\mathrm{c}_1 = (1,1), \mathrm{c}_2 = (10,10)$，请执行 k-means 算法，给出迭代过程和最终的类中心

(2) 如果初始类中心选择为 $\mathrm{c}_1 = (1,1), \mathrm{c}_2 = (2,2)$，请执行 k-means 算法，给出迭代过程和最终的类中心

比较两种初始类中心选择，可以感受到不同类中心选择对算法执行的影响

(3) 现在使用 k-means++ 算法，并希望将数据集分为 k $= 3$ 类，假设第一个类中心 c_1 已经被选择为 $(1,1)$，请计算每个点被选为第二个类中心 c_2 的概率

(4) 如果第二个类中心 c_2 被选择为 $(10,10)$, 请计算每个点被选为第三个类中心 c_3 的概率

解. (1) 初始类中心为 $\mathrm{c}_1 = (1,1), \mathrm{c}_2 = (10,10)$ 时：

第一次迭代：

分配点到最近的类中心：类 1:$(1,1),(1,2),(2,1),(2,2)$, 类 2: $(10,10),(10,11),(11,10),(11,11)$

更新类中心: 新 $c_1 = \left(\frac{1+1+2+2}{4}, \frac{1+2+1+2}{4}\right) = (1.5, 1.5)$, 新 $c_2 = \left(\frac{10+10+11+11}{4}, \frac{10+11+10+11}{4}\right) = (10.5, 10.5)$

第二次迭代：

分配点到最近的类中心：类 1: $(1,1),(1,2),(2,1),(2,2)$, 类 2: $(10,10),(10,11),(11,10),(11,11)$

类中心不变，算法收敛

最终类中心：

$$c_1 = (1.5, 1.5), \quad c_2 = (10.5, 10.5)$$

(2) 初始类中心为 $\mathrm{c}_1 = (1,1), \mathrm{c}_2 = (2,2)$ 时：

第一次迭代：

分配点到最近的类中心：类 1:$(1,1),(1,2),(2,1)$, 类 2: $(2,2),(10,10),(10,11),(11,10),(11,11)$

更新类中心: 新 $c_1 = \left(\frac{1+1+2}{3}, \frac{1+2+1}{3}\right) = (1.3, 1.3)$, 新 $c_2 = \left(\frac{2+10+10+11+11}{5}, \frac{2+10+11+10+11}{5}\right) = (8.8, 8.8)$

第二次迭代：

分配点到最近的类中心：类 1:$(1,1),(1,2),(2,1),(2,2)$, 类 2: $(10,10),(10,11),(11,10),(11,11)$

更新类中心: 新 $c_1 = \left(\frac{1+1+2+2}{4}, \frac{1+2+1+2}{4}\right) = (1.5, 1.5)$, 新 $c_2 = \left(\frac{10+10+11+11}{4}, \frac{10+11+10+11}{4}\right) = (10.5, 10.5)$

第三次迭代：

分配点到最近的类中心：类 1: $(1,1),(1,2),(2,1),(2,2)$, 类 2: $(10,10),(10,11),(11,10),(11,11)$

类中心不变，算法收敛

最终类中心：

$$c_1 = (1.5, 1.5), \quad c_2 = (10.5, 10.5)$$

(3) 使用 k-means++ 算法，k = 3,$\mathrm{c}_1 = (1,1)$ 首先计算每个点到 c_1 的平方距离：

$$D(x)^2 = \|x - c_1\|^2$$

然后计算概率:

$$P(x) = \frac{D(x)^2}{\sum_{x \in X} D(x)^2}$$

具体计算如下:

$$D((1,1))^2 = 0$$
$$D((1,2))^2 = 1$$
$$D((2,1))^2 = 1$$
$$D((2,2))^2 = 2$$
$$D((10,10))^2 = 162$$
$$D((10,11))^2 = 170$$
$$D((11,10))^2 = 170$$
$$D((11,11))^2 = 178$$

总距离平方和:

$$\sum D(x)^2 = 0 + 1 + 1 + 2 + 162 + 170 + 170 + 178 = 684$$

因此，每个点被选为 c_2 的概率为:

$$P((1,1)) = \frac{0}{684} = 0$$
$$P((1,2)) = \frac{1}{684} \approx 0.00146$$
$$P((2,1)) = \frac{1}{684} \approx 0.00146$$
$$P((2,2)) = \frac{2}{684} \approx 0.00292$$
$$P((10,10)) = \frac{162}{684} \approx 0.2368$$
$$P((10,11)) = \frac{170}{684} \approx 0.2485$$
$$P((11,10)) = \frac{170}{684} \approx 0.2485$$
$$P((11,11)) = \frac{178}{684} \approx 0.2602$$

(4) 如果 $c_2 = (10,10)$，计算每个点被选为 c_3 的概率首先计算每个点到最近类中心的平方距离:

$$D(x)^2 = \min(\|x - c_1\|^2, \|x - c_2\|^2)$$

然后计算概率:

$$P(x) = \frac{D(x)^2}{\sum_{x \in X} D(x)^2}$$

具体计算如下:

$$D((1,1))^2 = 0$$
$$D((1,2))^2 = 1$$
$$D((2,1))^2 = 1$$
$$D((2,2))^2 = 2$$
$$D((10,10))^2 = 0$$
$$D((10,11))^2 = 1$$
$$D((11,10))^2 = 1$$
$$D((11,11))^2 = 2$$

总距离平方和:

$$\sum D(x)^2 = 0+1+1+2+0+1+1+2 = 8$$

因此，每个点被选为 c_3 的概率为:

$$P((1,1)) = \frac{0}{8} = 0$$
$$P((1,2)) = \frac{1}{8} = 0.125$$
$$P((2,1)) = \frac{1}{8} = 0.125$$
$$P((2,2)) = \frac{2}{8} = 0.25$$
$$P((10,10)) = \frac{0}{8} = 0$$
$$P((10,11)) = \frac{1}{8} = 0.125$$
$$P((11,10)) = \frac{1}{8} = 0.125$$
$$P((11,11)) = \frac{2}{8} = 0.25$$

□

Problem 4. 回忆上课讲过的 Bicriteria approximation for k-means 算法，我们定义算法运行到第 i 步时类中心集合为 $\mathrm{S_i}$，初始化的集合为 $\mathrm{S_0} = \emptyset$。令 C_{opt} 为 k 个类的最优解，当算法运行到第 i 步时，我们将最优解导出的类 $\mathrm{A_1}, ..., \mathrm{A_k}$, 分为两种集合:

$$\mathrm{G}ood_i = \{\mathrm{A_j} | \phi_{\mathrm{A_j}}(\mathrm{S_{i-1}}) \leq 10\phi_{\mathrm{A_j}}(\mathrm{C}_{opt})\}$$

$$\mathrm{B}ad_i = \{\mathrm{A_1}, ..., \mathrm{A_k}\} \backslash \mathrm{G}ood_i$$

现在假设算法运行到第 i 步时，有 $\phi_{\mathrm{X}}(\mathrm{S_{i-1}}) \geq 10/\epsilon \phi_{\mathrm{X}}(\mathrm{C}_{opt})$, 请证明算法在该步选择的点 $\mathrm{c_i} \in \mathrm{B}ad_i$ 的概率至少为 $1-\epsilon$, 其中 $\epsilon \in (0,1)$

证明. 集合 X 对于类中心 $\mathrm{S_{i-1}}$ 的代价可以分为两部分, 集合 $\mathrm{G}ood_i$ 对于类中心的代价和集合 $\mathrm{B}ad_i$ 对于类中心的代价，即

$$\phi_X(S_{i-1}) = \sum_{A_j \in Good_i} \phi_{A_j}(S_{i-1}) + \sum_{A_j \in Bad_i} \phi_{A_j}(S_{i-1}).$$

由 Good 集合的定义知：$\phi_{A_j}(S_{i-1}) \leq 10\phi_{A_j}(C_{opt})$,
于是有:

$$\phi_X(S_{i-1}) \leq \sum_{A_j \in Good_i} 10\phi_{A_j}(C_{opt}) + \sum_{A_j \in Bad_i} \phi_{A_j}(S_{i-1}).$$

因为 $\phi_X(C_{opt}) = \sum_{A_j} \phi_{A_j}(C_{opt}) \geq \sum_{A_j \in Good_i} \phi_{A_j}(C_{opt})$
故

$$\phi_X(S_{i-1}) \leq 10\phi_X(C_{opt}) + \sum_{A_j \in Bad_i} \phi_{A_j}(S_{i-1}).$$

由于 $\phi_X(S_{i-1}) > 10/\epsilon\phi_X(C_{opt})$，因此

$$\sum_{A_j \in Bad_i} \phi_{A_j}(S_{i-1}) \geq (10/\epsilon - 10)\phi_X(C_{opt}).$$

算法运行到第 i 步时，算法在该步选择的点 $\mathrm{c_i} \in \mathrm{B}ad_i$ 的概率等于:
$\frac{\sum_{A_j \in Bad_i} \phi_{A_j}(S_{i-1})}{\phi_X(S_{i-1})}$
$= \frac{\sum_{A_j \in Bad_i} \phi_{A_j}(S_{i-1})}{\sum_{A_j \in Bad_i} \phi_{A_j}(S_{i-1}) + \sum_{A_j \in Good_i} \phi_{A_j}(S_{i-1})}$
$= \frac{1}{1 + \frac{\sum_{A_j \in Good_i} \phi_{A_j}(S_{i-1})}{\sum_{A_j \in Bad_i} \phi_{A_j}(S_{i-1})}}$
$\geq 1 - \epsilon$ □

1 Chromatic number(染色数) of Erdos-Renyi Graph

Definition 1.1. Erdos-Renyi 随机图 $G(n,p)$ 是一个 n 点图，每条边以概率 p 独立生成。A 表示所得图的染色数，定义为染色全部顶点时，使相邻顶点颜色不同所需最少的色数。定义随机变量 $X_1, \cdots, X_{\binom{n}{2}}$，如顶点对 t 是边, 则 $X_t = 1$，否则为 0. 可以证明随机变量 $Z_t = \mathbb{E}[A|X_1, ..., X_t]$ 构成鞅，称为 *edge exposure martingale*。

对应的，可以定义随机变量 $Y_1, \cdots, Y_n$，使得 Y_t 代表顶点 t 的邻点集，可知有 2^{n-1} 种取值。同样可以定义 *vertex exposure martingale* 为 $Z_1, \ldots, Z_n$，$Z_t = \mathbb{E}[A|Y_1, ..., Y_t]$.

Problem 5. 证明 vertex exposure martingale 是鞅.

证明. 首先，注意到 Z_t 在两种定义之下分别是 $X_1, \cdots, X_t$ 和 $Y_1, \cdots, Y_t$ 的函数。也就是说，随机性的来源（计算均值、方差积分使用的概率测度）是 X_i 和 Y_i。回顾条件全期望公式：如果 X, Y, Z 是在同一个概率空间的随机变量，且 $\sigma(Z) \subset \sigma(Y)$ (如 $Z = f(Y)$)，那么 $\mathbb{E}(X|Z) = \mathbb{E}(\mathbb{E}(X|Y,Z)|Z)$. 故而,

$$\begin{aligned}
&\mathbb{E}[Z_t|Z_1, \cdots, Z_{t-1}] \\
&= \int_{X_1, \cdots, X_{t-1}} \mathbb{E}[Z_t|Z_1, \cdots, Z_{t-1}, X_1, \cdots, X_{t-1}] \boldsymbol{d}\mathbb{P}(X_1, \cdots, X_{t-1}|Z_1, \cdots, Z_{t-1}) \qquad (1) \\
&= \int_{X_1, \cdots, X_{t-1}} \mathbb{E}[Z_t|X_1, \cdots, X_{t-1}] \boldsymbol{d}\mathbb{P}(X_1, \cdots, X_t|Z_1, \cdots, Z_{t-1}) \qquad (Z_i\text{是}X_j\text{的函数。})
\end{aligned}$$

式 (1) 源于全期望公式。我们接下来证明一个有普适意义的结论：对于任意的事件序列 $\{X_1, X_2, \cdots\}$ 和随机变量 A，定义 $Z_t = \mathbb{E}[A|X_1, \cdots, X_t]$，则有

$$\mathbb{E}[Z_t|X_1, \cdots, X_{t-1}] = \mathbb{E}[\mathbb{E}[A|X_1, \cdots, X_t]|X_1, \cdots, X_{t-1}] = \mathbb{E}[A|X_1, \cdots, X_{t-1}] = Z_{t-1} \tag{2}$$

上式第二个等号源于全期望公式。对于这样定义的 Z_t，称为关于 $\{X_i\}$ 的 Doob 鞅。代回前式，则有

$$\begin{aligned}\mathbb{E}[Z_t|Z_1, \cdots, Z_{t-1}] =& \int_{X_1,\cdots.X_{t-1}} Z_{t-1} \boldsymbol{d}\mathbb{P}(X_1, \cdots, X_t|Z_1, \cdots, Z_{t-1}) \\ =& Z_{t-1}\end{aligned}$$

□

Problem 6. 证明

$$\Pr(|A - \mathbb{E}[A]| \geq c\sqrt{n}) \leq 2e^{-\frac{c^2}{2}}$$

.

证明. 按照 vertex exposure martingale 的设定，$Z_0 = \mathbb{E}[A], Z_n = A$. 回顾 Azuma 不等式，立知只需证明 $|Z_i - Z_{i-1}| \leq 1$ 恒成立。（直接将对应定义和定理写清楚就能看出）

对于满足 $Y_1, \cdots, T_t$ 的任何图 G，设其最小染色方案为 $\mathcal{A}$，则在去掉点 t 的邻域信息 Y_t 后在全体样本空间对应的任一图 $\tilde{G}$, 因为 $\tilde{G}$ 的最小染色方案可以将 $G - t$ 正确染色 (给 t 采用新颜色则得到一染色方案)，所以满足 $|A(\tilde{G})| + 1 \geq |A(G)|$. 又因为 $\mathcal{A}$ 可以正确染色 $\tilde{G} - t$，所以 $|A(\tilde{G})| - 1 \leq |A(G)|$. 得证。 □

Remark 1.2. 我们使用了 vertex exposure martingale 来解决第九题，那如果使用 edge exposure martingale 会得到什么界呢？可见，为了更好地利用 Azuma 不等式，构建合适的鞅很重要。

Problem 7. 在课堂上，我们介绍了著名的 Goemans-Williamson 算法，该算法通过把最大割 (max-cut) 问题松弛为一个 SDP 问题，从而近似地求解最大割问题。对于这个松弛过程，请参照讲义 lecture3(1) 中 SDP 问题的定义（Example 1.3），写出在 Goemans-Williamson 算法中，我们需要求解的 SDP 问题的具体定义。

answer.

$$\begin{aligned}
\text{变量：}\quad & \{x_{ij} | 1 \leq i, j \leq n\} \\
\text{目标：}\quad & \mathbf{min} \sum_{i \leq j} w_{ij} x_{ij} \\
\text{约束：}\quad & \mathbb{X} = (x_{ij}) \quad \text{半正定} \\
& \mathbb{X} \quad \text{的对角线元素全为 } 1 \\
& \mathbb{X} \quad \text{是对称矩阵}
\end{aligned}$$

□

Problem 8. 考虑以下无向图 $G(V, E)$:
顶点集 $V = \{A, B, C, D, E, F\}$.

边集 $E = (A,B),(A,C),(A,E),(B,C),(D,E),(D,F),(E,F)$.
请回答以下问题:

(1) 画出图 G.

(2) 图 G 的全局最小割大小是多少? 列出 G 所有可能的全局最小割。

(3) 假设在执行 Karger 算法的第 1 步，我们随机选择并收缩了边 (E,F)。请画出收缩后的图 G'，并计算在我们执行完 Karger 算法的第 3 步以后，(2) 中的原始全局最小割被保留下来的概率。

解. (1)

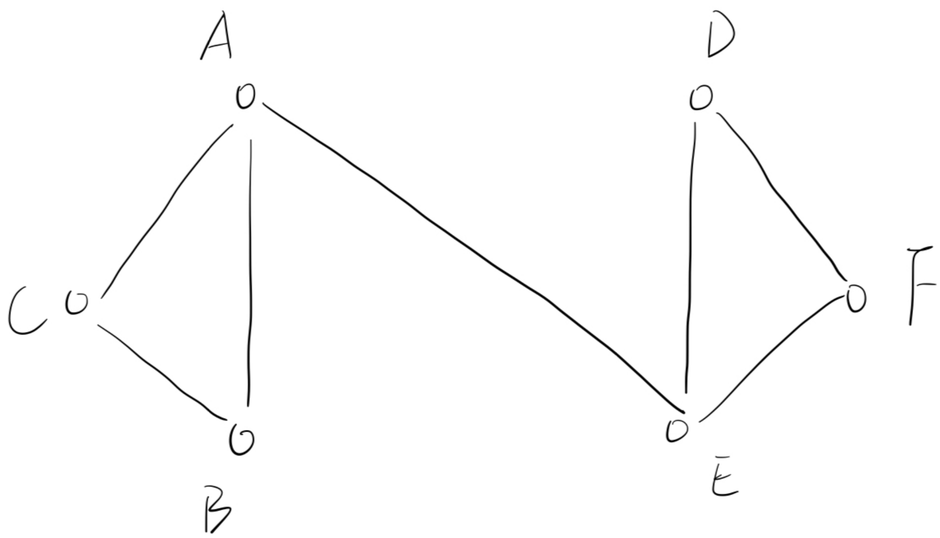

图 1: 图 G

(2) 全局最小割的大小是 1。该最小割:

$$\{\{A,B,C\},\{D,E,F\}\}$$

(3) 在 Karger 算法的第二步中，边 (A,S_{EF}) 的存活概率为 $\frac{5}{6}$.
在第三步中，若第二步收缩了边 (D,S_{EF})，则边 (A,S_{DEF}) 的存活概率为 $\frac{3}{4}$。反之，若第二步没有收缩边 (D,S_{EF})，则边 (A,S_{EF}) 的存活概率为 $\frac{4}{5}$。
综上，(2) 中的原始全局最小割被保留下来的概率为:

$$\frac{5}{6}\cdot(\frac{2}{6}\cdot\frac{3}{4}+\frac{4}{6}\cdot\frac{4}{5}) = \frac{47}{72}$$

. □

Problem ex1. (本题是开放性的思考题，不计入作业分数)K-means 算法的结果非常依赖于初始类中心的选择。虽然 K-means++ 算法通过改进初始化方式缓解了这一问题，但它仍然不能完全避免局部最优解。能否设计一种新的初始化方法，进一步减少 K-means 算法对初始类中心选择的敏感性?

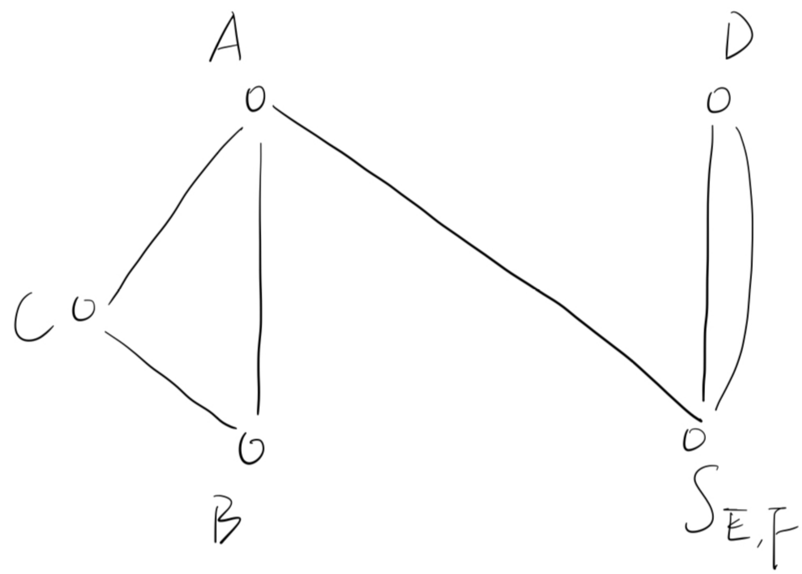

图 2: 图 G’