中国科学技术大学
计算机科学与技术学院

"大数据算法" 作业 2 (计分)

2023 年春

截止时间: 2023 年 4 月 17 日 17:59

习题 1 (15 分)
令 $\sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^\top$ 为矩阵 A 的奇异值分解 (SVD), 其中 $A \in \mathbb{R}^{n\times d}$。证明 $\|\boldsymbol{u}_1^\top A\| = \sigma_1$ 和 $\|\boldsymbol{u}_1^\top A\| = \max_{\|\boldsymbol{u}\|=1} \|\boldsymbol{u}^\top A\|$。
注: 对于向量 $\boldsymbol{x} \in \mathbb{R}^d$, $\|\boldsymbol{x}\| = \sqrt{\sum_{i=1}^{d} x_i^2}$。

习题 2 (25 分)
令 A 为一个 $n \times d$ 的矩阵, 其奇异值分解 (SVD) 为 $A = \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^\top$。令 $\boldsymbol{x} \in \mathbb{R}^d$ 为一个向量, 满足 $\|\boldsymbol{x}\|_2 = 1$ 并且对于某个 $\delta > 0$, 有 $|\boldsymbol{x}^\top \boldsymbol{v}_1| \geq \delta$。假设 $\sigma_2 < \frac{1}{2}\sigma_1$。令 $\boldsymbol{w}$ 为经过 $k = \log(1/\varepsilon\delta)$ 次幂法迭代后得到的向量, 即

$$\boldsymbol{w} = \frac{(A^\top A)^k \boldsymbol{x}}{\|(A^\top A)^k \boldsymbol{x}\|_2}.$$

证明 $\boldsymbol{w}$ 在第一个奇异向量 $\boldsymbol{v}_1$ 上投影的长度至少为 $1-\varepsilon$, 即 $|\boldsymbol{w}^\top \boldsymbol{v}_1| \geq 1-\varepsilon$。

习题 3 (20 分)
假设 $k < d$。假设 $U \in \mathbb{R}^{d\times k}$ 是一个随机矩阵, 其第 (i,j) 个元素记作 u_{ij}。这里 $\{u_{ij}\}$ 是独立的随机变量, 满足:

$$u_{ij} = \begin{cases} 1 & \text{以 } \frac{1}{2} \text{ 的概率}, \\ -1 & \text{以 } \frac{1}{2} \text{ 的概率}. \end{cases}$$

我们使用矩阵 U 作为一个随机投影矩阵。也就是说, 对于一个行向量 $\boldsymbol{a} \in \mathbb{R}^d$, 我们把它映射到

$$f(\boldsymbol{a}) = \frac{1}{\sqrt{k}} \boldsymbol{a} U.$$

对于 $1 \leq j \leq k$ 中的每个 j, 定义 $b_j = [f(\boldsymbol{a})]_j$, 即 b_j 是 $f(\boldsymbol{a})$ 的第 j 个元素。

- 计算 $\mathrm{E}[b_j]$。
- 计算 $\mathrm{E}[b_j^2]$。
- 计算 $\mathrm{E}[\|f(\boldsymbol{a})\|^2]$。

习题 4 (20 分)
在本课程中, 我们学习了一个解决 (c,r)-ANN 问题的算法 (记作 $\mathcal{A}$), 其成功概率至少为 0.6。也就是说, 针对一个查询点 x, 如果数据集合 $\mathcal{P}$ 中存在一个点 a^* 满足 $d(x,a^*) \leq r$, 那么算法 $\mathcal{A}$ 将会以至少 0.6 的概率输出某个点 $a \in \mathcal{P}$, 满足 $d(x,a) \leq c \cdot r$。

假设 $\delta \in (0,1)$。使用上述算法 $\mathcal{A}$ 作为一个子程序, 给出一个成功概率至少为 $1-\delta$ 的新算法 $\mathcal{B}$。也就是说, 对于上述查询点 x, 算法 $\mathcal{B}$ 将会以至少 $1-\delta$ 的概率输出某个点 $a \in \mathcal{P}$, 满足 $d(x,a) \le c \cdot r$。你的算法应该使用尽可能少的查询时间。假设 $\mathcal{A}$ 的查询时间是 $T_{\mathcal{A}}$, 说明你的算法的正确性和查询时间。

习题 5 (20 分)
假设 $\alpha \in (0,1]$。假如我们将 (基本的) Morris 算法修改如下:

(a) 初始化 $X \leftarrow 0$.

(b) 对于每次更新, 以 $\frac{1}{(1+\alpha)^X}$ 的概率使 X 增加 1.

(c) 对于查询, 输出 $\tilde{n} = \frac{(1+\alpha)^X - 1}{\alpha}$.

记 X_n 为上述算法中 n 次更新以后的 X。令 $\tilde{n} = \frac{(1+\alpha)^{X_n} - 1}{\alpha}$。

- 计算 $\mathrm{E}[\tilde{n}]$ 并给出 $\mathrm{Var}[\tilde{n}]$ 的一个上界。
- 假设 $\varepsilon, \delta \in (0,1)$。基于上述算法（你可以任意指定 α 的具体值）, 给出一个新算法, 使得新算法以至少 $1-\delta$ 的概率输出一个估计值 $\tilde{n}$, 满足 $|\tilde{n} - n| \le \varepsilon n$。说明你的算法的正确性与 (**最坏**) 空间复杂度（即算法使用的比特数）。你的算法只需要满足以至少 $1-\delta'$ 的概率, 其**最坏**空间复杂度为关于 $\frac{1}{\delta}$, $\frac{1}{\delta'}$, $\frac{1}{\varepsilon}$ 和 $\log\log n$ 的多项式 (即 $\mathrm{poly}(\frac{1}{\delta}, \frac{1}{\delta'}, \frac{1}{\varepsilon}, \log\log n)$)。

习题 6 (附加题 10 分)
在本课程中, 我们学习了一个基于降维来解决 (c,r)-ANN 问题的算法 (见 Lecture 7 讲义)。
假设 $0 < p \le \frac{1}{2}$。证明对于任意 $\boldsymbol{x}, \boldsymbol{y} \in \{0,1\}^d$, 都有

$$\Pr[(U\boldsymbol{x})_i \neq (U\boldsymbol{y})_i] = \frac{1}{2}\left(1 - (1-2p)^{\mathrm{Ham}(\boldsymbol{x},\boldsymbol{y})}\right),$$

其中 U 是一个 $k \times d$ 的随机矩阵, 其中的每个元素都是独立同分布的, 满足:

$$u_{ij} = \begin{cases} 1 & \text{以 } p \text{ 的概率}, \\ 0 & \text{以 } 1-p \text{ 的概率}, \end{cases}$$

并且所有的运算都在有限域 GF(2) 中进行 (即加法和乘法的结果都要模 2)。

提示: 你可以考虑使用如下事实: 假设 $\boldsymbol{w} \in \{0,1\}^d$ 为一个随机向量, 其中的每个元素 w_i 都是独立同分布的, 并且对于每个 $i \le d$, 都有 $\Pr[w_i = 1] = \Pr[w_i = 0] = \frac{1}{2}$。那么如果 $\boldsymbol{x} \neq \boldsymbol{y}$, 有 $\Pr[\boldsymbol{w}^\top \boldsymbol{x} \neq \boldsymbol{w}^\top \boldsymbol{y}] = \frac{1}{2}$。
