大数据算法 HW1

PB18111697 王章瀚

2021 年 5 月 17 日

1.

假设 $\delta,\theta\in(0,1)$, 给定集合 U 以及其一个子集 V, $|V|/|U|=a$

(1).

如果我们从 U 均匀采样本 S, 并确保 $S\cap V\neq\varnothing$ 的概率大于 $1-\delta$, $|S|$ 至少要多大?
考虑有放回采样:

$$\begin{aligned}P(S\cap V\neq\varnothing)&=1-P(S\cap V=\varnothing)\\&=1-\frac{|U|-|V|}{|U|}^{|S|}\\&=1-(1-a)^{|S|}\\&>1-\delta\end{aligned}$$

可知, $|S|$ 应当满足 $|S|>\log_{(1-a)}\delta$

(2).

如果我们要确保 $|S\cap V|\in(1\pm\theta)a|S|$ 的概率大于 $1-\delta$, $|S|$ 至少要多大?
显然, 随机抽样在 V 里的概率为 a, 可以考虑 $|S|$ 个独立同分布于 $B(1,a)$ 的随机变量 $X_1,\cdots,X_{|S|}$
即对 $i=1,\cdots,|S|$ 有

$$P(X_i=k)=\begin{cases}a & ,k=1\\1-a & ,k=0\end{cases}$$

那么 $P(|S\cap V|=k)=P(\sum_{i=1}^{|S|}X_i=k)$, 且 $E[|S\cap V|]=a|S|$
由 Chernoff Bounds:

$$\begin{aligned}P(|S\cap V|\notin(1\pm\theta)a|S|)&=P(|S\cap V|\notin(1\pm\theta)E[|S\cap V|])\\&=P\left(\left|\sum_{i=1}^{|S|}X_i\right|\notin a|S|\pm a\theta|S|\right)\\&\leq 2\exp(-2|S|(a\theta)^2)\end{aligned}$$

即 $P(|S\cap V|\in(1\pm\theta)a|S|)>1-2\exp(-2|S|(a\theta)^2)$
故要想$P(|S\cap V|\in(1\pm\theta)a|S|)>1-\delta$, 应确保

$$1-2\exp(-2|S|(a\theta)^2)>1-\delta$$

即

$$|S| > \frac{\ln \frac{2}{\delta}}{2(a\theta)^2}$$

2.

给定高维空间 R^d 的一个点集 P, $|P| = n$, 如果我们进行 k-means 聚类,(1). 能否通过 J-L 变换来降低复杂度? (2).能否通过 PCA 来降低复杂度? 请分别给出详细的理论分析. (比如, 如果我们有一个 λ 倍近似比的 k-means 聚类算法 A, 其复杂度为 T(n,d), 那么我们用 A 对降维之后的数据运算, 并将得到的结果返回原有空间, 得到近似比是多少? 整个过程复杂度是多少?) 不妨依据题目意思, 假设我们有一个 λ 倍近似比的 k-means 算法, 复杂度为 $T(n,d)$.

(1).

能否通过 J-L 变换来降低复杂度?

JL 变换的时间复杂度是 $O(nd\frac{\log n}{\epsilon^2})$, 能够保证的是降维到 $k = O(\frac{\log n}{\epsilon^2})$ 后, 仍有

$$(1-\epsilon)\|u-v\|^2 \leq \|f(u)-f(v)\|^2 \leq (1+\epsilon)\|u-v\|^2$$

即其降维后距离变化不会超过 ϵ 倍.

k-means 算法的复杂度是 $T(n,d)$, 近似比是 λ, 目标是

$$\text{minimize} \sum_{x\in X} \|\phi_S(x), x\|^2$$

因此降维之后, 相当于是

$$\text{minimize} \sum_{x\in X} \|\phi_S(f(x)), f(x)\|^2$$

且其中有

$$\begin{aligned}
C_{recover} &= \sum_{l=1}^{k} \sum_{x\in C_i'} \|x - C_l'\|^2 \\
&= \sum_{l=1}^{k} \sum_{x_i,x_j\in C_i'} \frac{\|x_i - x_j\|^2}{2n_l} \\
&\leq \frac{1}{1-\epsilon} \sum_{l=1}^{k} \sum_{x_i,x_j\in C_i'} \frac{\|\phi(x_i) - \phi(x_j)\|^2}{2n_l} \\
&\leq \frac{\lambda}{1-\epsilon} \sum_{l=1}^{k} \sum_{x_i,x_j\in C_i'} \frac{\|\phi(x_i) - \phi(x_j)\|^2}{2n_l} \\
&\leq \lambda\frac{1+\epsilon}{1-\epsilon} \sum_{l=1}^{k} \sum_{x_i,x_j\in C_i'} \frac{\|\phi(x_i) - \phi(x_j)\|^2}{2n_l} \\
&= \lambda\frac{1+\epsilon}{1-\epsilon} OPT
\end{aligned}$$

由此可知, 近似比是 $\lambda(1+\epsilon)$.

相应时间复杂度包含如下几步:

- JL 变换需要 $O(nd\frac{\log n}{\epsilon^2})$
- k-means 需要 $T(n,k)=T(n,\frac{\log n}{\epsilon^2})$
- JL 变换回去, 不考虑求逆矩阵的时间(JL正逆矩阵一般可以提前准备好, 与数据无关), 但逆变换依然需要一次矩阵乘法, 即 $O(nd\frac{\log n}{\epsilon^2})$

综上总共是 $O(2nd\frac{\log n}{\epsilon^2})+T(n,\frac{\log n}{\epsilon^2})=O(nd\frac{\log n}{\epsilon^2})+T(n,\frac{\log n}{\epsilon^2})$
因此, 只要能够保证 $nd\frac{\log n}{\epsilon^2}=\omega(T(n,d))$, 就能够保证在 $\lambda(1+\epsilon)$ 近似比的条件下降低复杂度.

(2).

能否通过 PCA 来降低复杂度?
对于 PCA 算法, 其复杂度包括 计算协方差矩阵 $O(d^2n)$ 和 特征值分解 $O(d^3)$ 及正逆变换 $O(ndk)$, 故总共需要 $O(d^2n+d^3+ndk)$. 这里复杂度已经非常高, 并不太适合作为 k-means 算法的降维处理操作.
另一方面, PCA 算法没有一个近似比保证, 从这个角度来说, 它并不合适.

设 P 已中心化, C 为最优解对应的 center 集合, 其张成 $k-1$ 维子空间 H. 考虑经 $W\in\mathbb{R}^{(k-1)\times d}$ 将数据集 P 投影到一个 $k-1$ 维超平面 H' 上, 则

$$\sum_{i=1}^{n}\|p_i-\pi_{H'}(p_i)\|^2\le\sum_{i=1}^{n}\|p_i-\pi_H(p_i)\|^2\le\sum_{i=1}^{n}\|p_i-C(p_i)\|^2$$

在 P' $k-1$ 维空间上运行 k-means 算法, 得到 $C'(P')$ 为其center. $\pi_{H'}(P_i)=W^TWP$ 用三角不等式:

$$\begin{aligned}C_{recover}&=\sum_{i=1}^{n}\|p_i-W^TC'(P_i)\|^2\\&\le\sum_{i=1}^{n}\|p_i-\pi_{H'}(p_i)\|^2+\|\pi_{H'}(p_i)-W^TC'(p_i)\|^2\\&=\sum_{i=1}^{n}\|p_i-\pi_{H'}(p_i)\|^2+\|W^T(Wp_i-C'(p_i))\|^2\end{aligned}$$

先估计这个:

$$\begin{aligned}\sum_{i=1}^{n}\|Wp_i-C'(p_i)\|^2&\le\lambda\sum_{i=1}^{n}\|Wp_i-WC(p_i)\|^2\\&=\lambda\sum_{i=1}^{n}\|W(p_i-WC(p_i))\|^2\\&\le\lambda\sum_{i=1}^{n}\|p_i-c(p_i)\|^2\end{aligned}$$

用这个可以放缩第二项, 最前面的不等式能放缩第一项.

3.

请构造一个反例, 证明 Gonzalez 算法不能代替 k-means++ 算法用于 k-means 聚类的初始点选择.

如果用 Gonzalez 算法, 考虑一个二分类问题, 里面有四个点, 分别为狭长的矩形的四个顶点, 如下图:

由于随机性, 第一步可能选取的是 A, 然后类中心变成了矩形中心. 而后, 如果第二步选取的是 D, 那么就会导致分类结果是 {A,B}, {C,D}, 这明显和最优结果 {A,D},{B,C} 相差很多, 且无法被 k-means 迭代修正.

而如果使用 k-means++ 的办法, 则依然有一定概率(只是比较小)能够在第二步选择 B 或 C, 从而使得整个聚类达到最优.

应当用 (A)-[distance: x]-(B)-[distance:y]-(C) 的形式, 其中 $x \ll y$.

4.

请计算平面上以下 range 的 VC Dimension:

(1). 矩形

考虑圆上 7 个点, (构成正七边形)

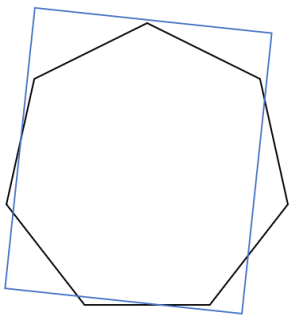

图 1: 4.(1). 3 个点的比较不易想到的方法示意图

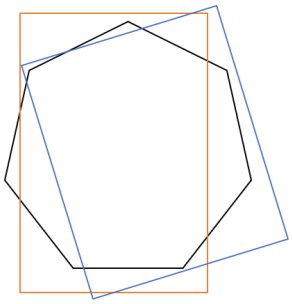

图 2: 4.(1). 4 个点的比较不易想到的方法示意图

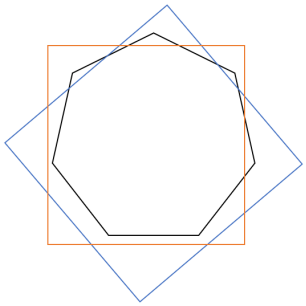

图 3: 4.(1). 5 个点的比较不易想到的方法示意图

则显然其任意子集必然可以被某个矩形划分(子集大小为 1, 2, 6, 7 显然, 为 3, 4, 5 的不显然情况则如上所示(结合对称性)), 因此矩形的 VC-dimension 至少是 5.

而如果考虑正八边形, 由于有些边夹角已经小于90度, 直觉上难以画出可行解, 但我无法证明其不行.

(2). 半圆形

同样, 考虑前面矩形的那个五点情况, 一样可以得出半圆形在平面上的 VC-dimension 至少是 5. (即考虑子集大小为 1, 4, 5 均比较显然, 大小为 2 时, 比较不显然的情况在图中用绿线标出, 大小为 3 时, 所有(对称地)情况都用蓝色标出了)

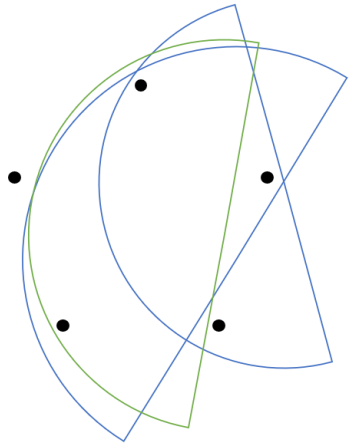

图 4: 4.(2). 示意图

如果能证明大于 5 不行, 那么半圆形的 VC 维就是 5.

(3). 凸多边形

显然, 凸多边形的 VC dimension 是无穷大的. 我们只需要考虑圆上均匀分割的 n 个点, 那么其中任意几个点构成的点集都可以用一个比由这些点构成的凸多边形包围. 且这里的 n 可以无限大.
因此, 凸多边形的 VC dimension 是 ∞ 的.