New Directions in Nearest Neighbor Searching with Applications to Lattice Sieving

背景介绍

- 最近邻搜索(NNS)问题:给定N个n维向量集合,通过预处理,使得寻找某个向量在集合中的最近邻向量时间在 $O(N^{\rho})(\rho < 1)$
- 近似NNS: 距离给定向量v最近的向量距离 r_1 (角度 θ_1), 其余向量大于距离 r_2 (角度 θ_2)
- 高维球面: $\mathcal{S}^{n-1}:=\{oldsymbol{x}\in\mathbb{R}^n,||oldsymbol{x}||=1\}$

石晋 2025.4.30 2214

局部敏感哈希(LSH)

- 局部敏感哈希(LSH): hash函数族升。
 - 其中的hash函数满足:相近的向量大概率hash值相同
- 预处理:取t组k个hash函数 $h_{i,j}\in\mathcal{H}$,对每个向量 $m{w}$ 计算t组 $h_i(m{w})=(h_{i,1}(m{w}),\cdots,h_{i,k}(m{w}))$,生成t个hash table
- 查找:对于向量 $oldsymbol{v}$,将 $h_i(oldsymbol{v})$ 在hash table中对应的所有 $oldsymbol{w}$ 纳入 $oldsymbol{v}$ 的近邻向量候选
- 定义hash函数的碰撞概率:

$$p(heta) := \Pr_{h \sim \mathcal{H}}[h(oldsymbol{v}) = h(oldsymbol{w}) | oldsymbol{v}, oldsymbol{w} \in \mathcal{S}^{n-1}, \langle oldsymbol{v}, oldsymbol{w}
angle = \cos heta]$$

• 若向量角度大于 θ_2 , 希望hash值相等的概率低: 调整k

$$egin{aligned} P_2 &= \Pr[h_i(oldsymbol{v}) = h_i(oldsymbol{w}) | oldsymbol{v}, oldsymbol{w} \in \mathcal{S}^{n-1}, \langle oldsymbol{v}, oldsymbol{w}
angle \leq \cos heta_2] \ &= \prod_{j=1}^k \Pr[h_{i,j}(oldsymbol{v}) = h_{i,j}(oldsymbol{w}) | oldsymbol{v}, oldsymbol{w} \in \mathcal{S}^{n-1}, \langle oldsymbol{v}, oldsymbol{w}
angle \leq \cos heta_2] \leq p(heta_2)^k \leq rac{1}{N} \end{aligned}$$

$$\circ$$
 $k \geq rac{\log N}{-\log p(heta_2)}$

• 若向量角度小于 θ_1 ,希望至少要有一组相同的hash值:调整t

$$P_1=\Pr[h_i(oldsymbol{v})=h_i(oldsymbol{w})|oldsymbol{v},oldsymbol{w}\in\mathcal{S}^{n-1},\langleoldsymbol{v},oldsymbol{w}
angle\geq\cos heta_1]\geq p(heta_1)^k\geq N^{-rac{\log p(heta_1)}{\log p(heta_2)}}$$
o $t\cdot P_1\geq 1$,即 $t\geq rac{1}{P_1}=N^{rac{\log p(heta_1)}{\log p(heta_2)}}$

• 时间复杂度: 计算 $t\cdot k$ 次hash, $\widetilde{O}(N^{rac{\log p(heta_1)}{\log p(heta_2)}})$

局部敏感过滤(LSF)

- 局部敏感过滤(LSF): 过滤函数族F
- 预处理:取t组k个过滤函数 $f_{i,j}\in\mathcal{F}$,对每个向量 $m{w}$ 进行t次组合过滤 $f_i(m{w})=(f_{i,1}(m{w}),\cdots,f_{i,k}(m{w}))$,生成t个过滤后的向量集合 (L_1,\cdots,L_t)
- 查找:对于向量v,将能通过的 $f_i(v)$ 对应的过滤集合中对应的所有w纳入v的近邻向量候选
- 定义过滤函数的碰撞概率:

$$p(heta) := \Pr_{f \sim \mathcal{F}}[oldsymbol{v}, oldsymbol{w} \in L_f | oldsymbol{v}, oldsymbol{w} \in \mathcal{S}^{n-1}, \langle oldsymbol{v}, oldsymbol{w}
angle = \cos heta]$$

- $m{\circ}$ p(0): 随机向量通过过滤函数f的概率 $\Pr[m{v} \in L_f | m{v} \in \mathcal{S}^{n-1}]$
- $\cdot t \cdot p(0)^k$: 随机向量能通过组合过滤的数量(向量能进到几个组)

• 若角度大于 θ_2 ,希望最终被分在一个组中的概率低:调整k

$$egin{aligned} P_2 &= \Pr[oldsymbol{w} \in L_i | oldsymbol{v}, oldsymbol{w} \in \mathcal{S}^{n-1}, oldsymbol{v} \in L_i, \langle oldsymbol{v}, oldsymbol{w}
angle \leq \cos heta_2] \ &= \prod_{j=1}^k rac{\Pr[oldsymbol{v}, oldsymbol{w} \in L_{i,j} | oldsymbol{v}, oldsymbol{w} \in \mathcal{S}^{n-1}, \langle oldsymbol{v}, oldsymbol{w}
angle \leq \cos heta_2]}{\Pr[oldsymbol{v} \in L_{i,j} | oldsymbol{v} \in \mathcal{S}^{n-1}]} \leq \left(rac{p(heta_2)}{p(0)}
ight)^k \leq rac{1}{N} \end{aligned}$$

$$\circ$$
 $k \geq rac{\log N}{\log p(0) - \log p(heta_2)}$

• 若角度小于 θ_1 , 希望最终被分在一个组中: 调整t

$$P_1 = \Pr[oldsymbol{w} \in L_i | oldsymbol{v}, oldsymbol{w} \in \mathcal{S}^{n-1}, oldsymbol{v} \in L_i, \langle oldsymbol{v}, oldsymbol{w}
angle \geq \cos heta_1] \geq \left(rac{p(heta_1)}{p(0)}
ight)^k \geq N^{-rac{\log p(0) - \log p(heta_1)}{\log p(0) - \log p(heta_2)}}$$

$$oldsymbol{\circ} t \cdot p(0)^k \cdot P_1 \geq 1$$
,即 $t \geq rac{1}{p(0)^k \cdot P_1} = N^{rac{-\log p(heta_1)}{\log p(0) - \log p(heta_2)}}$

• 时间复杂度: 计算 $t\cdot p(0)^k$ 次过滤函数, $\widetilde{O}(N^{rac{\log p(0)-\log p(heta_1)}{\log p(0)-\log p(heta_2)}})$

LSH和LSF实例

• Spherical LSH: 单个hash函数通过采样 $U=2^{\Theta(\sqrt{n})}$ 个单位向量 $m{s_1},m{s_2},\cdots,m{s_U}$,取 $lpha=n^{-\frac{1}{4}}$,构造

$$H_{oldsymbol{s}_i} := \mathcal{C}_{oldsymbol{v},lpha}ackslash egin{align*} \int\limits_{j=1}^{i-1} H_{oldsymbol{s}_j} & p(heta) = \exp\left[-rac{\sqrt{n}}{2} an^2\left(rac{ heta}{2}
ight)(1+o(1))
ight] \ & \log p(heta_i) = an^2(heta_i/2) \end{aligned}$$

$$ho_{LSH} = rac{\log p(heta_1)}{\log p(heta_2)} = rac{ an^2(heta_1/2)}{ an^2(heta_2/2)}(1+o(1))$$

• Spherical LSF: 单个过滤函数通过一个随机单位向量s, 和一个角度 α , 构造

$$egin{aligned} F_{m{s}} := \mathcal{C}_{m{v},lpha} & p(heta) = \exp\left[rac{n}{2} \ln\left(1 - rac{2lpha^2}{1 + \cos heta}
ight) (1 + o(1))
ight] \
ho_{LSF} = rac{\log(1 - lpha^2) - \log\left(1 - rac{2lpha^2}{1 + \cos heta_1}
ight)}{\log(1 - lpha^2) - \log\left(1 - rac{2lpha^2}{1 + \cos heta_2}
ight)} (1 + o(1)) \end{aligned}$$

α 的取值

$$ho_{LSF} \stackrel{lpha=0}{\sim} rac{\log p(heta_1)}{\log p(heta_2)} = rac{ an^2(heta_1/2)}{ an^2(heta_2/2)} =
ho_{LSH}$$

• 当 $\alpha = 0$ 时,从理论时间复杂度来看,Spherical LSF和Spherical LSH相同

$$k=rac{\log N}{\log p(0)-\log p(heta_2)}\geq 1\Rightarrow lpha\leq lpha_0:=\sqrt{1+rac{N^{2/n}(\cos heta_2-1)}{2N^{2/n}-\cos heta_2-1}}$$

• 为了使 ρ 小, α 尽可能大,当k=1时, α 取到最大值

BDGL构造

随机乘积编码(Random product codes): 使用直积进行Filter的构造

$$C = Q \cdot (C_1 imes C_2 imes \cdots imes C_m)$$

- Q代表 \mathbb{R}^n 上的均匀随机旋转, $n=m\cdot b$, $C_i\subset\sqrt{rac{1}{m}}\mathcal{S}^{n-1}$, $C_i=\{m{c_{i,1}},m{c_{i,2}},\cdots,m{c_{i,B}}\}$
- $m = O(\log n)$ 时理论最优
- 可以使用 $B \cdot m$ 个长度为b的随机向量表示 $M = B^m$ 个长度为n的随机向量
- 等价组合过滤器数量: $t = B^m$

解码:对于一个向量 $m{w}\in\mathcal{S}^{n-1}$,寻找所有 $m{s}=(m{c_{1,j_1}}|m{c_{2,j_2}}|\cdots|m{c_{m,j_m}})$,满足 $\langle m{w},Qm{s}\rangle\geq lpha$

算法: 点乘可以简单的拆分

- 计算 $oldsymbol{v} = Q^{-1}oldsymbol{w}$,并拆分 $oldsymbol{v} = (oldsymbol{v_1}|oldsymbol{v_2}|\cdots|oldsymbol{v_m})$
- 对所有 v_i , 计算 $\langle v_i, c_{i,j} \rangle$, 并排序
- 深度有限搜索, 通过改变阈值剪枝
- 一次查询时间复杂度:

$$\mathcal{T}_{LD}(t,lpha) = O(nB + mB\log B + mt\mathcal{C}_n(lpha))$$

石晋

符号介绍

- 高维球面: $\mathcal{S}^{n-1} := \{oldsymbol{x} \in \mathbb{R}^n, ||oldsymbol{x}|| = 1\}$
- 高维圆锥: $\mathcal{H}_{m{v},lpha}:=\{m{x}\in\mathbb{R}^n, \langle m{v},m{x}
 angle\geq lpha\}$
- 高维球冠: $\mathcal{C}_{oldsymbol{v},lpha}:=\mathcal{S}^{n-1}\cap\mathcal{H}_{oldsymbol{v},lpha}$

$${\mathcal C}_n(lpha) = rac{\mu({\mathcal C}_{oldsymbol{v},lpha})}{\mu({\mathcal S}^{n-1})} = ext{poly}(n) \cdot ig(1-lpha^2ig)^{rac{n}{2}}$$

• 高维球面楔: $\mathcal{W}_{m{v},lpha,m{w},eta}:=\mathcal{S}^{n-1}\cap\mathcal{H}_{m{v},lpha}\cap\mathcal{H}_{m{w},eta}$, 令 $\cos heta=\langlem{v},m{w}
angle$

$$\mathcal{W}_n(lpha,eta, heta) = rac{\mu(\mathcal{W}_{oldsymbol{v},lpha,oldsymbol{w},eta)}}{\mu(\mathcal{S}^{n-1})} = ext{poly}(n) \cdot \left(1 - rac{lpha^2 + eta^2 - 2lphaeta\cos heta}{\sin^2 heta}
ight)^{rac{n}{2}}$$

• GeoGebra 🛨

应用到筛法中

- 目标:找到所有角度小于π/3的向量
- 询问α: α越大, 角度越小, 查询代价越小
- 插入(预处理)β: β越大,角度越小,需要构建更多的过滤器以保证查询成功率
- 期望过滤器数量: $t = \tilde{\mathcal{O}}(1/\mathcal{W}_n(\alpha, \beta, \pi/3))$
- 询问期望能通过过滤器的数量: $t \cdot \mathcal{C}_n(\alpha)$
- 每个过滤器中的期望向量数量: $N \cdot \mathcal{C}_n(\beta)$

向量数量: $N=\left(rac{4}{3}
ight)^{rac{n}{2}+o(n)}$

查询时间复杂度:

$$\mathcal{T}_1 = ilde{\mathcal{O}}\left(rac{N\cdot\mathcal{C}_n(lpha)\cdot(1+N\cdot\mathcal{C}_n(eta))}{\mathcal{W}_n(lpha,eta,\pi/3)}
ight) = ilde{\mathcal{O}}\left(\left(rac{4(1-lpha^2)}{3-4(lpha^2+eta^2-lphaeta)}
ight)^{rac{n}{2}}\left[1+\left(rac{4(1-eta^2)}{3}
ight)^{rac{n}{2}}
ight]
ight)$$

预处理时间复杂度:

$$\mathcal{T}_2 = ilde{\mathcal{O}}\left(rac{N\cdot\mathcal{C}_n(eta)}{\mathcal{W}_n(lpha,eta,\pi/3)}
ight) = ilde{\mathcal{O}}\left(\left(rac{4(1-eta^2)}{3-4(lpha^2+eta^2-lphaeta)}
ight)^rac{n}{2}
ight)$$

空间复杂度:

$$\mathcal{S} = ilde{\mathcal{O}}\left(N + rac{N\cdot \mathcal{C}_n(eta)}{\mathcal{W}_n(lpha,eta,\pi/3)}
ight) = ilde{\mathcal{O}}\left(\left(rac{4}{3}
ight)^{rac{n}{2}} + \left(rac{4(1-eta^2)}{3-4(lpha^2+eta^2-lphaeta)}
ight)^{rac{n}{2}}
ight)$$

石晋

$$egin{align} ullet (lpha,eta) &= (rac{1}{2},rac{1}{2}) \ &\mathcal{T} &= (3/2)^{n/2+o(n)} pprox 2^{0.292n+o(n)} \ &\mathcal{S} &= (3/2)^{n/2+o(n)} pprox 2^{0.292n+o(n)} \end{aligned}$$

$$egin{align} ullet (lpha,eta) &= (rac{1}{4},rac{1}{2}) \ &\mathcal{T} &= (5/3)^{n/2+o(n)} pprox 2^{0.368n+o(n)} \ &\mathcal{S} &= (4/3)^{n/2+o(n)} pprox 2^{0.208n+o(n)} \ \end{aligned}$$

左下角的点只能使用NV筛

