随机算法选讲

张恒捷 (zhanghj15@mails.tsinghua.edu.cn)

清华大学 交叉信息研究院

January 29, 2016

最小圆覆盖

给出欧几里得平面上n个点，求覆盖住所有点的最小圆。

最小圆覆盖

```
Get(P1,P2)          //Get是求以这些点为边界确定的最小圆
for i = 3 to N
if not Inside(Pi)   //Inside是判断点是否在当前圆内
then
    Get(P1,Pi)
    for j = 2 to i-1
    if not Inside(Pj)
    then
        Get(Pi,Pj)
        for k = 1 to j
        if not Inside(a[k])
            Get(Pi,Pj,Pk)
        end if
    end if
end if
```

最小圆覆盖

算法流程：

- 假设该算法接收两个点集S,Q，输出一个最小圆包含S中所有点且Q中的点都在圆的边界上
- 以随机的顺序处理S中的点，且维护P为已处理的点集，维护包含P中所有点的最小圆
- 若下一个点r不在圆中，以P与Q+r为参数递归本算法，否则r加入P集合
- 令S为全集，Q为空集调用该算法可得到原问题的解

最小圆覆盖

- 正确性：只需注意到在第三步中，r必定在圆边界上
- 复杂度：由于随机性，在处理第i个点时调用递归的概率为$O(1/i)$，对 $|Q|$ 归纳证明期望复杂度 $O(n)$
- 可推广到常数维空间，期望复杂度仍然为O(n)

全局最小割

一张连通,无向,有重边的图$G=(V,E)$,
设$|V|=n,|E|=m$。将V划分成两个非空子集 A,B 使得端点分别在 A,B 上的边数尽可能少。

可以调用 n 次 s-t min-cut 算法解决此问题，而且为确定性算法。假设最快的流算法为 $O(nm)$,则此方法需要 $\Omega(n^2m)$时间。

有随机算法能做到 $O(n^2 log^{O(1)} n)$ 时间，在稠密图中胜过最快的 max-flow 算法。

全局最小割

算法 Contract:

- 1.等概率随机一条边，将两端的点收缩起来，得到新图G'(可带重边)
- 2.更新边集，去除自环
- 3.重复这个过程直到图中只剩两个点

第2步可以做到$O(n)$

时间复杂度：$O(n^2)$

正确率分析

一个显然的事实：一个割被该算法输出当且仅当割中的边没有被收缩

假设最小割 K 的值为 k ，由于每个点的度数大于等于k，故 $|E| \geq nk/2$

第一次没被收缩的概率：$\geq \frac{n-2}{n}$

第二次没被收缩的概率：$\geq \frac{n-3}{n-1}$

第 i 次没被收缩的概率：$\geq \frac{n-i-1}{n-i+1}$

正确率分析

$$Pr[K\text{被输出}] \geq \prod_{i=1}^{n-2} \frac{n-i-1}{n-i+1} = \prod_{i=3}^{n} \frac{i-2}{i} = 1/\binom{n}{2}$$

重复该算法 $O(n^2 log\ n)$ 次正确率会很高。
我们获得了一个 $O(n^4 log\ n)$ 拥有很高正确率的随机算法。

改进

正确率为瓶颈。
注意到收缩到 t 个点的时候正确率是 $O(\frac{t^2}{n^2})$。
启发我们在剩余点较少的时候用正确率高的算法。
然而最快的确定性算法是 $O(n^3)$ 的。
练习：检验以 $1/2$ 为正确率时，最快能做到 $\Omega(n^{8/3})$。

改进

放弃确定性算法，仍使用随机算法。

算法 FastCut:

1. $n = |V|$

2. 若 $n \le 6$, 暴力否则

   2.1 令 $t = \lceil 1 + n/\sqrt{2} \rceil$.

   2.2 独立两次使用收缩边的算法 **Contract** ，获得两个图$H1, H2$ ，各自有 t 个点。

   2.3 递归计算 $H1, H2$ 的最小割

   2.4 返回较小者

分析

时间复杂度：
设 n 为图的点数，算法 **Contract** 能用 $O(n^2)$ 获得 $H1, H2$。
$T(n) = 2T\left(\left\lceil 1 + \frac{n}{\sqrt{2}} \right\rceil\right) + O(n^2)$
$T(n) = O(n^2 log\ n)$

空间复杂度：
在深度为 d 的时候有 $O(n/2^{d/2})$ 个点，每层当然只需要处理一个图，总空间不超过
$O(\sum_{d=0}^{\infty} \frac{n^2}{2^d}) = O(n^2)$

分析

算法 **FastCut** 成功找到一个最小割的概率为 $\Omega(1/log\ n)$

Pf. 从 n 个点收缩到 $\lceil 1+n/\sqrt{2}\rceil$ 个点最小割仍存在的概率 $\frac{\lceil 1+n/\sqrt{2}\rceil(\lceil 1+n/\sqrt{2}\rceil-1)}{n(n-1)} \geq \frac{1}{2}$
设 $p(k)$ 为从底层往上第 k 层时成功的概率的下界

$$p(k+1) = 1-(1-\frac{1}{2}p(k))^2 = p(k) - \frac{p(k)^2}{4}$$

设 $q(k) = 4/p(k) - 1, p(k) = 4/(q(k)+1)$

$$q(k+1) = q(k) + 1 + \frac{1}{q(k)}$$

$$k < q(k) < k + H_{k-1} + 3$$

$$p(k) = \Theta(1/k)$$

(c,R) - NN
Randomized c-approximate R-near neighbor

问题：给出 d 维空间中的 n 个点，设点集为 P ，给出参数 R 与 δ ，要求回答：给出点 q ，如果存在 P 中点离 q 的距离小于等于 R ，则以 $1-\delta$ 的概率返回 P 中某一个离 q 小于 cR 距离的点。

Locality-Sensitive Hashing

考虑有一族定义在d维空间中的hash 函数$\mathcal{H}$
Definition:$\mathcal{H}$称为 (R,cR,P1,P2)-sensitive 需满足:
$\forall h \in \mathcal{H}, p, q \in R^d$
$if \quad dis(p,q) \leq R, \ then \ Pr[h(p) = h(q)] \geq P1$
$if \quad dis(p,q) \geq cR, \ then \ Pr(h(p) = h(q)] \leq P2$

Locality-Sensitive Hashing

令 $k = log_{p1/p2} n$
$\rho = \frac{ln \;\; p1}{ln \;\; p2}$
$L = n^{\rho} ln \frac{1}{\delta}$
构建 L 个新的hash函数 g_i ，构造 g_i 时，从 $\mathcal{H}$ 中随机 k 个hash函数组合在一起。
$g_i(q) = (h_{i,1}(q), h_{i,2}(q), \ldots, h_{i,k}(q))$

对于原问题，预处理所有点在每个 g_i 中的桶号。查询 q 时找每个 g_i 中与 q 桶号相同的点。

分析

若 $\exists p^*$， $dis(p^*, q) \leq R$， $Pr[g(p^*) = g(q)] \geq p1^k \geq n^{-\rho}$
共有 $L = n^\rho ln\frac{1}{\delta}$ 个 g_i ， $Pr[\exists g_i, g_i(p^*) = g_i(q)] \geq 1 - \delta$

若 $dis(p', q) \geq cR$,
$Pr[g(p') = g(q)|g(p^*) = g(q)] \leq \frac{Pr[g(p')=g(q)]}{Pr[g(p^*)=g(q)]} \leq \left(\frac{P2}{P1}\right)^k \leq \frac{1}{n}$

空间复杂度: $O(dn + n^{1+\rho} kln\frac{1}{\delta})$
询问复杂度: $O(dn^\rho kln\frac{1}{\delta})$
$\rho \approx 1/c$

$\mathcal{H}$的选取

$a = (a_1, a_2, \dots, a_d) \quad a_i \sim N(0,1)$

b在$[0, w]$中随机选取

$h_{a,b}(v) = \lfloor \frac{<a,v>+b}{w} \rfloor$

选择合适的 w 可以使 $\rho \approx 1/c$

期望线性最小生成树

假设无相同的边权（如果有可以按照编号比出大小）
Cycle Property：一个环中最大权的边不会在最小生成树中
Cut Property：任意一个非空子集X，恰只有一端落在X中的最轻边必定在最小生成树中。

期望线性最小生成树

notation:
$w(x,y):(x,y)$的边权
$F(x,y)$: F为森林，$F(x,y)$为 F 中 x 到 y 的路径（若存在）
$w_F(x,y):F(x,y)$上最大边权，若不连通则为 ∞
$F-heavy$: 边(x,y)称为 $F-heavy$ 若 $w(x,y)>w_F(x,y)$
$F-light$: 非 $F-heavy$

期望线性最小生成树

THM: 给出一个图G与森林F，所有 $F-heavy$ 的边能在 $O(n+m)$ 内全部确定。

见参考资料[6][7]

期望线性最小生成树

LEMMA: 设 H 为 G 中每条边独立以 p 的概率选出的子图, F 为 H 的最小生成森林。则 G 中 $F-light$ 的期望边数不超过 n/p ，其中 n 为 G 的点数。

Pf. 用 *Kruskal* 算法的思想，一开始 H, F 为空，边的权值从小到大处理。处理到边 e 时， e 是否为 $F-light$ 在当前是已知的，并且此后也不会改变。再以 p 的概率确定 e 是否被加入 H 中。若 e 为 $F-light$ 且 e 加入了 H ，则 e 理应加入 F 。
由于 $|F| \leq n-1$，所以结论成立。

期望线性最小生成树

定义：Boruvka step：每个点选择与它相邻的最小权的边，将这些边缩起来

算法流程：

$Step1$：做两次Boruvka step

$Step2$：以每条边 1/2 的概率选出一个子图 H，递归求 H 的最小生成森林设为 F，找到所有 $F-heavy$ 的边删除

$Step3$：在剩余的图中递归本算法，连同 $step1$ 中缩起来的边，得到最小生成森林

正确性：
根据Cut Property，Boruvka step中选取的边必定在最小生成森林中
根据Cycle Property，删去的 $F-heavy$ 边不会在最小生成森林中

时间复杂度：
用归纳以及期望线性性：
$T(n,m) \leq T(n/4, m/2) + T(n/4, n/2) + c(n+m)$
期望复杂度:$O(n+m)$

分类

Las Vegas型
Monte Carlo型
对于判定问题，可分为：
1. one-sided：e.g.素数判定
2. two-sided

Monte Carlo

Monte Carlo

- 算圆的面积

Monte Carlo

- 算圆的面积
- 算很复杂度的定积分 ——随机投点法，平均值法

Monte Carlo

使用随机投点法近似时，设所测空间与样本空间大小比值为 m

则测试数量大约与 $\frac{1}{m}$ 成正比

DNF Counting Problem

n个变量，m个子句 $C_1, C_2, ..., C_m$。
每个变量取 *True* 或 *False* ，每个子句是若干变量取反或不取反后 *And* 起来的表达式。
如： $(x_1 \wedge x_2 \wedge \neg x_3 \wedge x_4)$
求有多少种变量的取值使得表达式 $C_1 \vee C_2 \vee ... \vee C_m$ 输出为 *True* .
求 $(\epsilon, \delta) - FPTAS$ 算法

DNF Counting Problem

直接用 Monte Carlo 做效果不佳
小手段:
为表述方便，做一矩阵，横排表示每种变量的取值，共 2^n 排。
列表示每个子句的输出，共 m 列。
问题要求有多少排存在一个子句输出 *True*
给每排第一个 *True* 打上 $*$
求 $*$ 个数。

DNF Counting Problem

直接用 Monte Carlo 做效果不佳
小手段：
为表述方便，做一矩阵，横排表示每种变量的取值，共 2^n 排。
列表示每个子句的输出，共 m 列。
问题要求有多少排存在一个子句输出 *True*
给每排第一个 *True* 打上 $*$
求 $*$ 个数。

问题：

- 新的样本空间？
- 等概率随机？
- 判定？

小练习

[]~(￣▽￣)~*

Static Perfect Hashing

有 n 个元素，设计一种静态的hash-table，满足：
期望 $O(n)$ 预处理
$O(n)$ 的空间
查询最坏复杂度 $O(1)$

Static Perfect Hashing

假设有 n 个元素，开一张大小为 n 的表。
对于每个格子，假如有 B_i 个元素，开 B_i^2 大小的表不停做 *hash* 直至无重复。

二维平面最近点对

给出二维平面上一个大小为 n 的点集，求最近的两个点之间的距离
试用随机算法做到 期望 $O(n)$ 的复杂度

Hint：与最小圆覆盖类似

二维平面最近点对

经典做法：分治，复杂度　$O(nlog\ n)$
随机算法:

- 将点$P_1, P_2, ..., P_n$ 随机排列
- 求出$P_1 ... P_i$的的最近点对 δ，并且维护以 $\delta/2$ 为间隔的平面分割网格。显然每个点落在不同格子内
- 加入 $Pi + 1$ 时查找其所在网格周围25个格子内的信息，更新 δ 与分割网格（必要时重建）。

期望复杂度 $O(n)$

感谢大家的聆听

祝大家考试顺利！

参考文献

1 https://en.wikipedia.org/wiki/Locality-sensitive_hashing
2 https://en.wikipedia.org/wiki/Smallest-circle_problem
3 https://en.wikipedia.org/wiki/Monte_Carlo_algorithm
4 Near-Optimal Hashing Algorithms for Approximate Nearest Neighbor in High Dimensions , Alexandr Andoni and Piotr Indyk
5 A Randomized Linear-Time Algorithm to Find Minimum Spanning Trees,David R. Karger, Philip N. Klein, Robert E. Tarjan [1995]
6 Verification And Sensitivity Analysis Of Minimum Spanning Trees In Linear Time, Brandon Dixon, Monika Rauch, Robert E. Tarjan [1990]
7 Linear Verification For Spanning Trees, J.KOMLOS [1984]
8 Randomized Algorithm, Rajeev Motwani(Stanford University), Prabhakar Raghavan (IBM Thomas J.Watson Research Center)
9 Lijian, Algorithm Design Class