Sublinear Algorithm Examples

3.1 数据流中频繁元素

3.1.1 大数据的数据流模型

数据只能顺序扫描1次或几次
能够使用的内存是有限的
希望通过维护一个内存结果（概要）来给出相关性质的一个有效估计
数据流模型适用于大数据（顺序扫描数据仅一次/内存亚线性）

3.1.2 数据流模型

来自某个域中的元素序列$< x_1, x_2, x_3, x_4, \cdots >$
有限的内存：内存<<数据的规模 通常$O(\log_k n)$或$O(n^\alpha)$ for $a < 1$
快速处理每个元素

3.1.3 频繁元素计算算法（misra gries）

处理元素x
if 已经为x分配计算器，增加之
else if 没有相应计数器，但计数器个数小于k，为x分配计数器，并设为1
else 所有计数器减1，删除值为0的计数器

一个计数器x减少了几次？<=>有几个减少计算器的步骤
整个数据的权重（计数器的和）记作m'
整个数据流的权重（全部元素的数量）是m
每一个计数器降低的步骤减少从k个计数，但是并未计输入元素的此次出现，即$k+1$次未计入的元素出现。

> 最多有$\frac{m-m'}{k+1}$个减少步骤
> 估计值与真实值相差最多$\frac{m-m'}{k+1}$
> 当数据流中元素的总数>>$\frac{m-m'}{k+1}$时，得到x的一个好的估计
> 错误的界限和k成反比

3.2 最小生成树

输入：无向有权联通图$G=(V,E)$，其顶点的度最大为D，边上的权来自整数集合{$1, \cdots, W$}

输出：图G的最小生成树的权重。

3.2.1 精确解

贪心法（prime/kruskal）
时间复杂性：$O(m \log n)$
超过线性

3.2.2 亚线性算法的假设

图组织成邻接表的形式（可以直接访问每个结点的邻居）
可以随即均匀地选择结点

3.2.3 时间亚线性算法的思想

利用特定子图连通分量的数量估计最小生成树的权重
假设所有边的权重都是1或者2，最小生成树的权重
$= \#N_1 + \#N_2$($\#N_i$为最小生成树中权重至少为i的边的数量)
$= n - 1 + \#N_2$(最小生成树有$n-1$ 条边)
$= n - 1 +$ 权重为1边构成的导出子图的联通分量数 -1

3.2.4 最小生成树和连通分量的关系

一般的情况
1.G_i:G中包含所有权重小于i的边的子图
2.C_i:G_i中的连通分量数
3.最小生成树权重大于i的边数为$C_i - 1$

$W_{MST}(G) = n - w + \sum_{i=1}^{w-1} C_i$
证明:
令β_i为最小生成树中权重大于i的边的个数
每一条MST边对WMST基础贡献为1，每个权重大于1的边额外贡献了1，每条权重大于2的边贡献的更多，因此：

$$W_{MST}(G) = \sum_{i=0}^{w-1} \beta_i = \sum_{i=0}^{w-1}(C_i - 1) = -w + \sum_{i=0}^{w-1} C_i = n - w + \sum_{i=1}^{w-1} C_i$$

3.2.5 基础算法：连通分量个数的估计

输入：图G=(V,E)，有n个顶点，表示为邻接矩阵，结点最大度为d。

输出：连通分量的个数

精确解时间复杂性：$O(dn)$

利用随机化方法

估计联通分量个数$\#CC$
$\#CC \pm \epsilon$的概率 $\geq 2/3$
运行时间和n无关

C:连通分量的个数
对于每个结点u, n_u:u所在连通分量的结点数
对于每个连通分量：$\sum_{u \in A} \frac{1}{n_u} = 1$
故：$\sum_{u \in V} \frac{1}{n_u} = C$

通过估计抽样顶点的n_u来估计这个和
如果u所在的分量很小，其规模可以通过BFS估计
如果u所在的连通分量很大，$1/n_u$很小，对和的贡献很小

每个u所在连通分量结点数的估计
令$\hat{n}_u = \min(n_u, 2/\epsilon)$
在这种情况下，对C的估计

$$\hat{C} = \sum \frac{1}{\hat{n}_u}$$

则

$$|\hat{C} - C| = |\sum(\frac{1}{\hat{n}_u} - \frac{1}{n_u})| \leq \frac{\epsilon n}{2}$$

3.2.6 连通分量数估计算法

$CC(G,d,\epsilon)$
1.$for\ i=1\ to\ s=\theta(\frac{1}{\epsilon^2})\ do$
2. 随机选择点u
3. 从u开始BFS，将访问到的顶点存到排序序列L中，访问完连通分量或$L=2/\epsilon$时停止，$\hat{n}_u=|L|$
4. $N=N+\hat{n}_u$
5.返回$\hat{C}=s/N*n$

运行时间:$O(\frac{d}{\epsilon^3}\log\frac{1}{\epsilon})$

连通分量近似计数的分析
分析的目的:

$$Pr[|\widetilde{C}-\hat{C}|>\frac{\epsilon n}{2}]\leq\frac{1}{3}$$

估计值和真实值相差过大的概率很小

对于采样中的第i个结点u，令

$$Y_i=\frac{1}{\hat{n}_u}$$

$$Y=\sum_{i=1}^{s}Y_i=\frac{s\widetilde{C}}{n}$$

$$E[Y]=\sum_{i=1}^{s}E[Y_i]=sE[Y_1]=s\cdot\frac{1}{n}\sum_{u\in V}\frac{1}{\hat{n}_v}=\frac{s\widetilde{C}}{n}$$

$$Pr[|\widetilde{C}-\hat{C}|>\frac{\epsilon n}{2}]=Pr[|\frac{n}{s}Y-\frac{n}{s}E[Y]|>\frac{\epsilon n}{2}]=Pr[|Y-E[Y]|>\frac{\epsilon s}{2}]$$

Hoeffding界：$Y_1,\cdots,Y_s$为[0,1]区间内独立同分布的随机变量，令$Y=\sum_{i=1}^{s}Y_i$，则$Pr[|Y-E[Y]|\geq\delta\leq 2e^{-2\delta^2/s}]$

$$Pr[|\widetilde{C}-\hat{C}|>\frac{\epsilon n}{2}]=Pr[|Y-E[Y]|>\frac{\epsilon s}{2}]\leq 2e^{-\frac{\epsilon^2 s}{2}}$$

$$s=\theta(\frac{1}{\epsilon^2})=>Pr[|\widetilde{C}-\hat{C}|>\frac{\epsilon\epsilon n}{2}]\leq\frac{1}{3}$$

得出:

$$Pr[|\widetilde{C}-\hat{C}|>\frac{\epsilon n}{2}]\leq\frac{1}{3}$$

$$|\widetilde{C}-C|\leq\frac{\epsilon n}{2}$$

因此，下列事件发生的概率大于2/3:

$$|\widetilde{C}-C|\leq|\widetilde{C}-\hat{C}|+|\hat{C}-C|\leq\frac{\epsilon n}{2}+\frac{\epsilon n}{2}=\epsilon n$$

综上所述，有n个顶点的途中，若其顶点的度至多为d，则其连通分量的数量估计误差最多为$+\epsilon n$

3.2.7 最小生成树近似算法

$for\ i=1\ to\ w-1\ do$
$\quad\hat{C}_i=CC(G_i,d,\frac{\epsilon}{w})$
$return\ \widetilde{w}_{MST}=n-w+\sum_{i=1}^{w-1}\widetilde{C}_i$

假设C_i的所有估计都是正确的，$|\widetilde{C}_i - C_i| \le \frac{\epsilon}{w} n$，则$|\widetilde{w}_{MST} - w_{MST}| = |\sum_{i=1}^{w-1}(\widetilde{C}_i - C_i)| \le w \cdot \frac{\epsilon}{w} n = \epsilon n$
$Pr[所有w-1次估计都正确] \ge (2/3)^{w-1}$

3.3 序列有序的判定

输入：n个数的数组

输出：这个数组是否有序

近似版本：这个数组是有序的还是$epsilon$远离有序的

ϵ**远离**：必须删除大于ϵn个元素才能保证剩下的元素有序

亚线性算法
$for\ k = 1\ to\ w/\epsilon\ do$
$\quad 选择数组中第i个元素x_i$
$\quad 用x_i在数组中做二分查找$
$\quad if\ 发现i < j\ 但是x_i > x_j\ then$
$\quad\quad return\ false$
$return\ true$

算法的时间复杂性$O(\frac{1}{\epsilon}\log n)$

输入数组有序，则总返回True
下面证明：当输入数列ϵ远离有序时，算法返回false的概率大于2/3
首先证明：如果输入ϵ远离有序，则存在大于ϵn个坏索引
证明其逆否命题，即如果坏索引的个数小于ϵn，则其存在一个长度大于ϵn的单调递增子序列
对于任意好索引i和j，$x_i < x_j$
令k是在二分搜索中将x_i和x_j分开的最近顶点，也就是对于整个数组建一个二分搜索树，在二分搜索树中x_i和x_j的最近公共祖先，则$i < k < j$，因为i和j都是好索引，那么$x_i < x_k < x_j$

当输入数列ϵ远离有序时，算法返回false的概率大于2/3
证明：算法返回true的概率小于1/3
已经证明，如果输入ϵ远离有序，则存在大于ϵn个坏索引，即数组中坏索引的概率大于ϵ
当数组中坏索引的概率大于ϵ时，选择的索引都是好的概率小于$(1-\epsilon)^{2/\epsilon} < e^{-2} < \frac{1}{3}$