HIT CS&E

# 第6章 随机抽样和随机舍入

骆吉洲
计算机科学与技术学院

HIT CS&E

## 提纲



HIT CS&E

## 参考文献

- 《概率与计算》
  第10章
- 《Design and Analysis of Randomized Algorithm》
  第7章

HIT CS&E

## 6.1 随机游走

- 8.1.1 SAT问题的随机赋值算法分析
- 8.1.2 马尔科夫链
- 8.1.3 图上的随机游走

HIT CS&E

## 6.1.1 SAT问题的随机赋值算法

HIT CS&E

## 变量、文字和子句

变量：表示随机事件是否发生的量

文字：符号化的变量
- $x$ = 钱多
- $y$ = 事少
- $z$ = 离家近
- $w$ = 睡觉睡到自然醒

子句：用∧∨¬连接的若干文字
- $\neg x \vee \neg y$ 析取
- $\neg y \wedge \neg x$ 合取
- $y \vee \neg w$
- $\neg w \vee \neg y$
- $\neg z \vee x$
- $\neg z \vee \neg y$

形式演算
- $x \Rightarrow \neg y$
- $\neg y \not\Rightarrow x$
- $\neg y \Rightarrow \neg w$
- $w \Rightarrow \neg y$
- $z \Rightarrow x$
- $z \Rightarrow \neg y$

## 析取子句的表示和存储

变量：$x_1, x_2, \ldots, x_n$

析取子句$C$的表示：两个子集$C^+, C^- \subseteq \{1,2,\ldots n\}$

- $C^+$表示子句中不带否定算符的文字
- $C^-$表示子句中带否定算符的文字
- $|C| = |C^+| + |C^-|$是子句$C$中文字的个数

例：$x_1, x_2, x_3, x_4, x_5, x_6$

$C = <\{1,3\}, \{2,5,6\}>$表示如下子句

$x_1 \vee \neg x_2 \vee x_3 \vee \neg x_5 \vee \neg x_6$

$|C| = 2+3 = 5$

## k-SAT问题

*k*-SAT问题

输入：文字$x_1, x_2, \ldots, x_n$及其上的$m$个析取子句$C_1, \ldots, C_m$,

$|C_i| \leq k$ 对$i=1,2,\ldots,m$均成立

输出：是否存在$x_1, x_2, \ldots, x_n$的赋值使得$C_1, \ldots, C_m$均被满足

例 1 输入：$x_1 \vee \neg x_3$, $\neg x_2 \vee x_3$ , $\neg x_1 \vee \neg x_2$ , $x_2 \vee x_3$

输出：Yes ($x_1$=T, $x_2$=F, $x_3$=T)

例2 输入：$x_1 \vee \neg x_2$, $x_2 \vee \neg x_3$ , $x_3 \vee \neg x_1$ , $\neg x_1 \vee \neg x_3$

输出：Yes ($x_1$=F, $x_2$=F, $x_3$=F)

例3 输入：$x_1 \vee x_2$, $x_1 \vee \neg x_2$ , $x_3 \vee \neg x_1$ , $\neg x_1 \vee \neg x_3$

输出：No

## k-SAT问题的两个事实

*k*-SAT问题

输入：文字$x_1, x_2, \ldots, x_n$及其上的$m$个析取子句$C_1, \ldots, C_m$,

$|C_i| \leq k$ 对$i=1,2,\ldots,m$均成立

输出：是否存在$x_1, x_2, \ldots, x_n$的赋值使得$C_1, \ldots, C_m$均被满足

事实1： 3-SAT问题是NP-完全问题

还未找到求解3-SAT问题的多项式时间算法

事实2： 2-SAT问题存在多项式时间算法

*Aspvall-Plass-Tarjan, Information Processing Letters, 1979*

该算法很复杂

## 2-SAT问题的随机赋值算法

2-SAT问题的随机赋值算法[*Papadimitriou*: Focs 1991]

输入：文字$x_1, x_2, \ldots, x_n$及其上的$m$个析取子句$C_1, \ldots, C_m$,

$|C_i| \leq 2$ 对$i=1,2,\ldots,m$均成立

输出：是否存在$x_1, x_2, \ldots, x_n$的赋值使得$C_1, \ldots, C_m$均被满足

1.任取$x_1, x_2, \ldots, x_n$的一个布尔赋值

2.For $i$ = 1 To $N$ Do

3. If 当前赋值满足所有子句 Then 输出当前赋值并停止 $O(m)$

4. Else //设当前赋值不满足$C_j$

5. 均匀随机地取出$C_j$的一个变量$x_k$, 将$x_k$的赋值取反 $O(1)$

6.输出“无法满足” //结论不一定可靠

问题：$N$取多大，才能在$O(Nm)$时间内高概率得出正确解？

## 分析思路

2-SAT问题的随机赋值算法特点

算法输出“Yes”则结论可靠

算法输出“No”则结论不一定可靠

算法出错仅有一种可能：问题有解$S$,但算法未找到

分析思路

选取恰当的$N$值让算法出错的概率尽可能小

固定问题的一个满足性赋值$S$

算法循环变量为$i$时的赋值为$A_i$

若$A_i$与S完全一致，则算法不出错

若$A_i$与S不一致，$A_i$也可能是正确解

Pr[算法出错] ≤1- Pr[算法将$A_1$演变到$S$]

| $x_1$ | $x_2$ | $x_3$ | $x_4$ | | $x_{n-1}$ | $x_n$ |
|---|---|---|---|---|---|---|
| *T* | *F* | *F* | *F* | … | *F* | *T* |
| *F* | *T* | *T* | *F* | | *T* | *T* |

## 演变过程的状态表示

令$X_i$表示$x_1, \ldots, x_n$在$A_i$与S中具有一致取值的变量个数

S是固定的

$A_i$是随算法运行过程而变化的，是随机量

$X_i$是随机变量

$X_i$取值介于0~$n$之间

$X_1 \geq 0$

$X_i = n$表明$A_i = S$

| | $x_1$ | $x_2$ | $x_3$ | $x_4$ | | $x_{n-1}$ | $x_n$ |
|---|---|---|---|---|---|---|---|
| $S$ | *T* | *F* | *F* | *F* | … | *F* | *T* |
| $A_i$ | *F* | *T* | *T* | *F* | | *T* | *T* |

HIT CS&E

## $X_i$的变化规律

若$X_i$=0，则$X_{i+1}$=1

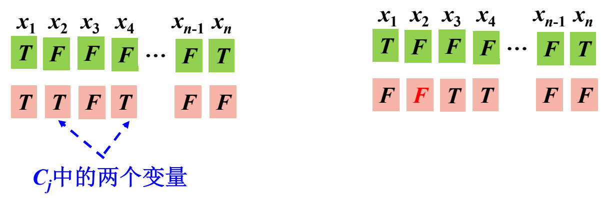

HIT CS&E

## $X_i$的变化规律

若$X_i$=$k$，则$X_{i+1}$=?

情形1：$C_j$中两个变量取值均与$S$中不一致

$X_{i+1} = k+1$




HIT CS&E

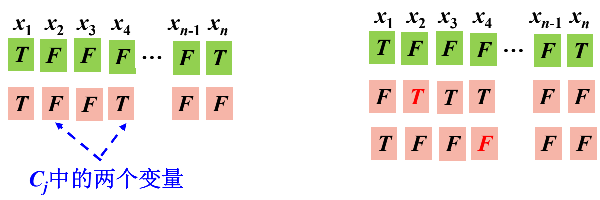
## $X_i$的变化规律

若$X_i$=$k$，则$X_{i+1}$=?

情形2：$C_j$中两个变量仅有一个与$S$中不一致

$X_{i+1} = k-1$　　　Pr[ $X_{i+1} = X_i$-1] = 1/2

$X_{i+1} = k+1$　　　Pr[ $X_{i+1} = X_i$+1] = 1/2




HIT CS&E

## $X_i$的变化规律

若$X_i$=$k$，则$X_{i+1}$=?

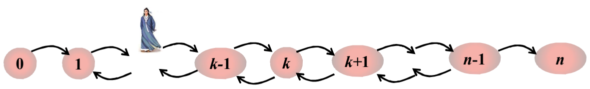
Pr[ $X_{i+1} = X_i$+1] = Pr[$X_{i+1} = X_i$+1| 情形1] Pr[情形1]
+ Pr[$X_{i+1} = X_i$+1| 情形2] Pr[情形2]
= 1·Pr[情形1]+ (1/2) Pr[情形2]
≥ ½(Pr[情形1]+ Pr[情形2])
= 1/2

Pr[ $X_{i+1} = X_i$-1] = 1- Pr[ $X_{i+1} = X_i$+1]
≤ 1/2

HIT CS&E

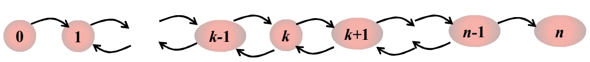
## $A_1$到$S$的演变

凌波微步到达$n$平均需要多长时间？




前进概率不低于1/2，后退概率不高于1/2

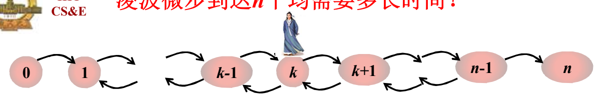
哪个需要的时间更长一些？　后者的时间更长更保守些




前进概率等于1/2，后退概率等于1/2

HIT CS&E

凌波微步到达$n$平均需要多长时间？




前进概率等于1/2，后退概率等于1/2

令 $h_k$表示从状态$k$达到状态$n$需要的时间

$h_k$是随机变量

$h_k$= 1+$h_{k-1}$　　这种情况发生的概率是1/2

$h_k$= 1+$h_{k+1}$　　这种情况发生的概率是1/2

E[$h_k$] = 1+ E[$h_{k-1}$]/2+E[$h_{k+1}$]/2

E[$h_0$] = 1+ E[$h_1$]

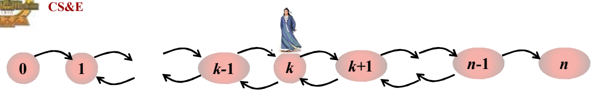
E[$h_n$] = 0

归纳证明：

E[$h_k$] = E[$h_{k+1}$]+2$k$+1

E[$h_0$] =E[$h_1$]+1= 1+3+E[$h_2$]=…= 1+3+5+…+(2$n$+1)= $n^2$

HIT CS&E




前进概率等于1/2，后退概率等于1/2

结论1：如果布尔表达式是可满足的，则从任意布尔赋值开始找到满足性赋值平均需要运行3-5步$n^2$遍

结论2：如果布尔表达式是可满足的，则从任意布尔赋值开始运行3-5步$2n^2$遍仍未找到满足性赋值的概率不超过1/2

$\Pr[h_0>2n^2] \le E[h_0]/(2n^2) \le ½$　Markov不等式

结论3：如果布尔表达式是可满足的，则从任意布尔赋值开始运行3-5步$2kn^2$遍找到满足性赋值的概率至少为$1-1/2^k$

HIT CS&E

## 2-SAT问题的随机赋值算法

2-SAT问题的随机赋值算法[*Papadimitriou*: Focs 1991]

输入：文字$x_1,x_2,\ldots,x_n$及其上的$m$个析取子句$C_1,\ldots,C_m$，$|C_i|\le 2$ 对$i$=1,2,…,$m$均成立

输出：是否存在$x_1,x_2,\ldots,x_n$的赋值使得$C_1,\ldots,C_m$均被满足

1.任取$x_1,x_2,\ldots,x_n$的一个布尔赋值
2.For $i$ = 1 To $2kn^2$ Do
3.　If 当前赋值满足所有子句 Then 输出当前赋值并停止　$O(m)$
4.　Else　　　　　　　　//设当前赋值不满足$C_j$
5.　　均匀随机地取出$C_j$的一个变量$x_k$, 将$x_k$的赋值取反　$O(1)$
6.输出“无法满足”　　//结论不一定可靠

结论：算法在$O(2kmn^2)$时间内找到正确解的概率至少为$1-1/2^k$

HIT CS&E

## 推广到3-SAT问题

3-SAT问题的随机赋值算法

输入：文字$x_1,x_2,\ldots,x_n$及其上的$m$个析取子句$C_1,\ldots,C_m$，$|C_i|\le 3$ 对$i$=1,2,…,$m$均成立

输出：是否存在$x_1,x_2,\ldots,x_n$的赋值使得$C_1,\ldots,C_m$均被满足

1. 任取$x_1,x_2,\ldots,x_n$的一个布尔赋值
2. For $i$ = 1 To $N$ Do
3. 　If 当前赋值满足所有子句 Then 输出当前赋值并停止$O(m)$
4. 　Else　　　　　　　　//设当前赋值不满足$C_j$
5. 　　均匀随机地取出$C_j$的一个变量$x_k$, 将$x_k$的赋值取反$O(1)$
6. 输出“无法满足”　　//结论不一定可靠

问题：$N$取多大，才能在$O(Nm)$时间内高概率得出正确解？

HIT CS&E

## $X_i$的变化规律

若$X_i$=0，则$X_{i+1}$=1

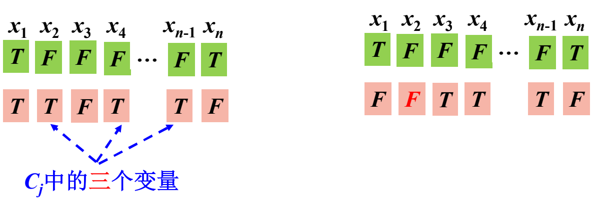

HIT CS&E

## $X_i$的变化规律

若$X_i=k$，则$X_{i+1}$=?

情形1：$C_j$中三个变量取值均与$S$中不一致

$X_{i+1} = k+1$




HIT CS&E

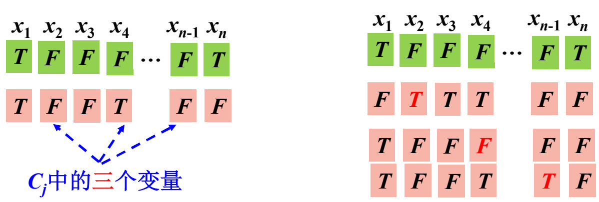
## $X_i$的变化规律

若$X_i=k$，则$X_{i+1}$=?

情形2：$C_j$中三个变量仅有一个与$S$中不一致

$X_{i+1} = k-1$　　$\Pr[X_{i+1} = X_i-1] = 2/3$

$X_{i+1} = k+1$　　$\Pr[X_{i+1} = X_i+1] = 1/3$



HIT CS&E

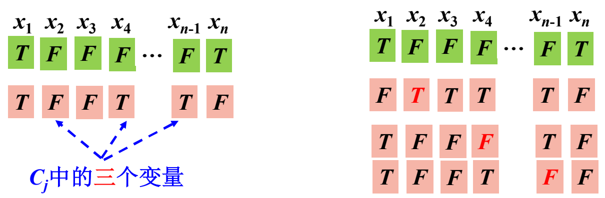
## $X_i$的变化规律

若$X_i=k$，则$X_{i+1}$=？

情形3：$C_j$中三个变量仅有两个与$S$中不一致

$X_{i+1}=k$-1　　　Pr[ $X_{i+1}=X_i$-1] = 1/3

$X_{i+1}=k$+1　　　Pr[ $X_{i+1}=X_i$+1] = 2/3




HIT CS&E

## $X_i$的变化规律

若$X_i=k$，则$X_{i+1}$=？

Pr[ $X_{i+1}=X_i$+1] = Pr[$X_{i+1}=X_i$+1| 情形1] Pr[情形1]

+ Pr[$X_{i+1}=X_i$+1| 情形2] Pr[情形2]

+ Pr[$X_{i+1}=X_i$+1| 情形3] Pr[情形3]

=1·Pr[情形1]+ (1/3) Pr[情形2]+(2/3) Pr[情形3]

≥ 1/3 · (Pr[情形1]+ Pr[情形2] + Pr[情形3])

= 1/3

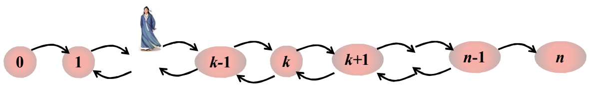
Pr[ $X_{i+1}=X_i$-1] = 1- Pr[ $X_{i+1}=X_i$+1]

≤ 2/3

HIT CS&E

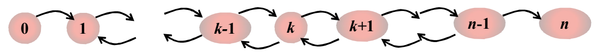
## $A_1$到$S$的演变

凌波微步到达$n$平均需要多长时间？




前进概率不低于1/3，后退概率不高于2/3

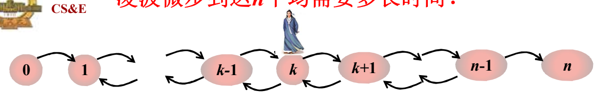
哪个需要的时间更长一些？　后者的时间更长更保守些




前进概率等于1/3，后退概率等于2/3

HIT CS&E

凌波微步到达$n$平均需要多长时间？




前进概率等于1/3，后退概率等于2/3

令 $h_k$表示从状态$k$达到状态$n$需要的时间

$h_k$是随机变量

$h_k$= 1+$h_{k-1}$　　这种情况发生的概率是2/3

$h_k$= 1+$h_{k+1}$　　这种情况发生的概率是1/3

E[$h_k$] = 1+ 2E[$h_{k-1}$]/3+E[$h_{k+1}$]/3

E[$h_0$] = 1+ E[$h_1$]

E[$h_n$] = 0

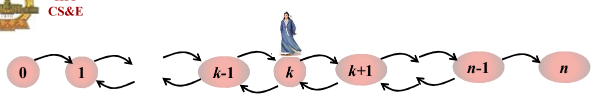
E[$h_k$] =$2^{n+2}$-$2^{k+2}$-3($n$-$k$)

归纳证明：

E[$h_k$] = E[$h_{k+1}$]+$2^{k+2}$-3

E[$h_0$] =$2^{n+2}$-3$n$-4

HIT CS&E




前进概率等于1/3，后退概率等于2/3

结论：如果布尔表达式是可满足的，则从任意布尔赋值开始找到满足性赋值平均需要运行3-5步$2^n$遍

布尔赋值总共才只有$2^n$种！

失败原因

前进概率小，后退概率大

运行时间越长，越有可能到达状态0，而非状态$n$

随机选取一个赋值，平均也将有$n$/2个变量与$S$一致

路在何方？

## 修正的3-SAT随机赋值算法

输入：文字$x_1,x_2,\ldots,x_n$及其上的$m$个析取子句$C_1,\ldots,C_m$，

$|C_i|$≤3 对$i$=1,2,…,$m$均成立

输出：是否存在$x_1,x_2,\ldots,x_n$的赋值使得$C_1,\ldots,C_m$均被满足

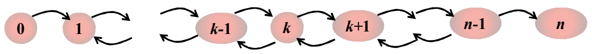
1. For $k$=1 To $N$ Do
2. 　任取$x_1,x_2,\ldots,x_n$的一个布尔赋值
3. 　For $l$ = 1 To 3$n$ Do
4. 　　If 当前赋值满足所有子句 Then 输出它并停止　　$O(m)$
5. 　　Else　　　//设当前赋值不满足$C_j$
6. 　　　均匀随机地取出$C_j$的一个变量并将其赋值取反　　$O(1)$
7. 输出“无法满足”　　//结论不一定可靠

问题：$N$取多大，才能在$O(Nmn)$时间内高概率得出正确解

## 随机赋值经3$n$步修正得到正确解的概率

- 假设3SAT是可满足的
- $S$是一个满足性赋值
- $Y$是算法第2步所取随机赋值与S不一致的变量个数
- $Y$=0,1,2,3,...是随机变量
- $q$是所取随机赋值经3$n$步修正得到$S$的概率
- $q_j$是在$Y=j$的条件下随机赋值经3$n$步修正得到$S$的概率

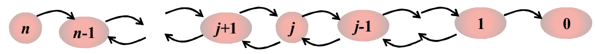
$$q = \Sigma_j\, q_j \cdot \Pr[\ Y=j\ ]$$




前进概率等于1/3，后退概率等于2/3

## $q_j$的计算

HIT CS&E




右行概率等于1/3，左行概率等于2/3

要将随机赋值修正到正确赋值$S$

左行$k$步

右行$j+k$

整体效果是右行$j$步

$j+2k \le 3n$

$$\binom{j+2k}{k}\left(\frac{2}{3}\right)^k\left(\frac{1}{3}\right)^{j+k}$$

$$q_j \ge \max_k \binom{j+2k}{k}\left(\frac{2}{3}\right)^k\left(\frac{1}{3}\right)^{j+k} \ge \binom{3j}{j}\left(\frac{2}{3}\right)^j\left(\frac{1}{3}\right)^{2j} \ge \frac{\sqrt{3}}{2\sqrt{\pi}}\frac{1}{\sqrt{j}}\frac{1}{2^j}$$

## 随机赋值经3$n$步修正得到正确解的概率

$$q = \sum_{j=0}^{n} q_j \cdot \Pr[\ Y=j\ ] \qquad q_j \ge \frac{\sqrt{3}}{2\sqrt{\pi}}\frac{1}{\sqrt{j}}\frac{1}{2^j}$$

$$\ge \frac{1}{2^n} + \sum_{j=1}^{n}\frac{c}{\sqrt{j}}\frac{1}{2^j}\binom{n}{j}\left(\frac{1}{2}\right)^j\left(\frac{1}{2}\right)^{n-j}$$

$$\ge \frac{c}{\sqrt{n}}\frac{1}{2^n}\sum_{j=0}^{n}\binom{n}{j}\left(\frac{1}{2}\right)^j 1^{n-j}$$

$$\ge \frac{c}{\sqrt{n}}\frac{1}{2^n}\left(\frac{3}{2}\right)^n$$

$$= \frac{c}{\sqrt{n}}\left(\frac{3}{4}\right)^n$$

## 随机赋值经3$n$步修正得到正确解的概率

$$q \ge \frac{c}{\sqrt{n}}\left(\frac{3}{4}\right)^n$$

“随机赋值+3$n$次修正”得到正确解的概率至少为$q$

由几何分布的数学期望公式可知
“随机赋值+3$n$次修正”平均要重复1/$q$次才能得到正确解

由Markov不等式可知
“随机赋值+3$n$次修正”重复2/$q$次得到正确解的概率大于1/2

由概率放大过程可知
“随机赋值+3$n$次修正”重复2$k$/$q$次得正确解的概率大于1-1/$2^k$

## 修正的3-SAT随机赋值算法

输入：文字$x_1,x_2,\ldots,x_n$及其上的$m$个析取子句$C_1,\ldots,C_m$，
$|C_i|\le 3$ 对$i$=1,2,…,$m$均成立

输出：是否存在$x_1,x_2,\ldots,x_n$的赋值使得$C_1,\ldots,C_m$均被满足

1. For $k$=1 To $2cKn^{1/2}(4/3)^n$ Do
2. 　任取$x_1,x_2,\ldots,x_n$的一个布尔赋值
3. 　For $l$ = 1 To 3$n$ Do
4. 　　If 当前赋值满足所有子句 Then 输出它并停止　$O(m)$
5. 　　Else　　//设当前赋值不满足$C_j$
6. 　　　均匀随机地取出$C_j$的一个变量并将其赋值取反　$O(1)$
7. 输出“无法满足”　//结论不一定可靠

结论：算法在$O(mn^{3/2}(4/3)^n)$时间内未找到正确解的概率≤ $2^{-K}$

## 本小节回顾

HIT CS&E

收获之一：一种典型的随机算法设计过程
- 先处理简单的2SAT
- 推广过程简单算法去处理难解问题
- 设法克服推广过程中遇到的困难

收获之二：一种可能值得一般化的工具—随机游走
- 2SAT时用过
- 算法推广到3SAT时用过
- 改进推广的3SAT随机赋值算法时也用过

收获之三：工具的综合应用
- 基本概率计算
- 几何分布
- 马尔科夫不等式
- 概率放大
- 参数化设计

HIT CS&E

# 6.1.2 马尔科夫链

- 请大家复习《随机计算》第5章

HIT CS&E

# 6.1.3 图上的随机游走

- 由无向连通图导出的特殊马尔科夫链
- 分析随机算法的一种强有力的工具

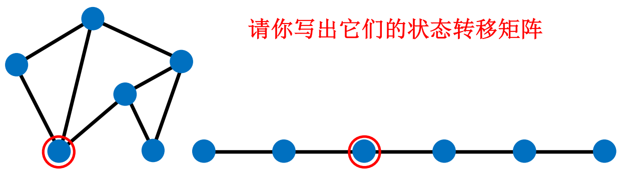
HIT CS&E

## 随机游走的定义

定义：给定无向连通图$G=(V=\{1,2,\ldots,n\},E)$, $G$上的随机游走是粒子受限于G中顶点邻接关系而在顶点间进行一系列随机移动诱导得出的马尔科夫链

- 粒子在$t$时刻的状态$X_t$是指该时刻粒子所在的顶点
- 若$X_t=i$，则$X_{t+1}$均匀分布于$N(i)=\{j| ij\in E\}$

请你写出它们的状态转移矩阵



HIT CS&E

## 二分图上的随机游走

从任意状态$j$开始

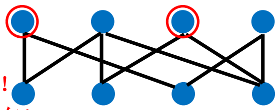
下一时刻的状态不可能是$j$

奇数时刻后的状态不可能是$j$

偶数时刻后的状态有可能是$j$

$\Pr[X_{t+s}=j| X_t=j] = 0$ If s = 1 mod 2

$\Pr[X_{t+s}=j| X_t=j] > 0$ If s = 0 mod 2



二分图上的随机游走是周期的！

其他图上的随机游走是非周期的！

其他图上的随机游走存在稳定分布！

HIT CS&E

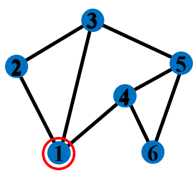
## 随机游走的稳定分布

定理：给定非二分无向连通图$G=(V=\{1,2,\ldots,n\},E)$, $G$上的随机游走的稳定分布是 $\pi_v = d(v)/2|E|$, $d(v)$表示顶点$v$的度

$\pi_v>0$ 且 $\Sigma_v\, \pi_v= 1$， 故$\pi$是一个概率分布

如果将随机游走的状态转移矩阵记为P，则$\pi= \pi P$

$$\pi_v = \sum_{u\in N(v)} \frac{d(u)}{2|E|}\,\frac{1}{d(u)} = \frac{d(v)}{2|E|}$$




"随机游走一段时间，取出状态"

重复操作相当于根据分布$\pi$对顶点抽样

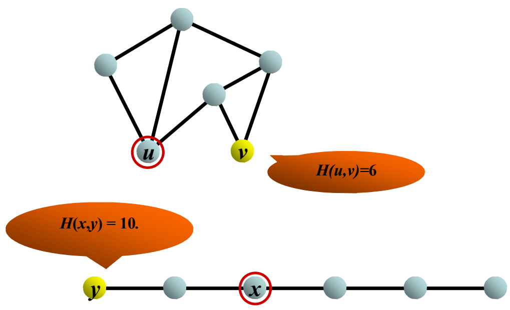
…

到底多长时间？

HIT CS&E

## Hitting Time

$H(u,v)$：从顶点$u$出发的随机游走首次访问顶点$v$的期望时间



## $H(v,v)$

结论：在任意不可约非周期正常返的有穷马尔科夫链存在稳定分布$\pi$，且 $\pi_i = 1/H(i,i)$

在随机游走中，由于

$$\pi_v = \frac{d(v)}{2|E|}$$

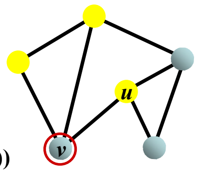
故

$$H(v,v) = \frac{2|E|}{d(v)}$$

## $H(v,u)$的简单情况

$$H(v,v) = \frac{2|E|}{d(v)}$$

$(v,u)\in E$的情况




$$H(v,v) = \frac{2|E|}{d(v)}$$

$$H(v,v) = \frac{1}{d(v)}\sum_{u\in N(v)}(1+H(u,v))$$

$$2|E| = \sum_{u\in N(v)}(1+H(u,v))$$

若$(v,u)\in E$，则$H(v,u)\le 2|E|$

## 简单情况导出的覆盖时间上界

若$(v,u)\in E$，则$H(v,u)\le 2|E|$

Cover($u$)：从$u$出发的随机游走遍历所有顶点的期望时间

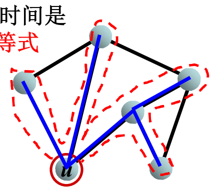
Cover($G$) = $\max_u$ Cover($u$) 称为图$G$的覆盖时间

为了得到Cover($u$)的上界，取定一棵以$u$为根的生成树

生成树的每条边走两遍，得到一个欧拉回路 $u, u_1,\ldots,u_{n-1},u$

随机游走完成欧拉回路 $u, u_1,\ldots,u_{n-1},u$的时间是

Cover($u$)的上界， 因为H(,)满足三角不等式




回路中共有2(|$V$|-1)条边

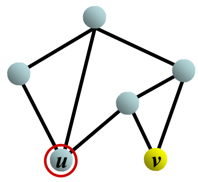
每条边的Hitting time不超过2|$E$|

Cover($u$) $\le$ 2(|$V$|-1) 2|E| < 4|$V$||$E$|

随机游走的覆盖时间不超过4|$V$||$E$|

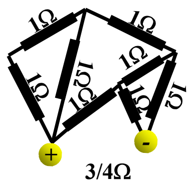
## $H(u,v)$的计算

计算$H(u,v)$

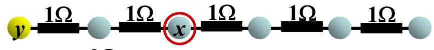



$H(u,v)$ = |E|×3/4 = 6

计算等效电阻




3/4Ω




2Ω

$H(x,y)$ = |E|×2 = 10

## $H(u,v)$的计算

定理[Chandra et al. 1989 STOC]

将图$G(V,E)$上随机游走的从$u$到$v$的Hitting time记为$H(u,v)$

将图$G(V,E)$视为电路，输入点为$u$,输出点为$v$, R($u,v$)表示两点间的电阻

则有

$$H(u,v)+H(v,u) = 2|E|R(u,v) \quad (*)$$

由于

$$H(u,v) = H(v,u)$$

故

$$H(u,v) = |E|R(u,v)$$

证明思路：证明（*）两端满足同一组方程，请查阅原文

## 简单应用

2-SAT随机赋值算法中，曾遇到如下的随机游走


$$H(0,n) = |E|R(0,n)$$
$$= n\times n$$
$$= n^2$$

HIT CS&E

# 6.2 基于随机抽样的算法

- 7.2.1 非二次剩余的随机抽样算法
- 7.2.2 水库抽样算法

HIT CS&E

# 6.2.1 搜寻二次非剩余

HIT CS&E

## 非二次剩余

二次剩余和二次非剩余

- 素数$p(p>2)$
- 二次剩余$x$： $x=a^2 \bmod p$ 对$a\in\{1,\ldots,p-1\}$成立
- 非二次剩余$x$：$x\neq a^2 \bmod p$ 对任意$a\in\{1,\ldots,p-1\}$成立

例

$p=5$，二次剩余{1,4}，非二次剩余{2,3}

$p=11$， $1^2= 1 \bmod 11$ $2^2=4 \bmod 11$ $3^2=9 \bmod 11$

$4^2= 5 \bmod 11$ $5^2=3 \bmod 11$ $6^2=3 \bmod 11$

$7^2= 5 \bmod 11$ $8^2=9 \bmod 11$ $9^2=4 \bmod 11$

$10^2=1 \bmod 11$

二次剩余{1,3,4,5,9}， 非二次剩余{2,6,7,8,10}

问题：$p$是4096位的整数，怎么找到非二次剩余？ 

HIT CS&E

## 非二次剩余寻找问题

问题定义

输入：素数$p(p>2)$

输出：$x\in\{1,2,\ldots,p-1\}$使得

$x\neq a^2 \bmod p$ 对任意$a\in\{1,\ldots,p-1\}$成立

常见应用

- 椭圆曲线加密算法
- Rabin公钥密码

计算难点

- 任意$x$，计算$x^2 \bmod p$可以高效完成
- 当$p$非常大时，二次剩余很多
- 枚举$\{1,2,\ldots,p-1\}$中元素可以发现非二次剩余
- 枚举过程的时间开销很大

HIT CS&E

## 非二次剩余抽样算法

抽样算法

输入：素数$p(p>2)$

输出：$x\in\{1,2,\ldots,p-1\}$使得$x$是$p$的非二次剩余

1. While (TRUE) do
2. 　从$\{1,2,\ldots,p-1\}$中均匀随机地抽取一个元素$x$
3. 　If $x$是$p$的非二次剩余 Then return $x$

这样也行？ 

问题：

第3步中的判断如何进行，有高效过程吗？

算法需要多长时间找到一个非二次剩余？

样本空间中有多少非二次剩余？

HIT CS&E

## 非二次剩余的判定

费尔马小定理：

若$p$是素数，则$a^{p-1}=1 \bmod p$对任意自然数$a$成立

由于$p>2$是素数，故$p$必然是奇数

设$p=2k+1$，则 $k=(p-1)/2$

$a^{p-1}=(a^{(p-1)/2})^2 = 1 \bmod p$ 　　$-1 = p-1 \bmod p$

$a^{(p-1)/2} = 1 \bmod p$ 或 $a^{(p-1)/2} = -1 \bmod p$对任意自然数$a$成立

若$x$是二次剩余，则$x=a^2 \bmod p$

$x^{(p-1)/2}=(a^2)^{(p-1)/2}=a^{p-1}=1 \bmod p$

若$x$是二次剩余，则$x^{(p-1)/2}=1 \bmod p$

HIT CS&E

## $a^{(p-1)/2} = 1$ 或 $a^{(p-1)/2} = -1$ mod $p$对任意自然数$a$成立

若$x$是非二次剩余

乘法群{1,2,…,$p$-1}是周期群，故存在生成元素$g$

于是 $x = g^{2l+1}$ (奇数次幂而非偶数次幂，$x$是非二次剩余)

$x^{(p-1)/2} = (g^{2l+1})^{(p-1)/2}$ mod $p$

$=(g^{l})^{(p-1)} g^{(p-1)/2}$ mod $p$

$=1\cdot g^{(p-1)/2}$ mod $p$ 费尔马小定理

$= g^{(p-1)/2}$ mod $p$ $g$的周期是$p$-1

$= -1$ mod $p$

若$x$是非二次剩余，则$x^{(p-1)/2} = -1$ mod $p$

HIT CS&E

## 非二次剩余抽样算法

抽样算法

输入：素数$p$($p$>2)

输出：$x\in${1,2,…,$p$-1}使得$x$是$p$的非二次剩余

1. While (TRUE) do
2. 从{1,2,…,$p$-1}中均匀随机地抽取一个元素$x$
3. If $x^{(p-1)/2} = -1$ Then return $x$

计算提速：

$x$ mod $p$

$x^2$ mod $p$

$x^4$ mod $p$

$x^8$ mod $p$

….

$x^{(p-1)/2}$ mod $p$

算法一遍成功的概率有多高？

样本空间中有多少非二次剩余？

HIT CS&E

## 二次剩余和二次非剩余一样多

令 quad($p$) = { $i^2$ | $i$=1, 2, ..., $p$-2, $p$-1}

只需证明 |quad($p$)| = ($p$-1)/2

只需证明 |quad($p$)| ≤ ($p$-1)/2 且 |quad($p$)| ≥ ($p$-1)/2

$(p-x)^2 = p^2-2px+x^2$ mod $p$

$= x^2$ mod $p$

|quad($p$)| ≤ ($p$-1)/2

$x^2 = y^2$ mod $p$

在数域中至多有两根±$y$

|quad($p$)| ≥ ($p$-1)/2

HIT CS&E

## 抽样算法的性能

抽样算法

输入：素数$p$($p$>2)

输出：$x\in${1,2,…,$p$-1}使得$x$是$p$的非二次剩余

1. While (TRUE) do
2. 从{1,2,…,$p$-1}中均匀随机地抽取一个元素$x$
3. If $x^{(p-1)/2} = -1$ Then return $x$

算法一遍成功的概率恰为1/2

算法平均需要运行两遍才结束

HIT CS&E

## 6.2.2 水库抽样

HIT CS&E

## 均匀抽样问题

均匀抽样问题

输入：$N$个对象

输出：从输入的$N$个对象中均匀地抽取$n$个对象

- $N$可以是已知的 （对数据库中的对象进行抽样）
- $N$也可以是未知的（对股票交易进行抽样）
- $n$可能受限于存储空间或其他预算
- 要求仅扫描数据一遍

HIT CS&E

## 选择抽样算法

选择抽样算法

输入：$N$个对象($N$已知)

输出：从输入的$N$个对象中均匀地抽取$n$个对象

1.$m$=0;

2.For $i$=1 To $N$ Do

3.  $O_i$被以概率 $(n-m)/(N-i+1)$的概率保存为样本

4.  If $O_i$被选为样本 Then $m = m+1$

5.输出选中的所有样本

Pr[$O_1$被选] = $n/N$

Pr[$O_2$被选] = Pr[$O_2$被选|$O_1$被选]·Pr[$O_1$被选]

+ Pr[$O_2$被选|$O_1$未被选]·Pr[$O_1$未被选]

= $n/N$

归纳证明，产生规模为$n$的均匀样本

HIT CS&E

## 水库抽样算法

水库抽样(Reservoir Sampling)

输入：$N$个对象($N$未知)

输出：从输入的$N$个对象中均匀地抽取$n$个对象

1.创建数组R[0:$n$-1];  //水库

2.For $i$=1 To $n$ Do

3.  $R[i]=O_i$  //初始化水库

4.For each $O_i$ Do  // $i>n$

5.  以概率$n/i$用$O_i$替换$R[0:n-1]$中均匀随机位置上的对象

当$i=n$时

Pr[$O_1$被选] = $n/n$ = 1

…

Pr[$O_n$被选] = $n/n$ = 1

假设$i$时

Pr[$O_1$被选] = $n/i$

…

Pr[$O_i$被选] = $n/i$

HIT CS&E

在$i$+1时

Pr[$O_{i+1}$被选] = $n/(i+1)$

对于$k<i+1$

Pr[$O_k$被选] = Pr[O$_k$在之前的样本中且未被替换]

= Pr[O$_k$在之前的样本中且$O_{i+1}$未被选为样本]

+Pr[$O_k$在之前的样本中且$O_{i+1}$选中后未替换$O_k$]

$$= \frac{n}{i}\left(1-\frac{n}{i+1}\right)+\frac{n}{i}\frac{n}{i+1}\left(1-\frac{1}{n}\right)$$

$$= \frac{n}{i+1}$$

由归纳法原理可知，

$N$个对象之后，每个对象被选为样本的概率均为$n/N$

HIT CS&E

# 6.3 蒙特卡罗方法

- 6.3.1 蒙特卡罗方法概述
- 6.3.2 DNF的满足性赋值计数
- 6.3.3 从随机抽样到随机计数
- 6.3.4 马尔科夫链蒙特卡罗方法

HIT CS&E

## 6.3.1 蒙特卡罗方法概述

- what?
- why?
- how?

HIT CS&E

## Monte Carlo方法

如何找出最大的苹果？

方法1：依次称量每个苹果

准确可靠

工作量大，效率低

方法2：先取一个苹果

重复如下过程$n$次

再取一个苹果

留大的，放回小的

$n$越大，结论越可靠

HIT CS&E

## Monte Carlo方法

蒙特卡罗方法
- 通过反复抽样完成计算的一大类算法
- 又称随机抽样方法或统计实验方法
- 用计算机实现的快速抽样和统计
- 为反映其概率统计特性，用赌城的名字命名

缺点
- 计算结果存在统计误差
- 方法各要素需要仔细设计才能平衡统计误差和系统误差

选用蒙特卡罗方法的两大决定因素
- 高维问题
  - 将问题分解成低维问题的近似表示时准确性很差
- 复杂结构问题
  - 用蒙特卡罗方法比其他方法更简单

HIT CS&E

## Monte Carlo方法运用过程

第一步：构造或描述概率过程
- 概率过程的数字特征(概率、期望...)与问题的解相关
- 问题本身具有随机性，关键在于描述的准确性
- 问题本身没有随机性，需要人为构造概率过程

第二步：实现从已知概率分布抽样
- 随机数产生算法是最基本的抽样工具
- 抽样质量决定Monte Carlo方法是否有效

第三步：建立统计量作为问题的近似解
- 无偏估计
- 对实验结果进行考察、登记，得出问题的解

HIT CS&E

## Monte Carlo方法应用领域

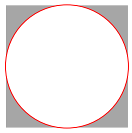
- 物理
  - 热力学运动
  - 原子相互作用
  - 量子运动规律

  ......
- 化学
- 工程
- 金融和风险评估

  ......

HIT CS&E

## π的计算



随机变量$X$表示
单位正方形内随机点是否位于单位圆内

$$X = \begin{cases} 1 & x^2+y^2\le 1 \\ 0 & \text{否则} \end{cases}$$

$\Pr[X=1] = \pi/4$  $E[X]= \pi/4$

$X_1,X_2,\ldots,X_n$是与$X$同分布的独立随机变量  $Y=\sum_{i=1}^{n} X_i$

$\frac{4}{n}\sum_{i=1}^{n} X_i$ 是π的无偏估计

$E[Y]= n\pi/4$

$E[Y/n]= \pi/4$

HIT CS&E

## π的无偏估计计算法

1. $Y$=0
2. For $i$=1 to $n$ Do
3. 均匀一致地产生随机数$x\in[0,1]$, $y\in[0,1]$
4. If $x^2+y^2\le 1$ Then $Y=Y+1$
5. 输出$4Y/n$

问题：$n$取多大才能使$4Y/n$与π的相对误差小于ε?

$|4Y/n-\pi|\le\varepsilon\pi \iff |Y/n - \pi/4|\le\varepsilon\pi/4$

$|Y/n - E[Y/n]|\le\varepsilon\, E[Y/n]$

可以用Chernoff界(第5章作业)

当 $n\ge \frac{12\ln(2/\delta)}{\varepsilon^2\pi}$ 时有 $\Pr[\,|\,4Y/n -\pi\,| \le \varepsilon\pi\,] \ge 1-\delta$

HIT CS&E

定理：如果$X_1,X_2,\ldots,X_n$是独立同分布的示性变量，$E[X_i]=\mu$，

则 $n\ge \frac{3\ln(2/\delta)}{\varepsilon^2\mu}$ 时有

$$\Pr\left[\,\left|\,\frac{1}{n}\sum_{i=1}^{n} X_i -\mu\,\right| \le \varepsilon\mu\,\right] \ge 1-\delta$$

如上例，利用定理 容易建立样本数和近似程度之间的关系

如果随机算法的输出值$X$与问题的解$V$满足

$\Pr[\,|X-V|\le\varepsilon V] \ge 1-\delta$

则称该随机算法是一个(ε,δ)-近似

如果随机算法对任意的ε>0, 0<δ<1能够在1/ε, $\ln\delta^{-1}$和问题输入规模的多项式时间内给出问题解$V$的(ε,δ)-近似，则称该随机算法是该问题的完全多项式随机近似方案(FPRAS)

HIT CS&E

# 6.3.2 DNF满足性赋值的近似计数

- 问题定义
- 朴素算法
- Ruboly-Karp算法

HIT CS&E

## DNF满足性赋值计数问题

DNF满足性赋值计数问题

输入：文字$x_1,x_2,\ldots,x_n$及其上的$m$个合取子句$C_1,\ldots,C_m$,

输出：能使$C_1,\ldots,C_m$之一被满足的$x_1,\ldots,x_n$赋值的个数

例　输入：$\neg x_1\wedge x_3$, $x_2\wedge\neg x_3$，$x_1\wedge x_2$，$\neg x_2\wedge\neg x_3$

输出：7

| | |
|---|---|
| TTT 满足$C_3$ | FTT 满足$C_1$ |
| TTF 满足$C_2$ | FTF 满足$C_2$ |
| TFT 不满足$C_i$ | FFT 满足$C_1$ |
| TFF 满足$C_4$ | FFF 满足$C_4$ |

HIT CS&E

## DNF满足性赋值计数问题的难度

CNF公式与DNF公式之间的对应关系

CNF公式的各个析取子句$C_1,\ldots,C_m$　存在满足性赋值

依次取否　⇕

DNF公式的各个合取子句$D_1,\ldots,D_m$　满足性赋值少于$2^n$

例　$x_1\vee\neg x_3$, $\neg x_2\vee x_3$，$\neg x_1\vee\neg x_2$，$x_2\vee x_3$　无法满足所有子句

各子句依次取否　同一赋值

$\neg x_1\wedge x_3$, $x_2\wedge\neg x_3$，$x_1\wedge x_2$，$\neg x_2\wedge\neg x_3$　满足一个子句

CNF公式的满足性问题是NP难的

DNF公式的满足性赋值计数问题也是难解的　　#P难

HIT CS&E

## 朴素算法

DNF满足性赋值计数问题

输入：文字$x_1,x_2,\ldots,x_n$上DNF公式 $F=C_1\vee\ldots\vee C_m$

输出：能使$C_1,\ldots,C_m$之一被满足的$x_1,\ldots,x_n$赋值的个数$c(F)$

1. $X$=0
2. For $k$=1 To $N$ Do
3. 　从$x_1,\ldots,x_n$的$2^n$种可能赋值中均匀随机地抽取一个赋值
4. 　IF 所取赋值满足$C_1,\ldots,C_m$中某个子句 Then　$X=X+1$
5. 返回 $Y=(X/N)2^n$

算法实质：(1)用蒙特卡罗方法得到近似概率$X/N\approx c(F)/2^n$

(2)用近似概率乘以样本空间大小得到近似计数

HIT CS&E

## 算法分析

$$X_i=\begin{cases}1 & \text{第}i\text{次抽取的随机样本满足}C_1,\ldots,C_m\text{之一}\\ 0 & \text{否则}\end{cases}$$

$X_1,\ldots,X_N$是独立同分布的两点分布　　$X=\sum_{i=1}^{N} X_i$

$\Pr[X_i=1] = c(F)/2^n$　　$E[X_i] = c(F)/2^n$　　$E[X/N]= c(F)/2^n$

由Chernoff界可知

$\Pr[\ |X/N- c(F)/2^n| > \varepsilon\, c(F)/2^n\ ] \le \delta$　　$N\ge 3\cdot 2^n \ln(2/\delta)/\varepsilon^2 c(F)$

$|X/N- c(F)/2^n| > \varepsilon\, c(F)/2^n$　$\Leftrightarrow$　$|Y- c(F)| > \varepsilon\, c(F)$

$\Pr[\ |Y- c(F)| > \varepsilon\, c(F)] \le \delta$　　$N\ge 3\cdot 2^n \ln(2/\delta)/\varepsilon^2 c(F)$

问题：抽样次数可能很大，尤其当$c(F)<<2^n$ 或 $c(F)=O(n^k)$

HIT CS&E

## 改造样本空间

改造样本空间的必要性

- 目标样本在样本空间内非常稀疏
- 需要很多次的抽样才能找到一个目标样本
- 在得到$(\varepsilon,\delta)$近似需要海量的抽样次数

改造样本空间的方法

- 找到样本空间的一个子空间，其大小易于计算
- 目标样本在子空间内稠密
- 实现子空间内的均匀抽样或根据已知分布抽样
- 建立$(\varepsilon,\delta)$近似

HIT CS&E

## 重审DNF计数问题

DNF满足性赋值计数问题

输入：文字$x_1,x_2,\ldots,x_n$及其上的$m$个合取子句$C_1,\ldots,C_m$,

输出：能使$C_1,\ldots,C_m$之一被满足的$x_1,\ldots,x_n$赋值的个数

例 输入：$\underbrace{\neg x_1\wedge x_3}_{C_1}$, $\underbrace{x_2\wedge\neg x_3}_{C_2}$, $\underbrace{x_1\wedge x_2}_{C_3}$, $\underbrace{\neg x_2\wedge\neg x_3}_{C_4}$

输出：7

仅有F?T形式的赋值才可能满足 $C_1$ 共2个

仅有?TF形式的赋值才可能满足 $C_2$ 共2个

仅有TT?形式的赋值才可能满足 $C_3$ 共2个

仅有?FF形式的赋值才可能满足 $C_4$ 共2个

共8个，重复1个
从8个中抽样，
统计不重复率
7= 8* (7/8)

HIT CS&E

## 重构样本空间

DNF满足性赋值计数问题

输入：文字$x_1,x_2,\ldots,x_n$上DNF公式F=$C_1\vee\ldots\vee C_m$,

输出：能使$C_1,\ldots,C_m$之一被满足的$x_1,\ldots,x_n$赋值的个数$c(F)$

对于第$i$个子句$C_i$，令$SC_i=\{a\mid$ 赋值$a$满足$C_i\}$ $|SC_i|=2^{n-|C_i|}$

$c(F)=|SC_1\cup SC_2\cup\ldots\cup SC_m|$

$U=\{(i,a)\mid 1\le i\le m, a\in SC_i\}$

$|U|=|SC_1|+|SC_2|+\ldots+|SC_m|$

$c(F)=\dfrac{c(F)}{|U|}\cdot|U|$

用蒙特卡罗模拟得

从$U$中均匀随机抽样$(i,a)$, $a$一定满足$C_i$

如果 $a$还满足$C_k(k<i)$,则$a$是重复的

HIT CS&E

## $U$上的均匀抽样

对于第$i$个子句$C_i$，令$SC_i=\{a\mid$ 赋值$a$满足$C_i\}$ $|SC_i|=2^{n-|C_i|}$

$U=\{(i,a)\mid 1\le i\le m, a\in SC_i\}$ $|U|=|SC_1|+|SC_2|+\ldots+|SC_m|$

以概率$|SC_i|/|U|$从$\{1,2,\ldots,m\}$中抽取$i$

均匀随机地从$SC_i$中抽取$a$ 固定$C_i$文字的值，其余随机赋值

Pr[取中$(i,a)$] = Pr[取中$a$|取中$i$]· Pr[取中$i$]

= $(1/|SC_i|)\cdot(|SC_i|/|U|)$

= $1/|U|$

HIT CS&E

## Buboly-Karp算法

DNF满足性赋值计数问题

输入：文字$x_1,x_2,\ldots,x_n$上DNF公式 $F=C_1\vee\ldots\vee C_m$

输出：能使$C_1,\ldots,C_m$之一被满足的$x_1,\ldots,x_n$赋值的个数$c(F)$

1. $X=0$
2. For $k=1$ To $N$ Do
3. 以概率$|SC_i|/|U|$从$\{1,2,3,\ldots,m\}$中抽取$i$
4. 从$SC_i$中均匀随机抽取$a$
5. If 不存在$j<i$使得$a$满足$C_j$ Then $X=X+1$
5. 返回 $Y=(X/N)|U|$

HIT CS&E

## 算法分析

$$X_i=\begin{cases}1 & \text{第}i\text{次抽取的随机样本仅满足}C_1,\ldots,C_m\text{之一}\\ 0 & \text{否则}\end{cases}$$

$X_1,\ldots,X_N$是独立同分布的两点分布 $X=\sum_{i=1}^{N}X_i$

$\Pr[X_i=1]=c(F)/|U|$ $E[X_i]=c(F)/|U|$ $E[X/N]=c(F)/|U|$

由Chernoff界可知

$\Pr[\,|X/N-c(F)/|U||>\varepsilon\, c(F)/|U|\,]\le\delta$ $N\ge 3\cdot|U|\ln(2/\delta)/\varepsilon^2 c(F)$

$|X/N-c(F)/|U||>\varepsilon\, c(F)/|U|$ $\Leftrightarrow$ $|Y-c(F)|>\varepsilon\, c(F)$

$\Pr[\,|Y-c(F)|>\varepsilon\, c(F)]\le\delta$ $N\ge 3\cdot|U|\ln(2/\delta)/\varepsilon^2 c(F)$

$N\ge 3\cdot|U|\ln(2/\delta)/\varepsilon^2$

HIT CS&E

## 6.3.3 从近似抽样到近似计数

- 抽样的近似性
- 近似抽样的可用性示例

HIT CS&E

## 抽样的近似性

抽样的近似性

- 依据某一分布从样本空间Ω抽样
- 抽样结果一般而言并不恰好是目标分布
- 抽样结果的分布与目标分布之间往往存在误差
- 这种误差对建立抽样和计数之间的联系是可容忍的

本小节讨论论证这种容忍度的一般框架

- 实例论证
- 由独立集的近似均匀抽样实现对独立集的近似计数
- 建立抽样的近似程度与统计结果准确度之间的联系

HIT CS&E

## 完全多项式时间几乎均匀抽样

定义：将样本空间Ω上抽样算法的一个输出样本记为$w$，如果

$$\left| \Pr[w \in S] - \frac{|S|}{|\Omega|} \right| \le \varepsilon$$

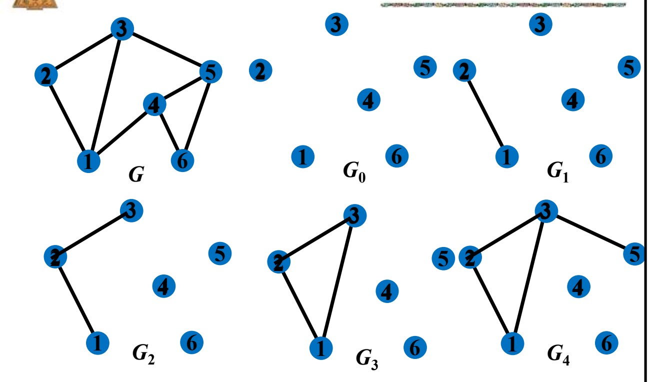
对任意子空间S⊆Ω成立，则称抽样算法是一个ε-均匀抽样

如果对输入$x$和ε，抽样算法在$\ln\varepsilon^{-1}$和$|x|$的多项式时间内输出Ω($x$)的一个ε-均匀抽样，则称该抽样算法是一个完全多项式时间几乎均匀抽样算法(FPAUS)

例如：输入图$G$和ε，FPAUS能够在多项式时间内返回一个独立集样本，该样本与均匀分布的误差不超过ε

HIT CS&E

## 图的序列化表示




HIT CS&E

## 图的序列化表示

图的序列化表示

- 给定图$G=(V,E)$, $|E|=m$
- 将$E$中所有边任意排定顺序$e_1,\ldots,e_m$
- $G_0=(V,\varnothing)$
- $G_1=(V,\{e_1\})$
- …
- $G_i=(V,\{e_1,\ldots,e_i\})$
- …
- $G_m=G$

Ω($G_i$)表示$G_i$的所有独立集构成的集合，则

$$|\Omega(G)| = \frac{|\Omega(G_m)|}{|\Omega(G_{m-1})|} \frac{|\Omega(G_{m-1})|}{|\Omega(G_{m-2})|} \cdots \frac{|\Omega(G_1)|}{|\Omega(G_0)|} |\Omega(G_0)|$$

$$|\Omega(G_0)| = 2^{|V|}$$

HIT CS&E

## 独立集总数估计蒙特卡罗方法

$$|\Omega(G)| = \frac{|\Omega(G_m)|}{|\Omega(G_{m-1})|} \frac{|\Omega(G_{m-1})|}{|\Omega(G_{m-2})|} \cdots \frac{|\Omega(G_1)|}{|\Omega(G_0)|} |\Omega(G_0)|$$

$$|\Omega(G_0)| = 2^{|V|}$$

令 $r_i = \dfrac{|\Omega(G_i)|}{|\Omega(G_{i-1})|}$　　$i=1,2,3,\ldots,m$

$\tilde{r}_i = r_i$的抽样估计值　　$i=1,2,3,\ldots,m$

$|\Omega(G)| = 2^{|V|} r_1 r_2 \ldots r_m$　　$|\Omega(G)|$ 估计值$=2^{|V|}\tilde{r}_1 \tilde{r}_2 \cdots \tilde{r}_m$

$$R = \frac{\tilde{r}_1}{r_1} \frac{\tilde{r}_2}{r_2} \cdots \frac{\tilde{r}_m}{r_m}$$

若$\Pr[\,|R-1| \le \varepsilon] \ge 1-\delta$, 则所给估计是$|\Omega(G)|$ 的(ε,δ)估计

HIT CS&E

## 估计方案的合理性

引理：$r_i \ge 1/2$ (亦即 Ω($G_i$)在Ω($G_{i-1}$)中是稠密的)

证明：$G_i$比$G_{i-1}$多一条边$e_i$，设$e_i$的端点为$u$和$v$

$\Omega(G_i) \subseteq \Omega(G_{i-1})$

$\forall I \in \Omega(G_{i-1}) \setminus \Omega(G_i)$, 必有 $u \in I$ 且$v \in I$

$$\forall I \in \Omega(G_{i-1}) \setminus \Omega(G_i) \xrightarrow{\text{单射}} I \setminus \{v\} \in \Omega(G_i)$$

$$|\Omega(G_{i-1}) \setminus \Omega(G_i)| \le |\Omega(G_i)|$$

$$r_i = \frac{|\Omega(G_i)|}{|\Omega(G_{i-1})|} = \frac{|\Omega(G_i)|}{|\Omega(G_i)| + |\Omega(G_{i-1}) \setminus \Omega(G_i)|} \ge 1/2$$

## 误差累积

引理：如果 $\tilde{r}_i$是$r_i$的$(\varepsilon/2m,\delta/m)$近似, $i=1,2,\ldots,m$, 则
$\Pr[\,|R-1| \le \varepsilon] \ge 1-\delta$

证明：$\Pr[\,|\tilde{r}_i-r_i| \le r_i\cdot\varepsilon/2m] \ge 1-\delta/m \qquad i=1,2,\ldots,m$

$\Pr[\,|\tilde{r}_i-r_i| > r_i\cdot\varepsilon/2m] < \delta/m \qquad i=1,2,\ldots,m$

由Union Bound知，$\exists i$使$|\tilde{r}_i-r_i| > r_i\cdot\varepsilon/2m$ 的概率$\le\delta$

$|\tilde{r}_i-r_i|\le r_i\cdot\varepsilon/2m$ 对所有$i$成立的概率$\ge 1-\delta$

$$1-\frac{\varepsilon}{2m} \le \frac{\tilde{r}_i}{r_i} \le 1+\frac{\varepsilon}{2m}$$ 对所有$i$成立的概率$\ge 1-\delta$

$$1-\varepsilon \le \left(1-\frac{\varepsilon}{2m}\right)^m \le R \le \left(1+\frac{\varepsilon}{2m}\right)^m \le 1+\varepsilon$$ 成立的概率$\ge 1-\delta$

## $r_i$的$(\varepsilon/2m,\delta/m)$近似

前面的分析表明

- 欲得$|\Omega(G)|$的$(\varepsilon/,\delta)$近似，需要$r_i$的$(\varepsilon/2m,\delta/m)$近似
- 若存在独立集的FPAUS，则可以建立所需的近似

$r_i$的$(\varepsilon/2m,\delta/m)$近似估计算法Estimate

输入：$G_{i-1}=(V,\{e_1,\ldots,e_{i-1}\})$和$G_i=(V,\{e_1,\ldots,e_i\})$

输出：$r_i$的$(\varepsilon/2m,\delta/m)$近似估计

1. $X\leftarrow 0$
2. For $i=1$ To $M=\lceil 1296m^2\varepsilon^{-2}\ln(2m/\delta)\rceil$ Do
3. 独立地调用FPAUS获取$|\Omega(G_{i-1})|$的一个$\varepsilon/6m$-均匀样本
4. 如果样本也是$G_i$的独立集，则$X\leftarrow X+1$
5. 输出$X/M$

---

引理：算法Estimate得到$r_i$的一个$(\varepsilon/2m,\delta/m)$近似

证明：$$X_k=\begin{cases}1 & \text{从}\Omega(G_{i-1})\text{抽取的第}k\text{个在}\Omega(G_i)\text{中}\\ 0 & \text{否则}\end{cases}$$

第3步的FPAUS得到$\varepsilon/6m$-均匀样本

$$\left|\Pr[X_k=1]-\frac{|\Omega(G_i)|}{|\Omega(G_{i-1})|}\right| \le \varepsilon/6m \;\Rightarrow\; \left|E[X_k]-\frac{|\Omega(G_i)|}{|\Omega(G_{i-1})|}\right| \le \varepsilon/6m$$

$$\Rightarrow \left|E\left(\frac{\sum_{k=1}^{M} X_k}{M}\right)-\frac{|\Omega(G_i)|}{|\Omega(G_{i-1})|}\right| \le \varepsilon/6m$$ 期望的线性性质

---

$$|E[\tilde{r}_i]-r_i| = \left|E\left(\frac{\sum_{k=1}^{M} X_k}{M}\right)-\frac{|\Omega(G_i)|}{|\Omega(G_{i-1})|}\right| \le \varepsilon/6m$$

$E[\tilde{r}_i] \ge r_i-\varepsilon/6m \ge 1/2-\varepsilon/6m \ge 1/3 \qquad r_i\ge 1/2$

$$M \ge \frac{3\ln(2m/\delta)}{(\varepsilon/12m)^2(1/3)} = 1296m^2\varepsilon^{-2}\ln(2m/\delta)$$ 后

$$\Pr\left(\left|\frac{\tilde{r}_i}{E[\tilde{r}_i]}-1\right| \ge \varepsilon/12m\right) = \Pr\left[\,|\tilde{r}_i-E[\tilde{r}_i]| \ge \frac{\varepsilon}{12m}\tilde{r}_i\right] \le \delta/m$$

$$1-\frac{\varepsilon}{12m} \le \frac{\tilde{r}_i}{E[\tilde{r}_i]} \le 1+\frac{\varepsilon}{12m}$$ 成立的概率$\ge 1-\delta/m$

将$|E[\tilde{r}_i]-r_i| \le \varepsilon/6m$带入上式,得

---

$$1-\frac{\varepsilon}{6mr_i} \le \frac{E[\tilde{r}_i]}{r_i} \le 1+\frac{\varepsilon}{6mr_i}$$

又由于$r_i \ge 1/2$, 所以

$$1-\frac{\varepsilon}{3m} \le \frac{E[\tilde{r}_i]}{r_i} \le 1+\frac{\varepsilon}{3m}$$

$$1-\frac{\varepsilon}{12m} \le \frac{\tilde{r}_i}{E[\tilde{r}_i]} \le 1+\frac{\varepsilon}{12m}$$ 成立的概率$\ge 1-\delta/m$

$$\left(1-\frac{\varepsilon}{2m}\right) \le \left(1-\frac{\varepsilon}{3m}\right)\left(1-\frac{\varepsilon}{12m}\right) \le \frac{\tilde{r}_i}{r_i} \le \left(1+\frac{\varepsilon}{3m}\right)\left(1+\frac{\varepsilon}{12m}\right) \le \left(1+\frac{\varepsilon}{2m}\right)$$

亦即，算法Estimate得到$r_i$的一个$(\varepsilon/2m,\delta/m)$近似

---

## 6.3.4 马尔科夫链蒙特卡罗方法

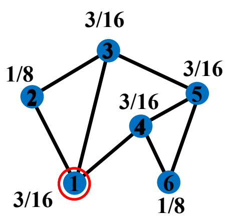
- 马尔科夫链蒙特卡罗方法
- Metropolis方法

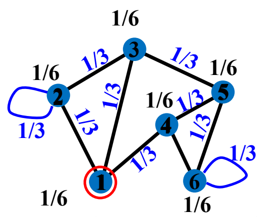
HIT CS&E

## 马尔科夫链蒙特卡罗方法

随机游走的稳定分布是$\pi_v = d(v)/|E|$

问题：能利用它产生顶点集上的均匀分布吗？




结论：在图上恰当地添加自环，随机游走可产生均匀分布

HIT CS&E

## 马尔科夫链蒙特卡罗方法

定理：给定有限状态空间Ω和邻域结构$\{N(x)|\ x\in\Omega\}$.
令$N=\max_{x\in\Omega}|N(x)|$, $M\geq N$. 定义马尔科夫链

$$P_{x,y}=\begin{cases}1/M & x\neq y\text{且 } y\in N(x) \quad \text{正常边}\\ 0 & x\neq y\text{且 } y\notin N(x) \quad \text{非边}\\ 1-|N(x)|/M & x=y \quad \text{自环}\end{cases}$$

如果该马尔科夫链是不可约非周期的，则其稳定分布是Ω上的均匀分布

证明：稳定分布π对任意$y\neq x$均有$\pi_x P_{x,y} = \pi_y P_{y,x}$
若$y,x$有边相连则$P_{x,y} = P_{y,x} = 1/M$，于是$\pi_x=\pi_y$
由于该链状态相互可达，故π是均匀分布

HIT CS&E

## 马尔科夫链蒙特卡罗方法

利用马尔科夫链蒙特卡罗方法实现均匀抽样

- 构造马尔科夫链，其状态空间为样本空间Ω
- 对$\forall x\in\Omega$，探究$N(x)$的大小，找出恰当$M$
- 根据定理，定义状态转移概率
- 论证马尔科夫链的不可约性和非周期性
- 该链的状态转移图就是带自环的图
- 寻找合适的$r$，实现抽样得到均匀样本$X_r, X_{2r}, X_{3r},\ldots$

HIT CS&E

## 例

给定无向连通图$G=(V,E)$, 对其独立集实现均匀抽样

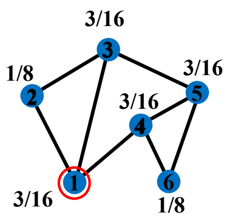
- $X_0$是$G$中任意一个独立集
- $X_{i+1}$如下构造
  - 均匀随机地选取顶点$v\in V$
  - If $v\in X_i$ Then $X_{i+1}= X_i-\{v\}$
  - Else If $X_i\cup\{v\}$仍是独立集 Then $X_{i+1}= X_i\cup\{v\}$
  - Else $X_{i+1}= X_i$
- $N(X)$——与$X$仅相差一个顶点的独立集
- $Y\in N(X)$, $P_{X,Y}= 1/|V|$
- $P_{X,X}= 1- |N(X)|/|V| >0$ 非周期
- 任意状态可以到达空集，也可以由空集达到 不可约
- 合适选取$r$，可得$G$上独立集均匀样本$X_r, X_{2r}, X_{3r},\ldots$

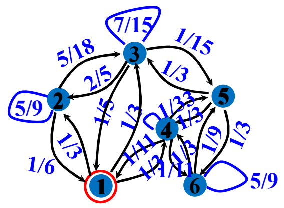
HIT CS&E

## Metropolis算法

随机游走的稳定分布是$\pi_v = d(v)/|E|$

问题：能用它产生$|V|$上分布(1/10, 1/5, 1/6,11/30, 1/30,1/10)?




结论：自环+转移概率分配，随机游走任意分布

HIT CS&E

## Metropolis算法

定理：给定有限状态空间Ω，Ω上的概率分布π和邻域结构$\{N(x)|\ x\in\Omega\}$. 令$N=\max_{x\in\Omega}|N(x)|$, $M\geq N$. 定义马尔科夫链

$$P_{x,y}=\begin{cases}1/M\cdot\min(1, \pi_y/\pi_x) & x\neq y\text{且 } y\in N(x) \quad \text{正常边}\\ 0 & x\neq y\text{且 } y\notin N(x) \quad \text{非边}\\ 1-\sum_{z\in N(x)}P_{x,z} & x=y \quad \text{自环}\end{cases}$$

如果该马尔科夫链是不可约非周期的，则其稳定分布是Ω上的分布π

证明：稳定分布π对任意$y\neq x$均有$\pi_x P_{x,y} = \pi_y P_{y,x}$
若$\pi_x\leq\pi_y$且$y,x$有边相连则$P_{x,y} = 1$, $P_{y,x} = \pi_x/\pi_y$
若$\pi_x>\pi_y$且$y,x$有边相连，$P_{y,x}=1$, $P_{x,y} = \pi_y/\pi_x$

## Metropolis算法

利用Metropolis算法实现目标分布抽样

- 构造马尔科夫链，其状态空间为样本空间Ω
- 对$\forall x\in\Omega$，探究$N(x)$的大小，找出恰当$M$
- 根据定理，定义状态转移概率
- 论证马尔科夫链的不可约性和非周期性
- 该链的状态转移图就是带自环的图
- 寻找合适的$r$，实现抽样得到目标分布样本 $X_r, X_{2r}, X_{3r}, \dots$

## 例

无向连通图$G=(V,E)$ 独立集目标分布

- $\Omega=\{I \mid I$是$G$的独立集$\}$
- 分布参数λ
  - $B=\sum_{I\in\Omega}\lambda^{|I|}$
  - 抽中$I$的概率$\lambda^{|I|}/B$
- λ=1, 目标分布是均匀分布
- λ<1, 小独立集被抽中的概率大，大独立集被抽中的概率小
- λ>1，大独立集被抽中的概率大，小独立集被抽中的概率小

## 例(续)

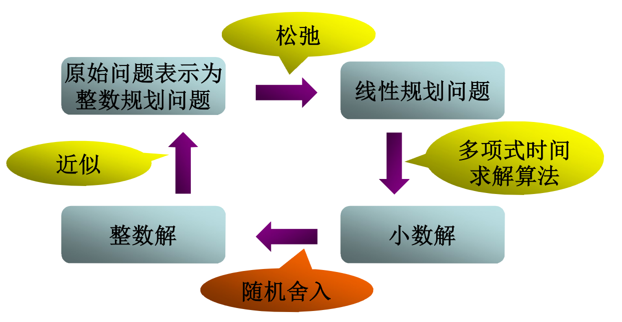
给定无向连通图$G=(V,E)$, 对其独立集实现目标分布抽样

- $X_0$是$G$中任意一个独立集
- $X_{i+1}$如下构造
  - 均匀随机地选取顶点$v\in V$
  - If $v\in X_i$ Then 以概率$\min(1,1/\lambda)$令$X_{i+1}=X_i-\{v\}$
  - Else If $X_i\cup\{v\}$独立 Then 以概率$\min(1,\lambda)$令$X_{i+1}=X_i\cup\{v\}$
  - Else $X_{i+1}=X_i$
- $N(X)$——与$X$仅相差一个顶点的独立集
- $Y\in N(X)$， $P_{X,Y}=1/|V|\min(1,1/\lambda)$ 或 $1/|V|\min(1,\lambda)$
- $P_{X,X}=1-\sum_{I\in N(X)}P_{X,I}>0$　　非周期
- 任意状态可以到达空集，也可以由空集达到　　不可约
- 合适选取$r$，可得$G$上独立集均匀样本 $X_r, X_{2r}, X_{3r}, \dots$

## 6.4 随机舍入

- 随机舍入算法的基本框架
- 顶点覆盖问题的随机舍入算法
- 集合覆盖问题的随机舍入算法

## 随机舍入算法的基本框架



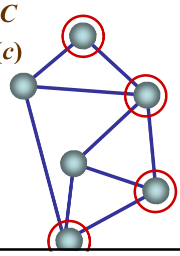

## 顶点覆盖问题

问题的定义

- 输入：无向图$G=(V, E)$, 每个节点具有权$w(v)$.
- 输出：$C\subseteq V$, 满足
  - (1). $\forall(u, v)\in E$, $u\in C$或者$v\in C$
  - (2). $w(C)$最小, $w(C)=\sum_{c\in C}w(c)$

在节点汇结点安装路灯

- 照亮每条街道
- 成本最低
- NP难



HIT CS&E

## 整数规划问题表示

- 问题转化为0-1线性规划问题$P_{0\text{-}1}$
  - 对于$\forall v\in V$, 定义 $x(v)\in\{0, 1\}$如下:
    - 若$v$在节点覆盖中, 则$x(v)=1$, 否则$x(v)=0$
    - $\forall(u, v)\in E$, 若$u$、$v$或两者在覆盖中, 则$x(u)+x(v)\geq 1$
  - 对应的0-1整数规划问题$P_{0\text{-}1}$
    - 优化目标: 最小化 $\sum_{v\in V} w(v)x(v)$
    - 约束条件: $x(u)+x(v)\geq 1$ for $\forall v\in V$

      $x(v)\in\{0, 1\}$ for $\forall v\in V$
  - 0-1整数规划问题也是NP-完全问题

HIT CS&E

## 松弛为线性规划问题

- 用线性规划问题的解近似0-1规划问题的解
  - 对于$\forall v\in V$, 定义 $x(v)\in[0, 1]$
  - $P_{0\text{-}1}$对应的线性规划问题LP
    - 优化目标: min $\sum_{v\in V} w(v)x(v)$
    - 约束条件: $x(u)+x(v)\geq 1$ *for* $\forall v\in V$

      $x(v)\in[0, 1]$ *for* $\forall v\in V$
  - 线性规划问题具有多项式时间算法
  - $P_{0\text{-}1}$的可能解是LP问题的可能解
  - $P_{0\text{-}1}$解的代价≥LP的解的代价

HIT CS&E

## 顶点覆盖问题的舍入算法

### 随机舍入算法

Approx-Min-VC($G$, $w$)
1. $C=\varnothing$;
2. 调用多项式时间算法计算$LP$问题的优化解$x$;
3. For each $v\in V$ Do
4. If $x(v)\geq 1/2$ Then $C=C\cup\{v\}$;

   /* 用四舍五入法把LP的解近似为$P_{0\text{-}1}$的解 */
5. Return $C$.

HIT CS&E

## 近似程度分析

- 算法的性能

  定理. *Approx-Min-VC*是一个多项式时间2-近似算法

  证.

  由于求解LP需多项式时间, Approx-Min-VC的For循环需要多项式时间, 所以算法需要多项式时间.

  下边证明Approx-Min-VC的近似比是2.

  往证算法产生的$C$是一个节点覆盖.

  $\forall(u, v)\in E$, 由约束条件可知 $x(u)+x(v)\geq 1$. 于是, $x(u)$和$x(v)$至少一个大于等于1/2, 即$u$、$v$或两者在$C$中. $C$是一个覆盖.

HIT CS&E

往证$w(C)/w(C^*)\leq 2$.

令$C^*$是$P_{0\text{-}1}$的优化解, $z^*$是LP优化解的代价. 因为$C^*$是LP的可能解, $w(C^*)\geq z^*$.

$$
\begin{aligned}
z^* = \sum_{v\in V} w(v)x(v) &\geq \sum_{v\in V:\, x(v)\geq 1/2} w(v)x(v)\\
&\geq \sum_{v\in V:\, x(v)\geq 1/2} w(v)1/2\\
&= \sum_{v\in C} w(v)1/2\\
&= (1/2)\sum_{v\in C} w(v)\\
&= (1/2)w(C).
\end{aligned}
$$

由$w(C^*)\geq z^*$, $w(C^*)\geq(1/2)w(C)$, 即 $w(C)/w(c^*)\leq 2$.

HIT CS&E

## 集合覆盖问题

- 输入:

  有限集$X$, $X$的一个子集族$F$, $X=\cup_{S\in F} S$, 每个集合$S$的代价$c(S)$
- 输出:

  $C\subseteq F$, 满足

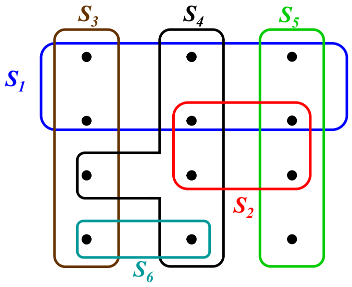
  (1). $X=\cup_{S\in C} S$,

  (2). $C$是满足条件(1)的代价最小的集族, 即$\sum_{S\in C} c(S)$最小.

*最小集合覆盖问题是很多实际问题的抽象.

*最小集合覆盖问题是NP-完全问题.

• 问题的实例




$X$=12个黑点， $\mathbf{S}=\{S_1, S_2, S_3, S_4, S_5, S_6\}$　$c(S_i)=1$

优化解$C=\{S_3, S_4, S_5\}$

## 集合覆盖问题的线性规划表示

对$F$中的每个集合$S$，引入一个变量$x_S$

$x_S=0$ 表示$S\notin C$　　$x_S=1$ 表示$S\in C$

原问题的线性规划表示

$$\begin{aligned} &\text{minimize} && \sum_{S\in F} C(S)\cdot x_S \\ &\text{subject to} && \sum_{S:e\in S} x_S \ge 1 && e\in X \\ & && x_S\in\{0,1\} && S\in F \end{aligned}$$

LP松弛问题

$$\begin{aligned} &\text{maximize} && \sum_{S\in F} C(S)\cdot x_S \\ &\text{subject to} && \sum_{S:e\in S} x_S \ge 1 && e\in X \\ & && x_S\ge 0 && S\in F \end{aligned}$$

$OPT$　原问题

$LP$松弛问题

## 舍入法

频率: 对于$e\in X$，$e$的频率指的是$F$中包含e的集合的个数

$f$: $X$中元素的最大频率

集合覆盖问题的LP-舍入算法

1. 用Karmarkar算法求得LP-松弛问题的最优解$x$
2. For $S\in F$ Do
   IF $x_S\ge 1/f$　THEN　$C=C\cup\{S\}$
3. 输出$C$

定理6. 对于集合覆盖问题，LP-舍入算法的近似比为$f$

证明：

对于任意$e\in X$, 由于$e$至多属于$f$个集合中，为了确保

$$\sum_{e\in S,\, S\in F} x_S \ge 1$$

必有某个$x_S$使得$x_S\ge 1/f$。因此，算法输出的集族中必有一个集合包含了$e$；进而，算法的输出覆盖了$X$。

在舍入过程中，对任意$S\in C$，$x_S$被舍入为1，至多被放大$f$倍。因此

$$OPT_f=\sum_{S\in S} c(S)x_S=\sum_{S\in C} c(S)x_S+\sum_{\text{其他}S} c(S)x_S\ge\sum_{S\in C} c(S)\frac{1}{f}+\sum_{\text{其他}S} c(S)x_S\ge\frac{1}{f}\sum_{S\in C} c(S)$$

从而，$\sum_{S\in C} c(S)\le f\cdot OPT_f\le f\cdot OPT$.

证毕

## 集合覆盖问题的LP-随机舍入算法

LP-随机舍入算法

1. 用Karmarkar算法求解LP-松弛问题得到最优解$x=\langle x_S:S\in F\rangle$
2. $C=\varnothing$
3. For $\forall S\in F$ Do
4. 　独立地产生一个随机数 rand
5. 　IF　rand > 1-$x_S$　THEN　$C=C\cup\{S\}$;
   /* $S$被选入$C$的概率为$x_S$ */
6. 输出$C$

定理. 对于集合覆盖问题的LP-随机舍入算法，$C$的代价的数学期望为$OPT_f$，其中$OPT_f$是LP-松弛问题的最优解的值

证明：$E(cost(C))=\sum_{S\in S} p_r[S被选入C].c(S)$

$=\sum_{S\in S} x_S\,.\,c(S)$

$=OPT_f$　　证毕

定理. 对于集合覆盖问题的LP-随机舍入算法，$\forall a\in X$被$C$覆盖的概率大于1-1/$e$

证明：

设$a$属于$F$的$k$个集合中，将LP-松弛问题中这些集合对应的变量记为$x_1,\ldots,x_k$.

在LP-松弛问题的优化解中，$x_1+\ldots+x_k\ge 1$

$P_r[a\text{未被}C\text{覆盖}] = (1-x_1)(1-x_2)\ldots(1-x_k)$

$\le (1-(x_1+\ldots+x_k)/k)^k$

$\le (1-1/k)^k$

$Pr[a\text{被}C\text{覆盖}] = 1- Pr[a\text{未被}C\text{覆盖}]$

$\ge 1-(1-1/k)^k$

$\ge 1-1/e$

证毕

改进策略：为了得到完整的集合覆盖，独立运行LP-随机舍入算法*clog n*次，其中*c*满足$(\frac{1}{e})^{c\log n} \le \frac{1}{4n}$，将所有输出集合求并得到*C'*，然后输出*C'*。

$P_r[C'\text{未覆盖}X] \le \sum_{a\in X} P_r[C'\text{未覆盖}a] \le n\cdot[(1/e)^{c\log n}] = 1/4$

$E[cost(C')]=OPT_f \cdot c\cdot \log n$

$P_r[cost(C') \ge OPT_f \cdot 4c\log n] \le 1/4$  Markov 不等式：$P_r(X>t) \le \frac{E(X)}{t}$

$P_r[C'\text{覆盖}X \text{ 且 } cost(C') \le OPT_f.4c\log n] = 1- P_r[C'\text{未覆盖}X \text{ 或 } cost(C') \ge OPT_f.4c\log n]$

$\le 1-(1/4+1/4)=1/2$

HIT CS&E

# 6.5 随机抽样与随机舍入混合使用

- 6.5.1 MAX-SAT的随机抽样算法
- 6.5.2 MAX-SAT的随机舍入算法
- 6.5.3 MAX-SAT的混合随机算法

HIT CS&E

## 6.5.1 MAX-SAT的随机抽样算法

HIT CS&E

### 随机抽样算法

MAX-SAT问题的随机抽样算法RandSample

输入：$n$个文字及其上的CNF公式$F=C_1\wedge\ldots\wedge C_m$

输出：文字赋值$x_1,\ldots,x_n$使得$C_1,\ldots,C_m$被同时满足的子句最多

1. For $i$=1 To $n$ Do
2. 第$i$个文字以概率1/2取真,以概率1/2取假
3. 返回1-2步得到的随机赋值

问题：$O(n)$时间内求得的解会不会很差？ 🤔

HIT CS&E

### 算法分析

$Pr[C_j\text{未被满足}] = \underbrace{(½)(½)\ldots(½)}_{|C_j|\text{个}} \le ½$

$Pr[C_j\text{被满足}] = 1- Pr[C_j\text{未被满足}] = 1-1/2^{|C_j|} \ge 1-1/2$

$Y_j = \begin{cases} 1 & \text{子句}C_j\text{被满足} \\ 0 & \text{子句}C_j\text{未被满足} \end{cases}$  $Pr[Y_j=1] \ge 1/2$, $E[Y_j] \ge 1/2$

$Y = Y_1+Y_2+\ldots+Y_m$ 表示被满足的子句的总数

$E[Y] = E[Y_1]+E[Y_2]+\ldots+E[Y_m] \ge m/2$

$$\frac{opt}{E[Y]} = \frac{\text{优化解中被满足的子句个数}}{E[Y]} \le \frac{m}{E[Y]} \le 2$$

结论：RandSample算法是一个多项式时间E[2]-近似算法

HIT CS&E

## 6.5.2 MAX-SAT的随机舍入算法

HIT CS&E

## 问题转化

**MAX-SAT问题**

输入：$n$个文字及其上的CNF公式$F=C_1\wedge\ldots\wedge C_m$

输出：文字赋值$x_1,\ldots,x_n$使得$C_1,\ldots,C_m$被同时满足的子句最多

$$x_i=\begin{cases}1 & 第i个文字取真\\ 0 & 第i个文字取假\end{cases}\qquad y_j=\begin{cases}1 & 子句C_j被满足\\ 0 & 子句C_j未被满足\end{cases}$$

**MAX-SAT表示为0-1规划**

$$\max\ y_1+y_2+\ldots+y_m$$

$$\begin{aligned} s.t.\quad & \Sigma_{i\in C_j^+}x_i+\Sigma_{i\in C_j^-}(1\text{-}x_i)\ge y_j && 1\le j\le m\\ & x_i\in\{0,1\} && 1\le i\le n\\ & y_j\in\{0,1\} && 1\le j\le m\end{aligned}$$

HIT CS&E

## 问题转化

**MAX-SAT问题**

输入：$n$个文字及其上的CNF公式$F=C_1\wedge\ldots\wedge C_m$

输出：文字赋值$x_1,\ldots,x_n$使得$C_1,\ldots,C_m$被同时满足的子句最多

$$x_i=\begin{cases}1 & 第i个文字取真\\ 0 & 第i个文字取假\end{cases}\qquad y_j=\begin{cases}1 & 子句C_j被满足\\ 0 & 子句C_j未被满足\end{cases}$$

**松弛0-1规划中的约束条件得线性规划问题**

$$\max\ y_1+y_2+\ldots+y_m$$

$$\begin{aligned} s.t.\quad & \Sigma_{i\in C_j^+}x_i+\Sigma_{i\in C_j^-}(1\text{-}x_i)\ge y_j && 1\le j\le m\\ & x_i\in[0,1] && 1\le i\le n\\ & y_j\in[0,1] && 1\le j\le m\end{aligned}$$

HIT CS&E

## 随机舍入算法

**MAX-SAT问题的随机舍入算法RandRound**

输入：$n$个文字及其上的CNF公式$F=C_1\wedge\ldots\wedge C_m$

输出：文字赋值$x_1,\ldots,x_n$使得$C_1,\ldots,C_m$被同时满足的子句最多

1. 将MAX-SAT表示为0-1规划问题并松弛为线性规划问题
2. 调用多项式时间算法求得线性规划问题的解$(x^*,y^*)$
   // $x^*=(x_1^*,\ldots,x_n^*)$，每个分量对应一个文字
   // $y^*=(y_1^*,\ldots,y_m^*)$，每个分量对应一个子句
   //能同时被满足的子句的个数不超过$y_1^*+\ldots+y_m^*$
3. For $i$=1 To $n$ Do
4. 第$i$个文字以概率$x_i^*$取真,以概率$1\text{-}x_i^*$取假
5. 返回3-4步得到的赋值

HIT CS&E

## 算法分析

引理：$\Pr[C_j被满足]\ge(1\text{-}1/e)y_j^*$ (待证)

$$Y_j=\begin{cases}1 & 子句C_j被满足\\ 0 & 子句C_j未被满足\end{cases}\qquad \begin{aligned}\Pr[Y_j{=}1]&\ge(1\text{-}1/e)y_j^*\\ E[Y_j]&\ge(1\text{-}1/e)y_j^*\end{aligned}$$

$Y=Y_1+Y_2+\ldots+Y_m$ 表示被满足的子句的总数

$$\begin{aligned}E[Y]&=E[Y_1]+E[Y_2]+\ldots+E[Y_m]\\ &=(1\text{-}1/e)(y_1^*+y_2^*+\ldots+y_m^*)\end{aligned}$$

$$\frac{opt}{E[Y]}=\frac{优化解中被满足的子句个数}{E[Y]}\le\frac{y_1^*+\ldots+y_m^*}{E[Y]}\le\frac{e}{e\text{-}1}$$

结论：RandRound算法是一个多项式时间$E[e/(e\text{-}1)]$-近似算法

HIT CS&E

## 引理证明

$$(a_1\cdot a_2\cdot\ldots\cdot a_k)^{1/k}\le\frac{a_1+a_2+\ldots+a_k}{k}$$

$$\begin{aligned}\Pr[C_j未被满足]&=\Pi_{i\in C_j^+}(1\text{-}x_i^*)\,\Pi_{i\in C_j^-}x_i^*\\ &\le\left(\frac{\Sigma_{i\in C_j^+}(1\text{-}x_i^*)+\Sigma_{i\in C_j^-}x_i^*}{|C_j|}\right)^{|C_j|}\\ &=\left(1\text{-}\frac{\Sigma_{i\in C_j^+}x_i^*+\Sigma_{i\in C_j^-}(1\text{-}x_i^*)}{|C_j|}\right)^{|C_j|}\\ &\le\left(1\text{-}\frac{y_j^*}{|C_j|}\right)^{|C_j|}\end{aligned}$$

$$\Pr[C_j被满足]\ge 1\text{-}\left(1\text{-}\frac{y_j^*}{|C_j|}\right)^{|C_j|}$$

HIT CS&E

论断：$1\text{-}(1\text{-}r/k)^k\ge[1\text{-}(1\text{-}1/k)^k]\cdot r$ 对任意$0\le r\le1$和整数$k$成立

证明：$k$=1,2时，直接验证

当$k$>2时，令 $f(r)=1\text{-}(1\text{-}r/k)^k$

$f(0)=0$

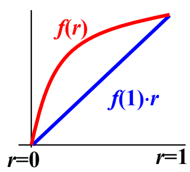
$f(1)=1\text{-}(1\text{-}1/k)^k$

$f'(r)=(1\text{-}r/k)^{k\text{-}1}\ \ge0$

$f''(r)=-\frac{k\text{-}1}{k}(1\text{-}r/k)^{k\text{-}2}\ \le0$

左端= 凸函数

右端=$f(1)\cdot r$是线性函数



HIT CS&E

$$\Pr[C_j\text{被满足}] \ge 1-\left(1-\frac{y_j^*}{|C_j|}\right)^{|C_j|}$$
$$\ge \left[1-\left(1-\frac{1}{|C_j|}\right)^{|C_j|}\right] y_j^*$$
$$\ge (1-1/e)y_j^*$$

HIT CS&E

## 6.5.3 MAX-SAT的混合随机算法

HIT CS&E

## 混合算法

MAX-SAT问题的混合随机算法RandMix

输入：$n$个文字及其上的CNF公式$F=C_1\wedge\ldots\wedge C_m$

输出：文字赋值$x_1,\ldots,x_n$使得$C_1,\ldots,C_m$被同时满足的子句最多

1. 调用RandSample得赋值$A$
   //$C_1,\ldots,C_m$中被$A$满足的子句个数记为$X$
2. 调用RandRound的赋值$B$
   //$C_1,\ldots,C_m$中被$B$满足的子句个数记为$Y$
3. If $X>Y$ Then $C=A$
4. Else $C=B$
5. 返回赋值$C$
   //$C_1,\ldots,C_m$中被$C$满足的子句个数记为Z, $Z\ge(X+Y)/2$

HIT CS&E

## 算法分析

$$Y_j=\begin{cases}1 & \text{子句}C_j\text{被赋值}B\text{满足}\\ 0 & \text{子句}C_j\text{未被赋值}B\text{满足}\end{cases}\qquad X_j=\begin{cases}1 & \text{子句}C_j\text{被赋值}A\text{满足}\\ 0 & \text{子句}C_j\text{未被}A\text{满足}\end{cases}$$

$$\Pr[Y_j=1] \ge \left[1-\left(1-\frac{1}{|C_j|}\right)^{|C_j|}\right] y_j^* \qquad \Pr[X_j=1] = 1-\frac{1}{2^{|C_j|}} \ge \left(1-\frac{1}{2^{|C_j|}}\right)\cdot y_j^*$$

$$E[Y_j] \ge \left[1-\left(1-\frac{1}{|C_j|}\right)^{|C_j|}\right] y_j^* \qquad E[X_j] \ge \left(1-\frac{1}{2^{|C_j|}}\right)\cdot y_j^*$$

$$2E[Z] \ge E[X]+E[Y]$$
$$=E[X_1]+\ldots+E[X_m]+E[Y_1]+\ldots+E[Y_m]$$
$$=(E[X_1]+E[Y_1])+\ldots+(E[X_m]+E[Y_m])$$

HIT CS&E

$$\ge \Sigma_j\left(\left(1-\frac{1}{2^{|C_j|}}\right)\cdot y_j^* + \left[1-\left(1-\frac{1}{|C_j|}\right)^{|C_j|}\right] y_j^*\right)$$
$$\ge \Sigma_j\left(\underbrace{\left(1-\frac{1}{2^{|C_j|}}\right) + \left[1-\left(1-\frac{1}{|C_j|}\right)^{|C_j|}\right]}_{\ge 3/2\text{恒成立}}\right)\cdot y_j^*$$
$$\ge \frac{3}{2}(y_1^*+\ldots+y_m^*)$$

由$2E[Z] \ge (3/2)(y_1^*+\ldots+y_m^*)$可得$E[Z] \ge (3/4)(y_1^*+\ldots+y_m^*)$

$$\frac{opt}{E[Z]} = \frac{\text{优化解中被满足的子句个数}}{E[Z]} \le \frac{y_1^*+\ldots+y_m^*}{E[Z]} \le \frac{4}{3}$$

结论：RandMix算法是一个多项式时间E[4/3]-近似算法