

- 教材讨论
 - JH第5章第3节第1、2、3小节

问题1: random sampling

- 什么是random sampling?
- 什么样的问题适合采用random sampling?
- 为什么quadratic residues适合采用random sampling?

问题1: random sampling (续)

- (i) there are many objects with the given property relative to the cardinality of the set of all objects considered,
- (ii) for a given object, one can efficiently verify whether it has the required property or not, and
- (iii) the distribution of the “right” objects among all objects is unknown and cannot be efficiently computed (or at least one does not know how to determine it efficiently).

- (A) For a given prime p and an $a \in \mathbb{Z}_p$, it is possible to decide whether a is a quadratic residue (mod p) in polynomial time.
- (B) For every prime p , exactly half of the elements of \mathbb{Z}_p are quadratic residues.

Theorem 5.3.2.2 (Euler’s Criterion). *For every $a \in \mathbb{Z}_p$,*

- (i) if a is a quadratic residue modulo p , then $a^{(p-1)/2} \equiv 1 \pmod{p}$, and*
- (ii) if a is a quadratic nonresidue modulo p , then $a^{(p-1)/2} \equiv -1 \pmod{p}$.*

Theorem 5.3.2.3. *For every odd prime p , exactly half³⁹ of the nonzero elements of \mathbb{Z}_p are quadratic residues modulo p .*

问题2: quadratic residues

Theorem 5.3.2.2 (Euler's Criterion). *For every $a \in \mathbb{Z}_p$,*

- (i) if a is a quadratic residue modulo p , then $a^{(p-1)/2} \equiv 1 \pmod{p}$, and*
- (ii) if a is a quadratic nonresidue modulo p , then $a^{(p-1)/2} \equiv -1 \pmod{p}$.*

- 判定quadratic residue的复杂度是多少？为什么？

Theorem 5.3.2.3. *For every odd prime p , exactly half³⁹ of the nonzero elements of \mathbb{Z}_p are quadratic residues modulo p .*

- 这个定理的证明是如何分解为两个步骤并分别证明的？

问题2: quadratic residues (续)

Algorithm 5.3.2.1. REPEATED SQUARING

Input: Positive integers a, b, p , where $b = \text{Number}(b_k b_{k-1} \dots b_0)$.

Step 1: $C := a; D := 1$.

Step 2: **for** $I := 0$ **to** k **do**
 begin **if** $b_I = 1$ **then** $D := D \cdot C \bmod p$;
 $C := C \cdot C \bmod p$
 end

Step 3: **return** D

Output: $D = a^b \bmod p$.

$$O(2(k+1) \times k^2)$$

问题2: quadratic residues (续)

Proof. We have to prove that

$$|\{1^2 \bmod p, 2^2 \bmod p, \dots, (p-1)^2 \bmod p\}| = (p-1)/2. \quad (5.9)$$

We observe that for every $x \in \{1, \dots, p-1\}$,

$$(p-x)^2 = p^2 - 2px + x^2 = p(p-2x) + x^2 \equiv x^2 \pmod{p}.$$

①

Thus, we have proved that the number of quadratic residues modulo p is at most $(p-1)/2$.

②

Now it is sufficient to prove that for every $x \in \{1, \dots, p-1\}$, the congruence $x^2 \equiv y^2 \pmod{p}$ has at most one solution $y \in \{1, 2, \dots, p-1\}$ different from x .

Without loss of generality we assume $y > x$, i.e., $y = x + i$ for some $i \in \{1, 2, \dots, p-2\}$. Thus,

$$x^2 \equiv (x+i)^2 \equiv x^2 + 2ix + i^2 \pmod{p}.$$

This directly implies

$$2ix + i^2 \equiv i(2x + i) \equiv 0 \pmod{p}.$$

Since \mathbb{Z}_p is a field⁴⁰ and $i \in \{1, 2, \dots, p-1\}$,⁴¹

$$2x + i \equiv 0 \pmod{p}. \quad (5.10)$$

Since the congruence (5.10) has exactly one solution⁴² $i \in \{1, \dots, p-1\}$, the proof is completed.⁴³ \square

问题3: SSSA

- 作为一个单边错Monte Carlo算法, SSSA是识别质数的还是识别合数的? 这两种说法有区别吗?
- 为什么SSSA是一个单边错Monte Carlo算法?
- 定理5.3.3.1的基本证明思路是什么?
- SSSA在使用时的局限是什么? 为什么这一局限难以打破?

问题3: SSSA (续)

- 为什么SSSA是一个单边错Monte Carlo算法?

Algorithm 5.3.3.5 (SSSA SIMPLIFIED SOLOVAY-STRASSEN ALGORITHM).

Input: An odd number n with odd $(n-1)/2$.
Step 1: Choose uniformly an $a \in \{1, 2, \dots, n-1\}$
Step 2: Compute $A := a^{\frac{n-1}{2}} \bmod n$
Step 3: if $A \in \{1, -1\}$
 then return ("PRIME") {reject}
 else return ("COMPOSITE") {accept}.

Theorem 5.3.3.1. For every odd n such that $(n-1)/2$ is odd (i.e., $n \equiv 3 \pmod{4}$),

- (i) if n is a prime, then $a^{(n-1)/2} \bmod n \in \{1, -1\}$ for all $a \in \{1, \dots, n-1\}$,
- (ii) if n is composite, then $a^{(n-1)/2} \bmod n \notin \{1, -1\}$ for at least one half of the a 's from $\{1, 2, \dots, n-1\}$.

问题3: SSSA (续)

Proof. Fact (i) is a direct consequence of Theorem 2.2.4.32.

To prove (ii) we consider the following strategy. Let n be composite. A number $a \in \mathbb{Z}_n$ is called **Eulerian** if $a^{(n-1)/2} \bmod n \in \{1, -1\}$. We claim that to prove (ii) it is sufficient to find a number $b \in \mathbb{Z}_n - \{0\}$ such that b is not Eulerian and there exists a multiplicative inverse b^{-1} to b . Let us prove this claim. Let $Eu_n = \{a \in \mathbb{Z}_n \mid a \text{ is Eulerian}\}$. The idea of the proof is that the multiplication of elements of Eu_n by b is an injective mapping into $\mathbb{Z}_n - Eu_n$. For every $a \in Eu_n$, $a \cdot b$ is not Eulerian because

$$(a \cdot b)^{\frac{n-1}{2}} \bmod n = \left(a^{\frac{n-1}{2}} \bmod n\right) \cdot \left(b^{\frac{n-1}{2}} \bmod n\right) = \pm b^{\frac{n-1}{2}} \bmod n \notin \{1, -1\}.$$

Now it remains to prove that $a_1 \cdot b \not\equiv a_2 \cdot b \pmod{n}$ if $a_1 \neq a_2$, $a_1, a_2 \in Eu_n$. Let $a_1 \cdot b \equiv a_2 \cdot b \pmod{n}$. Then by multiplying the congruence with b^{-1} we obtain

$$a_1 = a_1 \cdot b \cdot b^{-1} \bmod n = a_2 \cdot b \cdot b^{-1} \bmod n = a_2.$$

So, $|\mathbb{Z}_n - Eu_n| \geq |Eu_n|$.

问题3: SSSA (续)

- SSSA在使用时的局限是什么？为什么这一局限难以打破？

Algorithm 5.3.3.5 (SSSA SIMPLIFIED SOLOVAY-STRASSEN ALGORITHM).

Input: An odd number n with odd $(n-1)/2$.
Step 1: Choose uniformly an $a \in \{1, 2, \dots, n-1\}$
Step 2: Compute $A := a^{\frac{n-1}{2}} \bmod n$
Step 3: if $A \in \{1, -1\}$
 then return ("PRIME") {reject}
 else return ("COMPOSITE") {accept}.

Carmichael numbers:

$$a^{n-1} \equiv 1 \pmod{n} \text{ for all } a \in \{1, 2, \dots, n-1\} \text{ with } \gcd(a, n) = 1.$$

问题4: Solovay-Strassen和Miller-Rabin

- Miller-Rabin的基本原理是什么？

Algorithm 5.3.3.14. MILLER-RABIN ALGORITHM

```
Input:  An odd number  $n$ .
Step 1: Choose  $a$  uniformly at random from  $\{1, 2, \dots, n-1\}$ .
Step 2: Compute  $a^{n-1} \bmod n$ .
Step 3: if  $a^{n-1} \bmod n \neq 1$  then
        return ("COMPOSITE") -accept"
      else begin
        compute  $s$  and  $m$  such that  $n-1 = s \cdot 2^m$ ;
        for  $i := 0$  to  $m-1$  do
           $r[i] := a^{s \cdot 2^i} \bmod n$  -by repeated squaring";
           $r[m] := a^{n-1} \bmod n$ ;
          if there exists  $j \in \{0, 1, \dots, m-1\}$ , such that
             $r[m-j] = 1$  and  $r[m-j-1] \notin \{1, -1\}$ ,
          then return ("COMPOSITE") -accept"
          else return ("PRIME") -reject"
        end
```

- 就random sampling的一般原理而言，SSSA、Solovay-Strassen、Miller-Rabin的根本区别是什么？

问题5: prime generation

- 算法5.3.3.16的基本原理是什么？
- 为什么它几乎总能输出正确的结果？证明过程中两个概率算式的含义是什么？

问题5: prime generation (续)

Algorithm 5.3.3.16. PRIME GENERATION(l, k) (PG(l, k))

Input: l, k .

Step 1: Set $X := \text{"still not found"}$;
 $I := 0$

Step 2: **while** $X = \text{"still not found"}$ and $I < 2l^2$
 do begin generate randomly a bit sequence a_1, \dots, a_{l-2} and set
 $n = 2^{l-1} + \sum_{i=1}^{l-2} a_i 2^i + 1$;
 perform k runs of SOLOVAY-STRASSEN ALGORITHM on n ;
 if at least one of the k outputs is "Composite"
 then $I := I + 1$
 else do begin $X := \text{"already found"}$;
 output(n)
 end
 end

Step 3: **if** $I = 2l^2$ output("I did not find any prime").

问题5: prime generation (续)

- Probability of outputting “I did not find any prime”

$$\left[\left(1 - \frac{1}{2l}\right) \cdot \left(1 - \frac{1}{2^k}\right) \right]^{2l^2} < \left(1 - \frac{1}{2l}\right)^{2l^2} = \left[\left(1 - \frac{1}{2l}\right)^{2l} \right]^l < \left(\frac{1}{e}\right)^l = e^{-l}.$$

- Probability of outputting a composite number

$$\begin{aligned} & \left(1 - \frac{1}{2l}\right) \cdot \frac{1}{2^l} + \sum_{i=1}^{2l^2-1} \left[\left(1 - \frac{1}{2l}\right) \cdot \left(1 - \frac{1}{2^l}\right) \right]^i \cdot \left(1 - \frac{1}{2l}\right) \cdot \frac{1}{2^l} \\ & \leq \left(1 - \frac{1}{2l}\right) \cdot \frac{1}{2^l} \cdot \left(\sum_{i=1}^{2l^2-1} \left(1 - \frac{1}{2l}\right)^i + 1 \right) \\ & \leq \left(1 - \frac{1}{2l}\right) \cdot \frac{1}{2^l} \cdot 2l^2 \leq \frac{l^2}{2^{l-1}}. \end{aligned}$$

问题6: verifying matrix multiplication

Input: Matrix $A \in \mathbb{R}^m \times p$, $B \in \mathbb{R}^p \times n$, and $C \in \mathbb{R}^m \times n$.

Output: True if $C = A \cdot B$; false if $C \neq A \cdot B$

- 你能不能基于random sampling给出一个单边错Monte Carlo算法?

```
begin
  i=1
  repeat
    Choose  $r=(r_1, \dots, r_n) \in \{0,1\}^n$  at random.
    Compute  $C \cdot r$  and  $A \cdot (B \cdot r)$ 
    if  $C \cdot r \neq A \cdot (B \cdot r)$ 
      return FALSE
    endif
    i = i + 1
  until i=k
  return TRUE
end
```

问题6: verifying matrix multiplication (续)

Theorem: The algorithm is correct with probability at least $1 - (\frac{1}{2})^k$.

We will prove that if $A \cdot B \neq C$ then $Pr[A \cdot B \cdot r = C \cdot r] \leq 1/2$.

If $A \cdot B \neq C$, by definition we have $D = A \cdot B - C \neq 0$. Without loss of generality, we assume that $d_{11} \neq 0$.

On the other hand, $Pr[A \cdot B \cdot r = C \cdot r] = Pr[(A \cdot B - C) \cdot r = 0] = Pr[D \cdot r = 0]$.

If $D \cdot r = 0$, then the first entry of $D \cdot r$ is 0, that is

$$\sum_{j=1}^n d_{1j} r_j = 0$$

Since $d_{11} \neq 0$, we can solve for r_1 :

$$r_1 = \frac{-\sum_{j=2}^n d_{1j} r_j}{d_{11}}$$

If we fix all r_j except r_1 , the equality holds for at most one of the two choices for $r_1 \in \{0, 1\}$. Therefore, $Pr[ABr = Cr] \leq 1/2$.

We run the loop for k times. If $C = A \cdot B$, the algorithm is always correct; if $C \neq A \cdot B$, the probability of getting the correct answer is at least $1 - (\frac{1}{2})^k$.