高度セキュリティPBL
先進セキュリティPBL
セキュリティPBL特論I
演習0.　Pythonで数学を

大阪大学大学院 工学研究科
電気電子情報通信工学専攻情報通信工学部門
宮地研究室　奥村　伸也

目次

Pythonを用いた数学実験

1. 素因数分解
2. 乱数生成とχ^2 検定
3. 1次合同式の解法

Pythonを用いた数学実験

1. 素因数分解

sympy.factorint(n)：自然数 n の因数分解を返す関数

例：sympy.factorint(527589230328730923 87)

➔ $\{3:\ 1, 11:\ 1, 12113:\ 1, 131986728590803:\ 1\}$

➔ $527589230328730923 87$
$= 3 \times 11 \times 12113 \times 131986728590803$

どういった自然数がsympy.factorint()による因数分解が難しいかを見てみる.

演習1-1.

次の90ビットの整数を素因数分解して，
素因数分解が完了するまでの計算時間の違い
について考察してみよう．

(a) 778851447666082307730996709

(b) 113604182799520985615402523 1

(c) 639745742161865592519176783

※100回実験し，計算時間の最小値・最大値，
平均・分散を求めよう．

演習1-2 (チャレンジ課題).

演習1-1の結果から，演習1-1の(a)～(c)の合成数より素因数分解に時間がかかる90ビットの合成数を作ってみよう.

※10回実験し，計算時間の最小値・最大値,
平均・分散を求めよう.

2. 乱数生成とχ^2 検定

χ^2 検定(適合度検定)：(例えば)ある集合S の元 $y_1, \dots, y_k \in S$ $(k > \#S)$ が S 上のある確率分布に従ってサンプリングされたものかどうかを検定する手法.

例

・帰無仮説：元 $y_1, \dots, y_k \in S(k > \#S)$ は一様分布に従う

・対立仮説：元 $y_1, \dots, y_k \in S(k > \#S)$ は一様分布に従わない

どちらの仮説が正しいかを Python では以下のように検定する.

有意水準を α とする.

$S = \{s_1, \dots, s_h\}, n_i = \#\{j \mid y_j = s_i\},$

from scipy.stats import chisquare

chisquare($[n_1, \dots, n_h]$)[1]$< \alpha \Rightarrow$ 帰無仮説を棄却

※各 n_i を求めるには，collections.Counter()を利用するとよい.

演習2.

1. secrets.choice([1,2,3,4,5,6]) と random.randint(1, 6)はどちらも1〜6 の数字をランダムに返す関数である.
これらの関数を1〜6の目を持つさいころを振る操作に見立てて, 入力 n に対して n 回さいころを振る
関数 dice1(n) (choice), dice2(n) (randint) を作成せよ.
Input: n
Output: n回さいころを振って出た目のリスト T

2. $n = 60000$ としたとき, dice1(n), dice2(n)の出力が $\{1,2,3,4,5,6\}$ 上の一様分布に従っているか(帰無仮説)を有意水準 1% $(\alpha = 0.01)$で検定せよ. 検定は100回行い,
何度帰無仮説が棄却されるかをカウントせよ.

3. 1次合同式の解法

0 でない整数 $a, b, n\ (n > 0)$ について，次の一次合同式

$$ax \equiv b \pmod{n} \cdots (\#)$$

を解くことを考える：

(i) $\gcd(a, n) = 1$ のとき

合同式 $(\#)$ は 法 n に関してただ 1 つの解を持つ．つまり，

x, x' が $(\#)$ の解 $\Rightarrow x \equiv x' \pmod{n}$.

1. $aX + nY = 1 \cdots (*)$ を満たす整数 X, Y を求める.

Python では sympy.gcdex(a, n) で X, Y, $\gcd(a, n)$ が求まる.

2. $aX \equiv 1 \pmod{n}$ より $(ax)X \equiv (aX)x \equiv x \equiv bX \pmod{n}$
より x が求まる.

(ii) $\gcd(a, n) = d > 1$ のとき

合同式 $(\#)$ は 法 n に関して d 個の解を持つ.

$a = a'd, n = n'd$ とする．$ax = b + kn\ (\exists k \in \mathbb{Z})$ を満たす

$x \in \mathbb{Z}$ が $(\#)$ の解となる．$b = kn - ax$ より d は b を割り切る.

$b = b'd$ とおく.

$a'x = b' + kn'$ より $a'x \equiv b' \pmod{n'} \cdots (\#')$.
$\gcd(a', n') = 1$ より $(\#')$ は法 n' に関して 1 つの解を持つ.
$x = X \pmod{n'}$ を $(\#')$ の 1 つの解とする.
$a'X = b' + kn'$ より $aX = b + kn \Rightarrow x \equiv X \pmod{n}$ は $(\#)$ の解.
$a(X + jn') = aX + ajn' = aX + a'jn \equiv aX \pmod{n}$ より
$X, X + n', X + 2n', \ldots, X + (d-1)n'$ は $(\#)$ の解.
$(\#)$ は n を法として d 個の解を持つ.

<u>演習3.</u>
次の 1 次合同式の解のうち，合同式の法未満の解を全て
求めよ．すべての解をリストで出力すること.
(1) $759797598748x \equiv 579273497 \pmod{7529734979853354 3}$
(2) $5559529747936x \equiv 3739756 \pmod{75927344739475843 38}$
(3) $282814105706x \equiv 35258184 \pmod{642839066779736}$