Probabilistische Algorithmen

Zufallszahlen - Monte Carlo - Genetische Programmierung

Tobias Gaertner, Torsten Schröter

25. Mai 2009

Inhaltsverzeichnis

Pseudozufallszahlen
- Kongruenzmethode

Monte-Carlo-Algorithmen
- Äquivalenz von Multimengen
- Bsp Primzahltest

Genetische Programmierung

Woher bekomme ich Zufallszahlen?

- Der Mensch ist ein schlechter Zufallszahlengenerator
- Computer ist unvoreingenommen...aber arbeitet deterministisch

Woher bekomme ich Zufallszahlen?

- Der Mensch ist ein schlechter Zufallszahlengenerator
- Computer ist unvoreingenommen...aber arbeitet deterministisch

Zufall

Ein Zustand tritt dann zufällig ein, wenn er nicht vorhersagbar ist.

Beispiel

6 - 2 - 4 - 3 - 3 - 1 - 2 - 3 - 4 - 1 - 5 - 4 - 6 - 3 - 4 - 2

Beispiel

6 - 2 - 4 - 3 - 3 - 1 - 2 - 3 - 4 - 1 - 5 - 4 - 6 - 3 - 4 - 2

2 - 4 - 6 - 5 - 5 - 3 - 4 - 5 - 6 - 3 - 1 - 6 - 2 - 5 - 6 - 4

Beispiel

6 - 2 - 4 - 3 - 3 - 1 - 2 - 3 - 4 - 1 - 5 - 4 - 6 - 3 - 4 - 2

2 - 4 - 6 - 5 - 5 - 3 - 4 - 5 - 6 - 3 - 1 - 6 - 2 - 5 - 6 - 4

Zahlen unterscheiden sich jeweils um 2(mod 6)

Anforderungen an Zufallszahlen

- möglichst große bzw möglichst keine erkennbare Periode
- gleichförmige Verteilung
- keine erkennbaren Strukturen oder Zusammenhänge zwischen den gewonnenen Zahlen

Kongruenzmethode

- 1948 von LEHMER entwickelt
- deterministisches Verfahren zum erzeugen sogenannter **Pseudozufallszahlen**
- erzeugt eine "zufällige" Zahlenfolge ausgehend von einem Startwert (seed)

- $z_n = (a * z_{n-1} + b) mod\ c$
- für a,b,c sehr große Konstanten verwendet
  um nicht eine ungünstige Konstellation zu erwischen haben sich für a,b große Primzahlen und für c große 2er-Potenzen wie ($2^{32}, 2^{64} oder\ auch\ 2^{31} - 1$) bewährt

"bessere" Zufallszahlengeneratoren

- Gewinnung des seed aus einem unabhängigen, unvorhersehbaren Wert
- Systemzeit (z.B: ChezSCHEME - cpu-time)
- Gewinnung des seeds aus physikalischen Quellen

RandCam

Abbildung: http://von-und-fuer-lau.de/randcam.html

Ein Monte-Carlo-Algorithmus...

- liefert immer ein Ergebnis.
- kann sich mit einer festgelegten (geringen) Warscheinlichkeit irren.
- liefert keine Näherungslösungen.

Fehlerwarscheinlichkeit

- kann durch viele Wiederholungen so gering gehalten werden, dass er als sicher anzusehen ist
- wird daher sogar an sicherheitskritischen Stellen eingesetzt
- ein Ausfall aller Hardware in einem System ist um einiges! warscheinlicher

Äquivalenz von Multimengen

Multimengen

Multimenge ist eine Menge bei der jedes Element n-mal vorkommen darf.
Zum Beispiel ist $\{1, 2, 3, 4, 5\}$ eine Menge nicht aber $\{1, 2, 3, 3, 4, 5\}$.
Äquivalenzprüfung schwierig

Lösung

Äquivalenz von Multimengen

Multimengen

Multimenge ist eine Menge bei der jedes Element n-mal vorkommen darf.
Zum Beispiel ist $\{1, 2, 3, 4, 5\}$ eine Menge nicht aber $\{1, 2, 3, 3, 4, 5\}$.
Äquivalenzprüfung schwierig

Lösung

Sortieren und durchiterieren?
$\{2, 3, 4, 5, 2, 1, 3\}$
$\{1, 4, 5, 3, 3, 2, 2\}$
$\{1, 4, 5, 3, 3, 2, 7\}$

Äquivalenz von Multimengen

Multimengen

Multimenge ist eine Menge bei der jedes Element n-mal vorkommen darf.
Zum Beispiel ist $\{1, 2, 3, 4, 5\}$ eine Menge nicht aber $\{1, 2, 3, 3, 4, 5\}$.
Äquivalenzprüfung schwierig

Lösung

Sortieren und durchiterieren?
$\{1, 2, 2, 3, 3, 4, 5\}$
$\{1, 2, 2, 3, 3, 4, 5\}$
$\{1, 2, 3, 3, 4, 5, 7\}$

Äquivalenz von Multimengen

Multimengen

Multimenge ist eine Menge bei der jedes Element n-mal vorkommen darf.
Zum Beispiel ist $\{1, 2, 3, 4, 5\}$ eine Menge nicht aber $\{1, 2, 3, 3, 4, 5\}$.
Äquivalenzprüfung schwierig

Lösung

Sortieren und durchiterieren?
$\{1, 2, 2, 3, 3, 4, 5\}$
$\{1, 2, 2, 3, 3, 4, 5\}$
$\{1, 2, 3, 3, 4, 5, 7\}$
Sehr aufwendig! $\mathcal{O}(n^2)$

Äquivalenz von Multimengen

Alternative Vorgehensweise:

- Bilden eines Polynoms für jede Multimenge nach folgendem Muster $p_{M_m}(x) = (x - x_1)(x - x_2)...(x - x_n)$
- Die Multimengen sind höchstwahrscheinlich äquivalent wenn $p_{M_1}(x) = p_{M_2}(x)$ für **alle** ganzen Zahlen x zutrifft.
- nicht äquivalent wenn das Verfahren für mindestens ein x fehlschlägt

Äquivalenz von Multimengen

Beispiel mit $x = 3$:
Polynom für Multimenge 1:
$p_{M_1}(x) = (x-2)(x-3)(x-4)(x-5)(x-2)(x-1)(x-3)$

Multimengen	Polynomwert
{2, 3, 4, 5, 2, 1, 3}	0
{1, 4, 5, 3, 3, 2, 2}	0
{1, 4, 5, 3, 3, 2, 7}	0

Äquivalenz von Multimengen

Beispiel mit $x = 7$:
Polynom für Multimenge 1:
$p_{M_1}(x) = (x-2)(x-3)(x-4)(x-5)(x-2)(x-1)(x-3)$

Multimengen

$\{2, 3, 4, 5, 2, 1, 3\}$
$\{1, 4, 5, 3, 3, 2, 2\}$
$\{1, 4, 5, 3, 3, 2, 7\}$

Polynomwert

14400
14400
0

Primzahltest

Nach der klassischen Methode (durchprobieren mit Probeteilern) haben wir einen recht hohen Aufand, was für große Zahlen sehr ineffient ist.
Auf Grundlage des "Kleinen Satzes von Fermat" kann man mittels randomisieren einen Monte-Carlo Algorithmus entwickeln, der einen Aufwand von $\mathcal{O}(log\ n)$ hat.

Grundlagen

Der kleine Satz von Fermat:
$a^p \equiv a \ mod \ p$

Satz von Euler:
Für alle natürlichen Zahlen n und a mit $n \geq 2$ und $1 \leq a < n$, wobei $ggT(a, n) = 1$, gilt

$a^{\varphi(n)} \equiv 1 \ mod \ n$

daraus können wir für Primzahlen ableiten:
$a^{p-1} \equiv 1 \ mod \ p$

Pseudoprimzahlen

Zahlen die für einige Basen den Test bestehen aber dennoch keine Primzahlen sind.
z.B: $11 * 31 = 341$
$2^{341-1} \equiv 1 \ mod \ 341$

Carmichael-Zahlen sind Zahlen die zu allen teilerfremden Basen diesen Test bestehen. Die kleinste Carmichael-Zahl ist 561 $(= 3 * 11 * 17)$.

Primzahltest nach Solovay und Strassen

Vorgehensweise

- ggt(a,p) für zufälliges a, p ist prim
- $b = a^{\frac{n-1}{a}}$
- wenn $b \neq 1$ oder n-1 $\rightarrow$ keine Primzahl
- sonst Jacobisymbol berechnen um Pseudoprimzahlen auszuschließen

Genetische Programmierung

Was ist GP?

- Anfang der 90er Jahre von John Koza erfunden
- Genetische Programmierung geht von einer hohen Ebene der Abstraktion aus.
- Es wird eine Anforderung gestellt was getan werden muss und anschließend wird ein passgenaues Programm erstellt.

Genetische Programmierung

Vorgehensweise

- Anlegen einer Population von Programmen
- Bewerten der Programme anhand einer Fitnessfunktion (survival of the fittest) und Auswahl der besten.
- Analog zu natürlichen Prozessen lassen sich die Programme kreuzen, mutieren auch das duplizieren von Genen bzw. das löschen sind möglich.

Genetische Programmierung

Praktische Relevanz?

36 Konkurrenzfähige Lösungen von 1994-2002

2 Neuerfindungen

Fragen, Einwände, Diskussion ... jetzt.

Danke für Ihre Aufmerksamkeit!