Universidad del Valle

Algoritmos probabilisticos

750098M Simulación computacional

Contenido

Introducción

Estrategia para resolver problemas que tiene cierta probabilidad de NO dar el resultado esperado (puede equivocarse)

1. Son menos rigurosos que los algoritmos determinísticos
2. No tienen que ser previstos problemas como bucles infinitos, divisiones entre 0 u otras.
3. En caso de tener un comportamiento no deseado se aborta y se vuelve a comenzar

Introducción

Procedimiento

```
formular pregunta
   repetir n veces
       preguntar a una persona
       contabilizar la respuesta
determinar el resultado
```

El resultado puede mejorarse si se aumenta el número de repeticiones

Ejemplos

Ejemplo 1: Apuesta

30 personas participan en una fiesta. Se propone una apuesta. Por lo menos dos de ellas cumplen años el mismo día.

```
repetir n veces
     simular 30 días del año
     determinar si hay dos iguales
     contabilizar el resultado
determinar el resultado
```

Ejemplos

Ejemplo 1: Apuesta

¿Que tan cerca estamos con este resultado a la probabilidad verdadera? R./ Se puede determinar con un intervalo de confianza:

Ejemplos

Ejemplo 1: Apuesta

Procedimiento:

1. Repetir el algoritmo K veces. Obteniendo la probabilidad x_k de cada uno
2. Calcular el promedio $\bar{x}$ de cada uno de los resultados

$$\bar{x} = \sum_{i=1}^{k} x_k$$

K

3. Calcular el error estándar (estimación de la varianza)

$$S^2 = \sum_{i=1}^{k} \frac{\bar{x} - x_k}{k-1}$$

Ejemplos

Ejemplo 1: Apuesta

Procedimiento:

4. Determinar el intervalo de confianza (De acuerdo a la normal estandarizada)

$$-Z_\alpha \leq \sqrt{K}\frac{x-\mu}{S} \leq Z_\alpha$$

Ejemplos

Ejemplo 1: Apuesta

$n = 100$

Ahora tomamos 100 ejecuciones del algoritmo.

$\{0.72, 0.58, 0.72, 0.72, 0.7, 0.79, 0.66, 0.69, 0.79, 0.71,$
$0.71, 0.75, 0.77, 0.74, 0.73, 0.66, 0.64, 0.66, 0.72, 0.62,$
$0.62, 0.71, 0.64, 0.72, 0.73, 0.77, 0.75, 0.66, 0.77, 0.77,$
$0.8, 0.69, 0.69, 0.73, 0.67, 0.71, 0.62, 0.79, 0.69, 0.74,$
$0.74, 0.72, 0.71, 0.73, 0.68, 0.72, 0.67, 0.72, 0.71, 0.7,$
$0.7, 0.68, 0.74, 0.68, 0.76, 0.73, 0.64, 0.68, 0.7, 0.68,$
$0.69, 0.72, 0.79, 0.76, 0.74, 0.73, 0.6, 0.72, 0.69, 0.71,$
$0.7, 0.69, 0.76, 0.73, 0.71, 0.62, 0.74, 0.72, 0.71, 0.65,$
$0.74, 0.63, 0.67, 0.69, 0.75, 0.66, 0.79, 0.67, 0.69, 0.71,$
$0.65, 0.68, 0.78, 0.68, 0.69, 0.75, 0.74, 0.61, 0.66, 0.67\}$

Ejemplos

Ejemplo 1: Apuesta

Encontramos que:

$$\overline{x} = 0.7053$$

$$S^2 = 0.0021787$$

$$S = 0.0466765$$

$$\sum \frac{|\bar{x} - x_k|}{k-1} = S^2$$

y para α =0.05;

$$Z_{\alpha/2} = 1.96$$

$$0.696151 \leq \mu \leq 0.714449$$

$$\bar{x} - Z \times S \qquad \bar{x} + Z S$$

Ejemplos

Ejemplo 1: Apuesta

De acuerdo a este resultados, la probabilidad que se busca está en este intervalo:

$$0.696151 \leq prob \leq 0.714449$$

Contrastando contra el resultado análitico

$$prob = 1 - \frac{365 \cdot 364 \cdot \ldots \cdot 336}{365^{30}} = 0.706316$$

“Probalistico” no significa “Incierto” si se conoce una estimación del error.

Caracteristicas

1. Estos algoritmos pueden devolver un resultado diferente en cada ejecución
2. Debe usarse un buen generador de número Pseudoaleatorios
3. Se hace una búsqueda aleatoria en lugar de una exhaustiva
4. En el diseño de estos algoritmos se proveen mecanismos para controlar el error

Ventajas

1. Son eficientes (Aptos para problemas en los que la solución optima es muy costosa)
2. Son capaces de obtener soluciones buenas, en problemas que no son tratables. Ejemplo: Agente viajero, determinar si un número grande es primo, entre otros NP Hard.
3. Son capaces de encontrar varias soluciones. Ejemplo: n rei~~a~~nas.
4. Generalmente, no se utilizan conceptos matemáticos complejos.

Desventajas

1. No devuelven soluciones correctas
   a. Se puede controlar el error con métodos estocásticos
   b. Los algoritmos deterministicos pueden devolver resultados con errores de redondeo
2. Su análisis con frecuencia es dficiil, ya que deben utilizarse procedimientos probabilisticos
   a. No hay mejor ni peor caso
   b. Se el tiempo esperado o caso promedio para determinar la complejidad

Clasificación

- Algoritmo probabilistico numérico: Devuelven el resultado aproximado. El resultado mejora a medida que se aumenta el número de ejecucions. Para controlar el error se utiliza un intervalo de confianza.
- Algoritmos las vegas: Toman decisiones al azar, sin embargo admiten si encontraron una solución correcta o no. Al repetir el algoritmo se llega a una solución correcta.

Clasificación

.

- Algoritmos de MonteCarlo: Dan con alta probabilidad del resultado correcto, pero no pueden reconocer si es correcta o no. Se estima la probabilidad de equivocarse del algoritmo y la probabilidad crece con el número de repeticiones del algoritmo.

Clasificación

Ejemplo: Algoritmo probabilístico numérico

Problema: Encontrar para unjuego de póker (52 cartas) la probabilidad de encontrar una escalera (cinco cartas seguidas, que no son A,J,Q ni K). Realizamos ensayos

- 100 manos. p = 0
- 1000 manos. p = 0.001
- 10000 manos p = 0.0045
- 100000 manos p = 0.00352

Clasificación

Ejemplo: Algoritmo probabilístico numérico

A medida que se aumenta el número de iteraciones se mejora la solución. Ahora, el análisis numérico

- En total hay Comb(52.5) posibles manos (de 52 cartas tomo 5 arbitrariamente)
- Hay 9 posibles comienzos de escalera (Valores 2 a 10)
- Hay 4 palos, por lo que tenemos 4 opciones por posición, entonces:

$$p = \frac{9*4*4*4*4*\cancel{4}}{Comb(52,5)} = 0.00345603$$

Clasificación

Ejemplo: Las vegas

Problema: Rellenar una matriz 4x4 con los números 1 a 16 de tal forma, dada una posición cualquiera, ninguno de sus vecino tenga un número consecutivo.

El algoritmo las Vegas coloca los números en orden aleatorio. Tal que cada número cumple la condición con respecto a los anteriores. Si se llega a un punto en que no se puede colocar ningún número, el algoritmo terminará sin éxito

Clasificación

Ejemplo: Las vegas

6	9	12	10
4	2	7	3
13	11	15	1
8	5		

sin éxito

1	13	4	11
9	16	10	7
5	14	3	12
2	8	6	15

repuesta correcto

4	9	6	8
16	11	14	12
5	2	7	1
10	15	13	3

repuesta correcto

12	1	8	4
16	3	13	15
5	10	6	11
2	14	9	

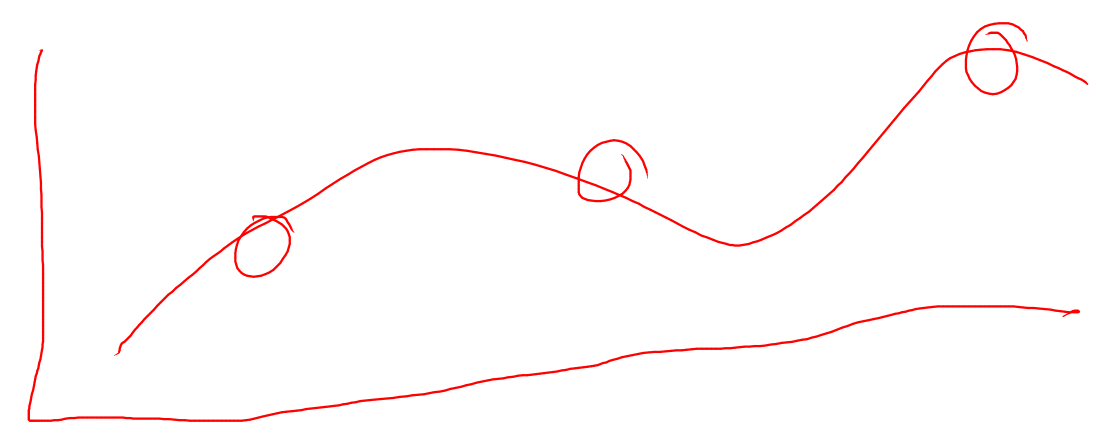
sin éxito

El algoritmo toma decisiones aleatorias que no siempren llegan a una solución. El algoritmo reconoce si falla. Se repite hasta que se llegue a una solución correcta.

Clasificación

Algoritmos de MonteCarlo

Problema: Una función f está definida dentro del rango [-20000,20000]. Se quiere saber si la función es siempre positiva.



El algoritmo de MonteCarlo genera un número aleatorio en el rango y evalúa la función. Si f(a) < 0 devuelve "la función es positiva" y si f(a) >= 0 devuelve "la función es positiva" Se repite muchas veces hasta que la probabilidad de acierto supera un valor, ejemplo 0.99

Conclusiones

- Los algoritmos probabilisticos resuelven un problema paso a paso, pero existe la probabilidad de que no generen un resultado esperado
- Como cualquier algoritmo, se debe analizar el error. Al ser probabilisticos, el error es una variable aleatoria.
- Existen diferente tipo de algoritmos (Numéricos, LasVegas y MonteCarlo). Para un problema específico se debe seleccionar el tipo de algoritmo apropiado.
- El análisis de eficiente se realizan bajo caso promedio de ejecución.