Universidad Nacional Autónoma de México

Escuela Nacional de Estudios Superiores

Unidad Juriquilla

Computación II

Practica 4

Interpolación con método de diferencias divididas de Newton

Bruno Arturo López Pacheco

Dr. Ulises Olivares Pinto

No. De Cuenta 317347140

Licenciatura en Tecnología

Objetivo

Utilizar el método de diferencias divididas de Newton para interpolar un conjunto de puntos de entrada en un plano. Hacer uso de un dataset disponibles en bancos de datos para probar este método con datos reales.

Marco teórico

Método de diferencias divididas de Newton

Este método nos dice que:

Sea f_n una variable discreta de n elementos y sea x_n otra variable discreta de n elementos los cuales corresponden, por parejas, a la imagen u ordenada y abscisa de los datos que se quieran interpolar, respectivamente, tales que:

$$f(x_k) = f_k, \quad k = 0, \dots, n$$

Este método es muy cómodo en ciertos casos, sobre todo cuando se quiere calcular un polinomio interpolador de grado elevado.

Este polinomio de grado n tiene la forma

$$a_0 + \sum_{j=1}^{n} a_j \, g_j(x)$$

Definiendo $g_j(x)$ como

$$g_j(x) = \prod_{i=0}^{j-1} (x - x_i)$$

Y definiendo a_j como

$$a_0 = f[x_0], a_1 = f[x_0, x_1], \dots, a_j = f\left[x_0, x_1, \dots, x_{j-1}, x_j\right]$$

Los coeficientes a_j son las llamadas diferencias divididas.

Una vez que se hayan realizado todos los cálculos, se puede observar que hay muchas más diferencias divididas que coeficientes a_j. El cálculo de todos los términos intermedios debe realizarse simplemente porque son necesarios para poder formar todos los términos finales. Pero los términos usados en la construcción del polinomio interpolador son todos aquellos que involucren a x_0.

Estos coeficientes se calculan mediante los datos que se conocen de la función f.

En donde $f\left[x_0, x_1, \dots, x_{j-1}, x_j\right]$ queda definido como:

$$f\left[x_i, x_{i+1}, \dots, x_{i+j-1}, x_{x+j}\right] = \frac{f\left[x_{i+1}, \dots, x_{i+j-1}, x_{i+j}\right] - f\left[x_i, x_{i+1}, \dots, x_{i+j-1}\right]}{x_{i+j} - x_i}, f[x_i] = f(x_i)$$

Metodología

Pseudocódigo del método de diferencias divididas de Newton

Para este algoritmo se crea una tabla de diferencias divididas para así manejar más fácil los datos:

Dados x_i (valores de x), y f_i (valores de y):

```
#Tabla
n = tamaño(xi)
ki = arreglo(0,n,1)
tabla = concantenar(([ki],[xi],[fi]),eje=0)
tabla = transpuesta(tabla)

#Tabla para diferencias divididas
tabDiferencias = zeros(shape=(n,n),dtype=float)
tabla = concatenar((tabla,tabDiferencias), eje=1)

#Llenado de tabla
[n,m] = forma(tabla)
diagonal = n-1

Para j en rango(3,m):
  paso=j-2 #Inicia en 1
  Para i en rango(0,diagonal):
    tabla[i,j] = (tabla[i+1,j-1]-tabla[i,j-1])/(xi[i+paso]-xi[i])
  diagonal = diagonal-1

#Creación de polinomio
dDividida = tabla[0,3:]
n = tamaño(tabDiferencias)
polinomio = fi[0]
Para j en rango(0,n-1):
    factor = dDividida[j]
    termino = 1
    Para k en rango(0,j+1):
        termino = termino*(x-xi[k])
    polinomio = polinomio + termino*factor
polisimple = polinomio.expand()

imprimir("Polinomio simplificado\n",polisimple)
return polisimple
```

Pruebas

Para las pruebas de este algoritmo se escogió un dataset obtenido de la colección de datos de "Our world in data", el dataset contiene información de las muertes por VIH, incidencia de casos y prevalencia de casos del VIH. Viene información de cada país por año. Para probar este algoritmo se uso de información del país de México acotando el tiempo entre el año 2013 y 2017.

También se hizo una prueba haciendo uso de datos aleatorios. El pseudocódigo es el siguiente:

```
def valuesRandomInt(self,minX,maxX,minRandom,maxRandom):
    #Entradas:minimo valor de X, máximo valor de X
    #         intervalo mínimo y máximo de valores aleatorios
    X=arreglo(minX,maxX+1,1)
    Y=arreglo De Ceros Como El Vector X

    Para i en rango(len(Y)):
      Y[i]=EnterosAleatorios(minRandom,maxRandom+1)

    return X,Y
```

Resultados

Se obtuvo el siguiente polinomio:

$$f(x) = 51.4424x^4 - 414637.9726x^3 + 1253277858 - 6996x^2 - 1683616047977.19x + 848144649769592$$

Este polinomio se ajusta muy bien a los datos y tiene un error de 0.009%, comparando con el mismo dataset.

Se tienen muchas discusiones con respecto a los datos. Ya que primero se hizo la prueba del algoritmo haciendo uso de un dataset del índice de la temperatura global. Pero por la naturaleza de los datos hacía que solo se pudiera tener un polinomio decente cuando los valores en x eran 5 datos. Esto es un problema porque son muy pocos datos. Después se realizó la prueba con los datos reportados en esta práctica, que es sobre las muertes por VIH, pero de igual forma, debido a la naturaleza de los datos, los polinomios eran erróneos.

Después se corrió una última prueba haciendo uso de la función descrita en el apartado de "Pruebas". Estos datos aleatorios pueden trabajar hasta con 22 datos, lo cual no es demasiado, pero si es 4 veces más que con los datos recabados de bancos de datos.

Conclusión

Este algoritmo es muy bueno para datos que no presenten tantas anomalías, o para datos introducidos por el usuario, que no varían demasiado. De igual forma, este algoritmo presenta problemas cuando el grado del polinomio crece demasiado, incluso la evaluación de este polinomio hace que el algoritmo sea más tardado; pero este método es muy eficaz para pocos datos.

Referencias

Chapra, S., & Canale, R. (2011). METODOS NUMERICOS PARA INGENIEROS (Spanish Edition) (6th ed.). McGraw-Hill Interamericana de España S.L.

Datos recabados de: Our World in Data https://ourworldindata.org/health-meta#infectious-diseases