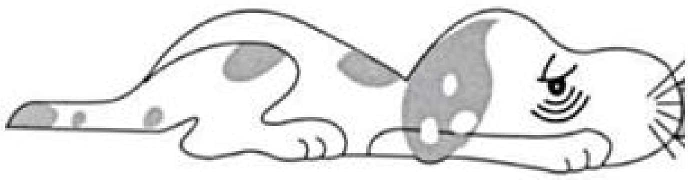
Juan Castañeda
David Vanegas
Daniel Beltrán

Interpolador

Objetivo

Desarrollar un interpolador para representar de la forma más exacta posible la imagen de un perro, empleando la menor cantidad de datos.

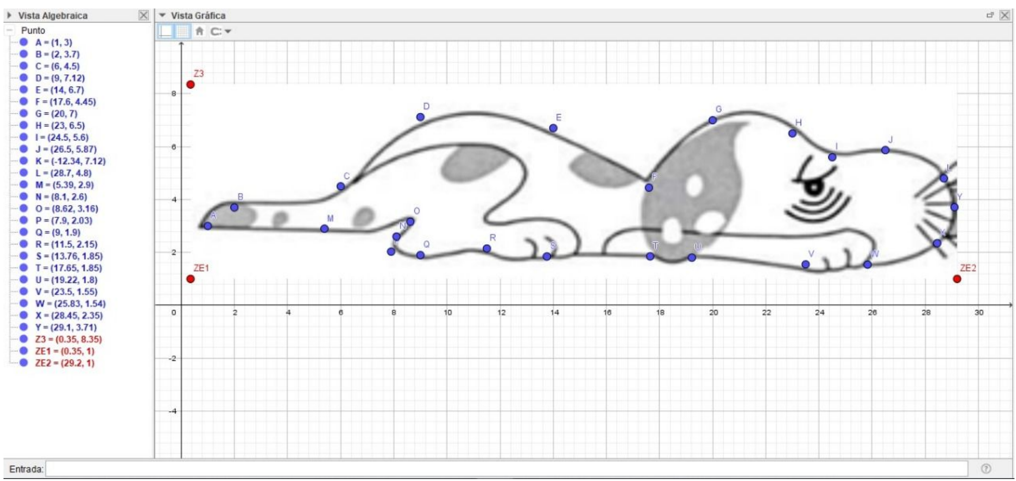


Metodología empleada

1. Metodología que explique cómo se seleccionaron k puntos con k<n con n el total de puntos dados(Selección de más puntos o de los puntos de la parte de abajo)

   La manera en la que se escogió los n puntos se realizaron a partir de los siguientes criterios partiendo que P(x) = $ax^3 + bx^2 + cx + d$:

   - Que las partes de la función a trozos P(x) pasen por ese punto. Es decir, que las dos Pn(x) que rodean al f(x) que queremos aproximar, sean igual a f(x) en cada uno de estos puntos.
   - De manera intuitiva podemos afirmar cuáles eran los puntos más importantes, esto debido a que representan un maximo o minimo relativo entre un intervalo [a, b]

   Aquí se puede ver todos los puntos que dejan ver de manera clara que son máximos o mínimos como el punto D entre el intervalo [C, E].

   - A medida que la curva es más suave podemos afirmar que se requiere de menos puntos mientras que si esta tiene un comportamiento más abrupto se necesita de más puntos para la correcta interpolación

Juan Castañeda
David Vanegas
Daniel Beltrán

2. Algoritmo que se aplicó por ejemplo,interpolación polinómica y como soluciono el sistema

Algoritmo de Splines

En el subcampo matemático del análisis numérico, un spline es una curva diferenciable definida en porciones mediante polinomios. En los problemas de interpolación, se utiliza a menudo la interpolación mediante splines porque da lugar a resultados similares requiriendo solamente el uso de polinomios de bajo grado, evitando así las oscilaciones, indeseables en la mayoría de las aplicaciones, encontradas al interpolar mediante polinomios de grado elevado.

Cuando se habla de oscilaciones se habla de manera precisa del fenómeno de Runge el cual es un problema cuando se usa interpolación polinómica con polinomios de alto grado entre nodos equidistantes Para el ajuste de curvas, los splines se utilizan para aproximar formas complicadas.

- Splines cúbicos

Un trazador cúbico S es una función a trozos que interpola a f en los n+1 puntos (x0,y0),(x1,y1),(x2,y2), ...,(xn,yn) (con a=x0<x1<...<xn =b). S es definida de la siguiente manera:

$$S(x) = \begin{cases} S_0(x) & \text{si} \quad x \in [x_0, x_1], \\ S_1(x) & \text{si} \quad x \in [x_1, x_2], \\ \vdots & \quad \vdots \\ S_{n-1}(x) & \text{si} \quad x \in [x_{n-1}, x_n], \end{cases}$$

- ***IMPLEMENTACIÓN***

Algunas curvas presentan "picos" así que se construye un trazador para cada curva entre cada dos picos. El tratamiento de picos requiere usualmente un trazador con frontera sujeta.
El proceso de construcción del trazador cúbico consiste en determinar cada polinomio cúbico Sj(x), es decir, buscar sus coeficientes ai, bi, ci y di. La definición nos da las condiciones que se deben cumplir. De estas condiciones podemos obtener un sistema de ecuaciones 4n×4n, donde las incógnitas son todos los coeficientes ai, bi, ci y di donde i = 0,1,...,n−1. Lo que obtenemos es un trazador cúbico único.

3. **Validación de Resultados**

Para validar los resultados se tendrá principalmente en cuenta el contorno producido a partir de la interpolación. Es decir, que debe ser posible reconocer la figura compuesta por curvas suaves. De igual forma es deseable que los errores obtenidos en cada punto presenten una tendencia a 0.

PRODUCTOS

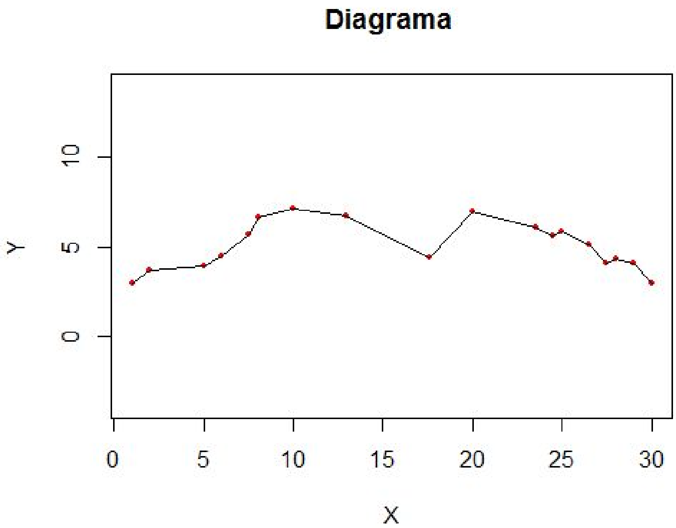
1. **Algoritmo, requerimientos, codificación**

```
x=c(1,2,6,9,14,17.6)
x1=c(17.6,20,23,24.5,26.5,28.7,29.1)
x2=c(28.4,29,29.275)
y=c(3,3.7,4.5,7.12,6.7,4.45)
y1=c(4.45,7,6.5,5.6,5.87,4.8,3.71)
y2=c(5.1,4.1,3)

a=c(1, 5.39, 8.62)
a1=c(7.9,8.1,8.62)
a2=c(7.9,9,11.5,13.76,16.95,20.22,23.5,26.83,28.45,29.1)
ay=c(3,2.5,3.16)
ay1=c(2.03,2.6,3.16)
ay2=c(2.03,1.9,2.15,1.85,1.85,1.8,1.55,1.54,2.35,3.71)

plot(x, y, xlim=c(1,30), ylim=c(2, 20))
lines(spline(x, y, n = 20), col = 2)
par(new = TRUE)
plot(x1, y1, xlim=c(1,30), ylim=c(2, 20))
lines(spline(x1, y1, n = 20), col = 2)
par(new = TRUE)
plot(a, ay, xlim=c(1,30), ylim=c(2, 20))
lines(spline(a, ay, n = 20), col = 2)
par(new = TRUE)
plot(a1, ay1, xlim=c(1,30), ylim=c(2, 20))
lines(spline(a1, ay1, n = 20), col = 2)
par(new = TRUE)
plot(a2, ay2, xlim=c(1,30), ylim=c(2, 20))
lines(spline(a2, ay2, n = 20), col = 2)
```

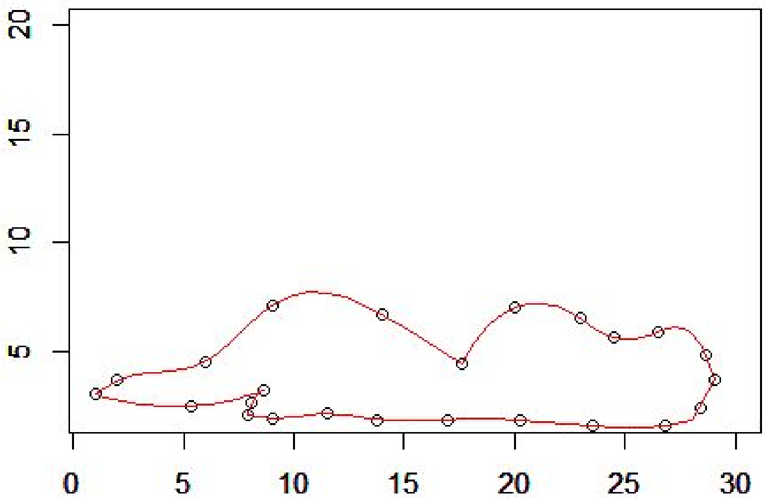
2. **Codificación, tabla donde esta la interpolación en los n-k puntos (no seleccionados), el polinomio o la función interpolante. En un plano los puntos originales, los utilizados,el contorno y el interpolado(utilice el grosor minimo para la curva)**

- Puntos y gráficos originales

Juan Castañeda
David Vanegas
Daniel Beltrán



- Puntos y gráficos del interpolador



3. **Cota de error**

La cota de error se define por la función:

$$\max_{a \le x \le b} |f(x) - S(x)| \le \frac{5M}{384}$$

$$|f(x) - S(x)| \le \frac{5M}{384}$$

Juan Castañeda
David Vanegas
Daniel Beltrán

4. **Tabla donde están los valores interpolados(tenga en cuenta los que no utilizo), los originales y el error relativo , calcule un error relativo total como la suma de los errores relativos**

x	y	f(x)	Error relativo
1	3	3	0
2	3.7	3.7	0
5	3.9	4.11	0.0538461538
6	4.5	4.5	0
7.5	5.7	5.84	0.0245614035
8.1	6.69	6.43	0.0388639761
10	7.12	7.41	0.0407303371
13	6.7	6.99	0.0432835821
17.6	4.45	4.45	0
20	7	7	0
23.5	6.1	6.16	0.00983606557
24.5	5.6	5.6	0
25	5.87	5.52	0.0596252129
26.5	5.15	5.87	0.122657581
27.5	4.1	6.07	0.513715711
28	4.3	5.84	0.358139535
29	4.1	4.02	0.0195121951
30	3	-0.442	1.14733333

Error relativo total = 2,432105083

5. **Índice de Jaccard**

Se calcula el número de aciertos con precisión de una cifra:

A = {3, 3.7, 4.1, 4.5, 5.7 , 6.3, 7.6, 7.2 , 4.4 ,7, 6.1, 5.6 , 5.5, 5.8, 6.0, 5.8, 4.0, -0.04}
B = {3, 3.7, 3.9 , 4.5, 5.7, 6.6, 7.1, 6.7, 4.4, 7 , 6.1, 5.6, 5.8, 5.1, 4.1, 4.3, 4.1, 3}

Juan Castañeda
David Vanegas
Daniel Beltrán

J(A,B) = |A ∩ B| / |A ∪ B| = 8/18 = 0.44

6. Eficiencia del método

Dado que no se posee un método directo para medir la eficiencia del algoritmo se utilizó la funcionalidad proc.time() en R para medir el tiempo de respuesta del algoritmo. Los resultados obtenidos fueron los siguientes:

Al ingresar los puntos escogidos el tiempo de respuesta del programa fue de 0.28 segundos

```
>
> proc.time()-t    # Detiene el cronómetro
   user  system elapsed
   0.05    0.03    0.28
>
> #t
```

Posteriormente se añaden 4 puntos más y se vuelve a realizar la interpolación. El nuevo tiempo obtenido es de 0.31 segundos, es decir que se presentó un incremento del 3% al incrementar la entrada en un 16% (De 24 puntos a 28 puntos) Con lo cual se concluye que el método de spline presenta un rendimiento aceptable

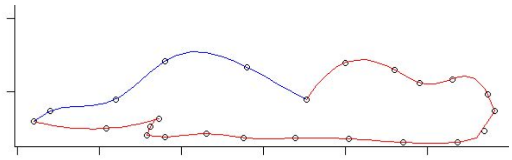
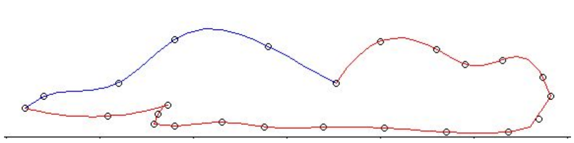
```
> lines(spline(a2, ay2, n = 20), col = "red")
>
> proc.time()-t # Inicia el cronómetro
   user  system elapsed
   0.08    0.01    0.31
> # NUESTRO CODIGO
>
> nuevo_x = c(16, 7, 27,63, 11,38)
```

Preguntas

A. El origen se puede modificar?
   Si es posible modificar el origen, dado que se tiene una figura cerrada sin importar el punto de comienzo de los datos el resultado producido será el mismo.
   Dado que en la implementación utilizada se crean los splines de forma independiente entre sí, la creación de un spline no incide en la formación de los demás.

```
x=c(1,2,6,9,14,17.6)
y=c(3,3.7,4.5,7.12,6.7,4.45)
```

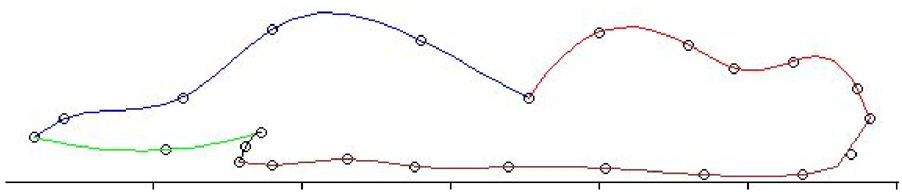
```
x=c(17.6, 14, 9, 6 ,2 ,1)
y=c(4.45, 6.7, 7.12,4.5, 3.7 ,3)
```

Se invirtió el orden del spline en color azul, tal como se puede observar se produce el mismo resultado en ambos casos.

Juan Castañeda
David Vanegas
Daniel Beltrán

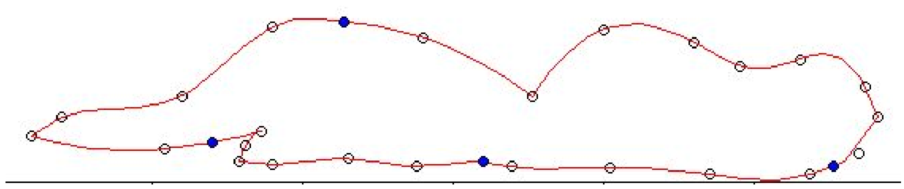
B. Si tenemos nueva información cómo podemos implementar esa información en el algoritmo de interpolación?



En la imagen anterior cada spline está representado con un color diferente. Con lo cual podemos concluir que al insertar un nuevo nodo resulta necesario recalcular el spline que va a contener al nuevo nodo. Está solución no es totalmente eficiente, sin embargo, resulta más viable que tener que recalcular nuevamente todos los splines.

C. Su método es robusto, en el sentido que si se tienen más puntos la exactitud no disminuye?
Dada la restricción de que para ingresar un nuevo punto el spline debe ser recalculado, la naturaleza del spline cúbico corresponderá a que la exactitud incrementara si se agregan más valores. Es decir, que a cambio de sacrificar eficiencia por tener que recalcular nuevamente algún spline, se garantiza que la solución se mantendrá robusta.
Por ejemplo si se añaden los siguientes puntos

x	y
16	2
7	2.75
27.63	1.88
11.38	7.29



Juan Castañeda
David Vanegas
Daniel Beltrán
Como se puede observar en la imagen los nuevos puntos están marcados con color azul. El resultado de agregarlos es que permiten definir con mayor precisión la imagen. Con lo cual se sustenta la afirmación de que se está utilizando un método robusto.

D. Suponga que tiene más puntos con más cifras significativas como se comporta su algoritmo ? la exactitud decae?

No dado que la naturaleza del algoritmo consiste en que a mayor número de puntos el resultado obtenido es más exacto. Siempre y cuando los puntos introducidos no generen cambios abruptos en el comportamiento de los splines. Es decir, que no se debe solapar con puntos previamente ingresados.