

Reto Mano

Analisis Numerico 2019-1

Por Carlos Restrepo

En que consiste el reto:

Construir un Interpolador (no necesariamente en forma polinomial) utilizando la menor cantidad de puntos k (parte superior y/o inferior o en total) y reproducir el dibujo del contorno completo del perrito sin bigotes (mejor exactitud) con la información dada :

Criterios:

1. Metodología que explique como 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)
2. Algoritmo que se aplico (justificación) aplico por ejemplo, interpolación polinomial y como soluciono el sistema
3. Validación del resultado

Productos

1. Algoritmo, requerimientos, codificación
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 mínimo para la curva).
 - Calcular la cota de error de su método con los datos experimentales y comparela con la cota teórica.
3. Tabla donde estén 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.
 - Cree una función que cuente el número aciertos y el número de diferencias en una cifra entre su función de interpolación y los originales y implementelo como el índice de Jaccard.
4. Cree una función que muestre la eficiencia de su método
5. Preguntas?
 - El origen se puede modificar? NO
 - Si tenemos nueva información o sea nodos como podemos implementar esa información en el algoritmo de interpolación? Anexar la información y analizar el nuevo rango para evaluar los puntos para interpolar.

- Su método es robusto, en el sentido que si se tienen más puntos la exactitud no disminuye? Se tiene un tope para trabajar, al aumentar los puntos se presentarían errores que pueden afectar.
- Suponga que tiene más puntos con más cifras significativas como se comporta su algoritmo? La exactitud decae? Depende, la cantidad de datos y los números ingresados, sería observar y ver a qué nivel se presentaría la falla.

```
require(pracma)
```

```
## Loading required package: pracma
```

```
xi=c(14.65, 14.71, 14.6, 14.8, 15.2, 15.6, 15.7, 17.0, 17.6, 17.52, 17.3, 16.8, 15.4, 14.83, 14.4, 14.5, 15.0, 15.1, 15.0, 14.9, 14.6, 14.3, 14.0, 13.9, 13.8, 13.5, 13.1, 13.0, 13.3, 13.2, 13.1, 12.9, 12.4, 11.9, 11.7, 11.6, 11.3, 10.9, 10.7, 10.6, 10.6, 10.1, 9.7, 9.4, 9.3, 9.6, 9.9, 10.1, 10.2, 10.3, 9.10, 8.6, 7.5, 7.0, 6.7, 6.6, 7.7 0, 8.00, 8.10, 8.40, 9.20, 9.30, 10, 10.2, 10.3, 10.0, 9.50)
yi=c(14.7, 14.33, 13.4, 12.33, 11.0, 10.5, 10.22, 8.20, 7.10, 6.70, 6.60, 6.80, 8.30, 8.80, 9.30, 8.80, 6.30, 5.5 0, 5.00, 4.70, 4.60, 4.50, 4.90, 5.40, 5.80, 6.90, 8.20, 7.60, 5.80, 4.50, 4.30, 3.90, 4.20, 5.70, 7.00, 7.90, 8. 20, 7.30, 6.70, 5.50, 5.10, 4.60, 4.7, 5.0, 5.5, 7.2, 7.8, 8.60, 9.40, 10.0, 10.7, 9.9, 9.0, 9.1, 9.3, 9.7, 11.7, 12.3, 12.5, 13.0, 13.91, 14.9, 16, 16.4, 16.8, 10.7, 11.0)
```

Selección de Puntos para Interpolación

Gráfica de los puntos sin Interpolación

```
print.table(xi)
```

```
## [1] 14.65 14.71 14.60 14.80 15.20 15.60 15.70 17.00 17.60 17.52 17.30
## [12] 16.80 15.40 14.83 14.40 14.50 15.00 15.10 15.00 14.90 14.60 14.30
## [23] 14.00 13.90 13.80 13.50 13.10 13.00 13.30 13.20 13.10 12.90 12.40
## [34] 11.90 11.70 11.60 11.30 10.90 10.70 10.60 10.60 10.10 9.70 9.40
## [45] 9.30 9.60 9.90 10.10 10.20 10.30 9.10 8.60 7.50 7.00 6.70
## [56] 6.60 7.70 8.00 8.10 8.40 9.20 9.30 10.00 10.20 10.30 10.00
## [67] 9.50
```

```
print.table(yi)
```

```
## [1] 14.70 14.33 13.40 12.33 11.00 10.50 10.22 8.20 7.10 6.70 6.60
## [12] 6.80 8.30 8.80 9.30 8.80 6.30 5.50 5.00 4.70 4.60 4.50
## [23] 4.90 5.40 5.80 6.90 8.20 7.60 5.80 4.50 4.30 3.90 4.20
## [34] 5.70 7.00 7.90 8.20 7.30 6.70 5.50 5.10 4.60 4.70 5.00
## [45] 5.50 7.20 7.80 8.60 9.40 10.00 10.70 9.90 9.00 9.10 9.30
## [56] 9.70 11.70 12.30 12.50 13.00 13.91 14.90 16.00 16.40 16.80 10.70
## [67] 11.00
```

Medias

Media #datos	Media X	Media Y
34	12.1	8.5

Regiones a trabajar

Regiones a trabajar	Palma	Meñique	Anular	Medio	Indice	Pulgar
3	1-6 y 60-65	6-15	15-28	28-36	36-50	50-60

Numero de rangos de interpolacion

+ (29)

Algoritmo Usado

- Metodo de la forma baricéntrica de Lagrange:
- Recibe el intervalo de nodos para realizar la interpolación
- La forma de Lagrange del polinomio interpolante es atractiva para propósitos teóricos. Sin embargo se puede re-escribir en una forma que se vuelva eficiente para el cálculo computacional además de ser numéricamente mucho más estable. La forma modificada y la forma baricéntrica de Lagrange son útiles cuando queremos interpolar una función en todo un intervalo con un con un polinomio interpolante.
- 1. Paquete Pracma

- - Este paquete proporciona implementaciones en R de funciones más avanzadas en análisis numérico, con una vista especial sobre rutinas de optimización y series de tiempo.
- **2. Seq**
 - - La función `seq ()` en R genera una secuencia de números.
 - - `seq(from, to, by, length.out)`
 - - desde,hasta, rango, tamaño de secuencia
- **3. Barylag**
 - - Barylag interpola los datos dados utilizando la fórmula de interpolación de Lagrange baricéntrica (vectorizada para eliminar todos los bucles).
 - - `barylag(xi, yi, x)`
 - - xi. y (yi) Cordenadas para los nodos
 - - x. Puntos de Interpolacion

Cotas:

Cota teoricas:

X	YProf	YLibro
5	3.9	3.9
10	7.12	7.1
20	7	7
26	5.15	5.2
28	4.3	4.3

Resultados graficos para datos profesora

```
#Gráfica de los puntos
plot(xi,yi, pch=7, cex=0.5, col = "red", asp=1,xlab="X", ylab="Y", main="Mano")

DibujarLinea<-function(inicio, final){
  xi0 = xi[inicio:final]
  yi0 = yi[inicio:final]
  x0 <- seq(xi[inicio], xi[final], len=20)
  y0 <- barylag(xi0, yi0, x0)
  print(x0)
  lines(x0, y0, col="blue")
}

#Borde Mano
DibujarLinea (1,4)
```

```
## [1] 14.65000 14.65789 14.66579 14.67368 14.68158 14.68947 14.69737
## [8] 14.70526 14.71316 14.72105 14.72895 14.73684 14.74474 14.75263
## [15] 14.76053 14.76842 14.77632 14.78421 14.79211 14.80000
```

```
DibujarLinea (4,6)
```

```
## [1] 14.80000 14.84211 14.88421 14.92632 14.96842 15.01053 15.05263
## [8] 15.09474 15.13684 15.17895 15.22105 15.26316 15.30526 15.34737
## [15] 15.38947 15.43158 15.47368 15.51579 15.55789 15.60000
```

```
#Dedo der
DibujarLinea (6,8)
```

```
## [1] 15.60000 15.67368 15.74737 15.82105 15.89474 15.96842 16.04211
## [8] 16.11579 16.18947 16.26316 16.33684 16.41053 16.48421 16.55789
## [15] 16.63158 16.70526 16.77895 16.85263 16.92632 17.00000
```

```
DibujarLinea (8,9)
```

```
## [1] 17.00000 17.03158 17.06316 17.09474 17.12632 17.15789 17.18947
## [8] 17.22105 17.25263 17.28421 17.31579 17.34737 17.37895 17.41053
## [15] 17.44211 17.47368 17.50526 17.53684 17.56842 17.60000
```

```
DibujarLinea (9,11)
```

```
## [1] 17.60000 17.58421 17.56842 17.55263 17.53684 17.52105 17.50526
## [8] 17.48947 17.47368 17.45789 17.44211 17.42632 17.41053 17.39474
## [15] 17.37895 17.36316 17.34737 17.33158 17.31579 17.30000
```

```
DibujarLinea (11,13)
```

```
## [1] 17.3 17.2 17.1 17.0 16.9 16.8 16.7 16.6 16.5 16.4 16.3 16.2 16.1 16.0
## [15] 15.9 15.8 15.7 15.6 15.5 15.4
```

```
DibujarLinea (13,15)
```

```
## [1] 15.40000 15.34737 15.29474 15.24211 15.18947 15.13684 15.08421
## [8] 15.03158 14.97895 14.92632 14.87368 14.82105 14.76842 14.71579
## [15] 14.66316 14.61053 14.55789 14.50526 14.45263 14.40000
```

```
#Dedo Medio der
DibujarLinea (15,18)
```

```
## [1] 14.40000 14.43684 14.47368 14.51053 14.54737 14.58421 14.62105
## [8] 14.65789 14.69474 14.73158 14.76842 14.80526 14.84211 14.87895
## [15] 14.91579 14.95263 14.98947 15.02632 15.06316 15.10000
```

```
DibujarLinea (18,24)
```

```
## [1] 15.10000 15.03684 14.97368 14.91053 14.84737 14.78421 14.72105
## [8] 14.65789 14.59474 14.53158 14.46842 14.40526 14.34211 14.27895
## [15] 14.21579 14.15263 14.08947 14.02632 13.96316 13.90000
```

DibujarLinea (24,28)

```
## [1] 13.90000 13.85263 13.80526 13.75789 13.71053 13.66316 13.61579
## [8] 13.56842 13.52105 13.47368 13.42632 13.37895 13.33158 13.28421
## [15] 13.23684 13.18947 13.14211 13.09474 13.04737 13.00000
```

#Dedo Medio Iz
DibujarLinea (28,29)

```
## [1] 13.00000 13.01579 13.03158 13.04737 13.06316 13.07895 13.09474
## [8] 13.11053 13.12632 13.14211 13.15789 13.17368 13.18947 13.20526
## [15] 13.22105 13.23684 13.25263 13.26842 13.28421 13.30000
```

DibujarLinea (29,31)

```
## [1] 13.30000 13.28947 13.27895 13.26842 13.25789 13.24737 13.23684
## [8] 13.22632 13.21579 13.20526 13.19474 13.18421 13.17368 13.16316
## [15] 13.15263 13.14211 13.13158 13.12105 13.11053 13.10000
```

DibujarLinea (31,36)

```
## [1] 13.10000 13.02105 12.94211 12.86316 12.78421 12.70526 12.62632
## [8] 12.54737 12.46842 12.38947 12.31053 12.23158 12.15263 12.07368
## [15] 11.99474 11.91579 11.83684 11.75789 11.67895 11.60000
```

Dedo Iz
DibujarLinea (36,38)

```
## [1] 11.60000 11.56316 11.52632 11.48947 11.45263 11.41579 11.37895
## [8] 11.34211 11.30526 11.26842 11.23158 11.19474 11.15789 11.12105
## [15] 11.08421 11.04737 11.01053 10.97368 10.93684 10.90000
```

DibujarLinea (38,40)

```
## [1] 10.90000 10.88421 10.86842 10.85263 10.83684 10.82105 10.80526
## [8] 10.78947 10.77368 10.75789 10.74211 10.72632 10.71053 10.69474
## [15] 10.67895 10.66316 10.64737 10.63158 10.61579 10.60000
```

DibujarLinea (40,41)

```
## [1] 10.6 10.6 10.6 10.6 10.6 10.6 10.6 10.6 10.6 10.6 10.6 10.6 10.6 10.6
## [15] 10.6 10.6 10.6 10.6 10.6 10.6
```

DibujarLinea (41,42)

```
## [1] 10.60000 10.57368 10.54737 10.52105 10.49474 10.46842 10.44211
## [8] 10.41579 10.38947 10.36316 10.33684 10.31053 10.28421 10.25789
## [15] 10.23158 10.20526 10.17895 10.15263 10.12632 10.10000
```

DibujarLinea (42,44)

```
## [1] 10.100000 10.063158 10.026316 9.989474 9.952632 9.915789 9.878947
## [8] 9.842105 9.805263 9.768421 9.731579 9.694737 9.657895 9.621053
## [15] 9.584211 9.547368 9.510526 9.473684 9.436842 9.400000
```

DibujarLinea (44,46)

```
## [1] 9.400000 9.410526 9.421053 9.431579 9.442105 9.452632 9.463158
## [8] 9.473684 9.484211 9.494737 9.505263 9.515789 9.526316 9.536842
## [15] 9.547368 9.557895 9.568421 9.578947 9.589474 9.600000
```


DibujarLinea (46,48)

```
## [1] 9.600000 9.626316 9.652632 9.678947 9.705263 9.731579 9.757895
## [8] 9.784211 9.810526 9.836842 9.863158 9.889474 9.915789 9.942105
## [15] 9.968421 9.994737 10.021053 10.047368 10.073684 10.100000
```

DibujarLinea (48,50)

```
## [1] 10.10000 10.11053 10.12105 10.13158 10.14211 10.15263 10.16316
## [8] 10.17368 10.18421 10.19474 10.20526 10.21579 10.22632 10.23684
## [15] 10.24737 10.25789 10.26842 10.27895 10.28947 10.30000
```

#Dedo Pulgar

DibujarLinea (50,52)

```
## [1] 10.300000 10.210526 10.121053 10.031579 9.942105 9.852632 9.763158
## [8] 9.673684 9.584211 9.494737 9.405263 9.315789 9.226316 9.136842
## [15] 9.047368 8.957895 8.868421 8.778947 8.689474 8.600000
```

DibujarLinea (52,54)

```
## [1] 8.600000 8.515789 8.431579 8.347368 8.263158 8.178947 8.094737
## [8] 8.010526 7.926316 7.842105 7.757895 7.673684 7.589474 7.505263
## [15] 7.421053 7.336842 7.252632 7.168421 7.084211 7.000000
```

DibujarLinea (54,56)

```
## [1] 7.000000 6.978947 6.957895 6.936842 6.915789 6.894737 6.873684
## [8] 6.852632 6.831579 6.810526 6.789474 6.768421 6.747368 6.726316
## [15] 6.705263 6.684211 6.663158 6.642105 6.621053 6.600000
```

DibujarLinea (56,58)

```
## [1] 6.600000 6.673684 6.747368 6.821053 6.894737 6.968421 7.042105
## [8] 7.115789 7.189474 7.263158 7.336842 7.410526 7.484211 7.557895
## [15] 7.631579 7.705263 7.778947 7.852632 7.926316 8.000000
```

DibujarLinea (58,60)

```
## [1] 8.000000 8.021053 8.042105 8.063158 8.084211 8.105263 8.126316
## [8] 8.147368 8.168421 8.189474 8.210526 8.231579 8.252632 8.273684
## [15] 8.294737 8.315789 8.336842 8.357895 8.378947 8.400000
```

#Borde Mano

DibujarLinea (60,61)

```
## [1] 8.400000 8.442105 8.484211 8.526316 8.568421 8.610526 8.652632
## [8] 8.694737 8.736842 8.778947 8.821053 8.863158 8.905263 8.947368
## [15] 8.989474 9.031579 9.073684 9.115789 9.157895 9.200000
```

DibujarLinea (61,62)

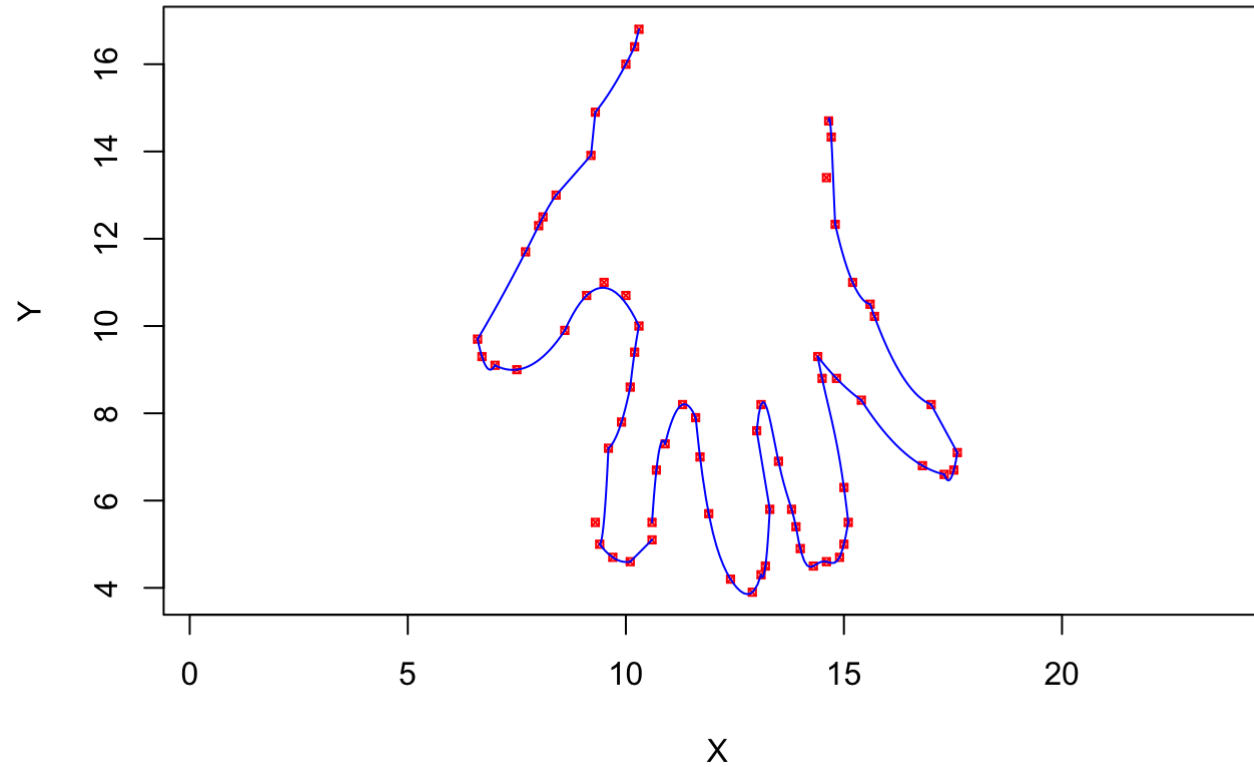
```
## [1] 9.200000 9.205263 9.210526 9.215789 9.221053 9.226316 9.231579
## [8] 9.236842 9.242105 9.247368 9.252632 9.257895 9.263158 9.268421
## [15] 9.273684 9.278947 9.284211 9.289474 9.294737 9.300000
```

DibujarLinea (62,64)

```
## [1] 9.300000 9.347368 9.394737 9.442105 9.489474 9.536842 9.584211
## [8] 9.631579 9.678947 9.726316 9.773684 9.821053 9.868421 9.915789
## [15] 9.963158 10.010526 10.057895 10.105263 10.152632 10.200000
```

DibujarLinea (64,65)

Mano



```
## [1] 10.20000 10.20526 10.21053 10.21579 10.22105 10.22632 10.23158  
## [8] 10.23684 10.24211 10.24737 10.25263 10.25789 10.26316 10.26842  
## [15] 10.27368 10.27895 10.28421 10.28947 10.29474 10.30000
```

Error Relativo

```
errorx <- 0.01*abs(xi)
varianciax <- var(xi)
mediax <- mean(xi)
Ex.abs <- sum(abs(2*xi-2*sum(xi)/67)*errorx)/(67-1)
Ex.rel <- Ex.abs/varianciax
print.table(Ex.rel)
```

```
## [1] 0.0731685
```

```
errory <- 0.01*abs(yi)
varianciay <- var(yi)
mediay <- mean(yi)
Ey.abs <- sum(abs(2*yi-2*sum(yi)/67)*errory)/(67-1)
Ey.rel <- Ey.abs/varianciay
print.table(Ey.rel)
```

```
## [1] 0.04612405
```

1. Error Relativo para X del 7.3%

2. Error Relativo para Y del 4.6%

<http://disi.unal.edu.co/~lctorress/MetNum/MeNuCI03.pdf> (<http://disi.unal.edu.co/~lctorress/MetNum/MeNuCI03.pdf>)

https://fac.ksu.edu.sa/sites/default/files/numerical_analysis_9th.pdf
(https://fac.ksu.edu.sa/sites/default/files/numerical_analysis_9th.pdf)

http://numat.net/tutor/error_interp.pdf (http://numat.net/tutor/error_interp.pdf)

<https://w3.ual.es/~aposadas/TeoriaErrores.pdf> (<https://w3.ual.es/~aposadas/TeoriaErrores.pdf>)

<http://www.sc.ehu.es/nmwmigaj/instrum.htm> (<http://www.sc.ehu.es/nmwmigaj/instrum.htm>)

https://tarwi.lamolina.edu.pe/~fmendiburu/index-filer/academic/script_numerico.htm
(https://tarwi.lamolina.edu.pe/~fmendiburu/index-filer/academic/script_numerico.htm)

Carlos Manuel Restrepo Riveros

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.