Programación: Polinomio interpolante de Lagrange

Objetivos. Programar el cálculo del polinomio interpolante con la fórmula de Lagrange.

Requisitos. Multiplicación de un polinomio por un binomio, evaluación de un polinomio en un punto.

Funciones auxiliares para trabajar con polinomios

1. Guardar polinomios como listas de coeficientes. Guardamos polinomios como listas de coeficientes empezando con el coeficiente constante. Por ejemplo, guardamos $x^2 - 3x + 5$ como la lista $5, -3, 1$.

2. Evaluación de un polinomio en un punto. En la primera unidad del curso escribimos la función PolEval de dos argumentos `pcoefs` y x que calcula el valor del polinomio con coeficientes `pcoefs` en el punto x.

3. Prueba. Consideremos el polinomio $f(x) = x^2 - 5x + 2$. Es fácil ver que $f(-2) = 16$, por eso el siguiente comando debe regresar 16.

```
PolEval[{2, -5, 1}, -2]
```

4. Problema PolValues (1 %). Escriba una función que calcule los valores de un polinomio (dado por su lista de coeficientes) en una lista de puntos.

> Entrada: la lista `pcoefs` de los coeficientes del polinomio, la lista `points` de los puntos.
>
> Salida: la lista de valores.

Indicación: puede usar las funciones `PolEval`, `Table` y `Length`.

5. Prueba. Consideremos el polinomio $f(x) = x^2 - 5x + 8$. Se puede ver que

$$f(-1) = 14, \quad f(1) = 4, \quad f(0) = 8, \quad f(3) = 2.$$

Por eso el comando

```
PolValues[{8, -5, 1}, {-1, 1, 0, 3}]
```

tendrá que regresar la lista $14, 4, 8, 2$.

6. Multiplicación de un polinomio por un binomio. En la primera unidad del curso escribimos la función MulPolBinom de dos argumentos $\mathtt{pcoefs}$ y b que multiplica el polinomio con coeficientes $\mathtt{pcoefs}$ por el binomio $x+b$ y regresa la lista de los coeficientes del producto.

7. Prueba. `MulPolBinom[{5, 4, -1, 3}, 2]` tiene que regresar `{10, 13, 2, 5, 3}`, pues

$$(3x^3 - x^2 + 4x + 5)(x + 2) = 3x^4 + 5x^3 + 2x^2 + 13x + 10.$$

8. Ejercicio auxiliar: ciclo sobre $j \in \{1, \ldots, n\}$, $j \neq k$. Para programar el cálculo de los polinomios básicos de Lagrange hay que aprender cómo organizar un ciclo sobre un conjunto de índices de la forma $\{1, \ldots, n\} \setminus \{k\}$, por ejemplo un ciclo sobre $\{1, 2, 4, 5, 6, 7\}$. Hay por lo menos dos maneras de hacerlo.

1. Escribir un ciclo `For` con una condición `If`:

   ```
   For[j = 1, j <= 7, j++, If[j != 3, Print[j]]]
   ```

2. Escribir dos ciclos `For`, el primero sobre $\{1, \ldots, k-1\}$ y el segundo sobre $\{k+1, \ldots, n\}$:

   ```
   For[j = 1, j <= 2, j++, Print[j]]
   For[j = 4, j <= 7, j++, Print[j]]
   ```

9. Operaciones lineales con vectores (repaso). Nosotros guardamos polinomios como listas de coeficientes y podemos usar las operaciones internas de Wolfram Mathematica para sumar estas listas o multiplicarlas por escalares:

```
fcoefs = {10, 20, 30}; gcoefs = {4, 5, 6};
fcoefs + gcoefs
3 * fcoefs
fcoefs / 2
```

Polinomio básico de Lagrange

Fórmula para calcular el polinomio básico de Lagrange. Sean $x_1, \ldots, x_n$ algunos puntos diferentes por pares y sea $k \in \{1, \ldots, n\}$. Entonces el siguiente polinomio toma valor 1 en el punto x_k y valor 0 en los demás puntos x_j, $j \neq k$:

$$L_k(x) = \prod_{\substack{1 \le j \le n \\ j \neq k}} \frac{x - x_j}{x_k - x_j} = \frac{1}{\prod\limits_{\substack{1 \le j \le n \\ j \neq k}} (x_k - x_j)} \cdot \prod_{\substack{1 \le j \le n \\ j \neq k}} (x - x_j). \qquad (1)$$

10. Cálculo del polinomio básico de Lagrange para $n = 4$ y $k = 3$.
Notemos que el numerador de la fórmula (1) es un polinomio dado como un producto de binomios, y el denominador es un producto de números. Si $n = 4$ y $k = 3$, entonces el polinomio (1) se puede calcular de la siguiente manera (llene los espacios):

$$P(x) := \qquad; \qquad \mu := 1;$$

$$P(x) := P(x) \cdot (x - \qquad); \qquad \mu := \mu \cdot (\qquad\qquad);$$

$$P(x) := P(x) \cdot (x - \qquad); \qquad \mu := \mu \cdot (\qquad\qquad);$$

$$P(x) := P(x) \cdot (x - \qquad); \qquad \mu := \mu \cdot (\qquad\qquad);$$

$$P(x) := P(x)/\mu.$$

El valor inicial de $P(x)$ debe ser el polinomio que hace el papel del "elemento neutro bajo la multiplicación".

Sugerencias para realizar el algoritmo.

- En lugar del polinomio $P(x)$ trabajar con su lista de coeficientes `pcoefs`.
- Para multiplicar un polinomio por el binomio $(x - x_j)$ usar la función `MulPolBinom`.
- La división de una lista entre un escalar ya está realizada en Wolfram Mathematica. Por ejemplo, `{10,15,-20} / 5` regresa `{2,3,-4}`.

11. Problema LagrangeBasic (2 %). Escriba una función de dos argumentos `points` y `k` que calcule la lista de los coeficientes del polinomio básico de Lagrange L_k correspondiente a los puntos `points` y el índice `k`.

```
Entrada: points, k;
Variables locales: ???;
pcoefs := ???;
mu := ???;
n := longitud(points);
Para cada j de 1 a n excepto k:
    pcoefs := MulPolBinom(coefs, ???);
    mu := mu * (points[???] - points[???]);
pcoefs := pcoefs / mu;
Salida: pcoefs.
```

12. Comprobación.

`pcoefs = LagrangeBasic[{-1, 2, 4}, 3]`

`PolValues[pcoefs, {-1, 2, 4}]` (tiene que regresar `0, 0, 1`)

Cálculo del polinomio interpolante con la fórmula de Lagrange

13. Fórmula de Lagrange para el polinomio interpolante. Sean $x_1, \ldots, x_n$ números diferentes por pares y sean $y_1, \ldots, y_n$ algunos números. Para cada $k \in \{1, \ldots, n\}$ denotemos por $L_k(x)$ al polinomio básico de Lagrange que vale 1 en x_k y vale 0 en los demás puntos x_i. Entonces el polinomio

$$P(x) = \sum_{k=1}^{n} y_k L_k(x).$$

es de grado $\leq n-1$ y toma valor y_i en el punto x_i para cada $i \in \{1, \ldots, n\}$.

14. Sugerencias para realizar el algoritmo. Wolfram Mathematica trata listas como vectores y sabe hacer operaciones lineales:

`4 * {3, -5, 1, 7}` regresa `{12,-20,4,28}`

`{-5, 3, 10, 2} + {3, 30, 1, -7}` regresa `{-2, 33, 11, -5}`

15. Problema LagrangeInterpol (2 %). Escriba una función que calcule los coeficientes del polinomio de grado mínimo que tome valores dados en puntos dados.

Entrada: la lista points de los puntos, la lista values de los valores; se supone que estas dos listas son de la misma longitud y que los elementos de la lista points son diferentes a pares.

Salida: la lista de los coeficientes del polinomio interpolante.

16. Prueba.

```
points = {2, 3, -1, 5}; values = {3, 11, -9, 69};

fcoefs = LagrangeInterpol[points, values]

PolEval[fcoefs, -1]

PolValues[fcoefs, points]
```