Overview

- Function approximation criteria: interpolation, least squares, min-max
- Bases for polynomial interpolation
- Polynomial interpolant forms
  - Lagrange
  - Barycentric Lagrange

- Consider $f: \mathbb{R} \to \mathbb{R}$, "difficult" to compute, and known through a sample

$$\mathcal{D} = \{(x_i, y_i), i = 0, 1, 2, \ldots, m\}, y_i = f(x_i), i \neq j \Rightarrow x_i \neq x_j.$$

- Introduce *approximant* $g: \mathbb{R} \to \mathbb{R}$, "easy" to compute, $z_i = g(x_i)$
- Introduce vectors: $\boldsymbol{x} = [\ x_0 \ \ldots \ x_m\ ]^T$, $\boldsymbol{y} = [\ y_0 \ \ldots \ y_m\ ]^T$,$\boldsymbol{z} = [\ z_0 \ \ldots \ z_m\ ]^T$
- Assess accuracy of approximation through norm of error vector $\delta = \|\boldsymbol{z} - \boldsymbol{y}\|$
- Various types of norms and error bounds can be considered:
  - $\delta = 0 \Rightarrow \boldsymbol{z} = \boldsymbol{y} \Rightarrow g(x_i) = y_i, i = 0, \ldots, m$ is known as *interpolation*
  - Assume g is defined by $n+1$ parameters, $\boldsymbol{c} = [\ c_0 \ \ldots \ c_n\ ]^T$, $z_i = g(x_i, \boldsymbol{c})$. The approximant obtained by minimization of the two-norm is known as the *least squares approximant*

$$\min_{\boldsymbol{c}} \|\boldsymbol{z} - \boldsymbol{y}\|_2 \Leftrightarrow \min_{\boldsymbol{c}} \Sigma_{i=0}^{m} (z_i - y_i)^2$$

  - The approximant obtained by the minimization of the inf-norm is known as the *minmax approximant*

$$\min_{\boldsymbol{c}} \|\boldsymbol{z} - \boldsymbol{y}\|_\infty \Leftrightarrow \min_{\boldsymbol{c}} \max_{0 \leqslant i \leqslant m} |z_i - y_i|, i \in \mathbb{N}.$$

- Consider a set of linearly independent functions $\{g_0(t), g_1(t), \ldots, g_n(t)\}$

$$g(t) = c_0\, g_0(t) + c_1\, g_1(t) + \cdots + c_n g_n(t) = 0 \Rightarrow c_0 = c_1 = \cdots = c_n = 0$$

- Construct approximation of $f(t)$ by

$$f(t) \cong g(t) = c_0\, g_0(t) + c_1\, g_1(t) + \cdots + c_n g_n(t) = \sum_{j=0}^{n} c_j\, g_j(t)$$

- Even though $f(t), g_0(t), \ldots, g_n(t)$ might be nonlinear the above expresses f as a *linear combination* of $\{g_0(t), g_1(t), \ldots, g_n(t)\}$
- Various choices can be made for the basis functions $\{g_0(t), g_1(t), \ldots, g_n(t)\}$

- The interpolation conditions $g(x_i) = y_i$ can be achieved if
  - $n = m$
  - $g_i(x_j) = \delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$
- In this case the linear combination coefficients a_i are simply the function values

$$g(x_j) = \sum_{i=0}^{m} c_i\, g_i(x_j) = \sum_{i=0}^{m} c_i\, \delta_{ij} = c_i = y_i, i = 0, 1, \ldots, m$$

- Construct polynomial of degree m to satisfy above conditions

$$l_i(t) = \prod_{j=0, j\neq i}^{m} (t - x_j) \equiv \prod_{j=0}^{m}{}' (t - x_j), \ell_i(t) = \frac{\prod_{j=0}^{m}{}' (t - x_j)}{\prod_{j=0}^{m}{}' (x_i - x_j)}$$

$$l_i(x_i) = \prod_{j=0, j\neq i}^{m} (x_i - x_j), \ell_i(t) = \frac{l_i(t)}{l_i(x_i)}, \ell_i(x_j) = \delta_{ij}$$

- $\ell_i(t)$ are the *Lagrange basis* polynomials, $l_i(t)$ are the unnormalized version

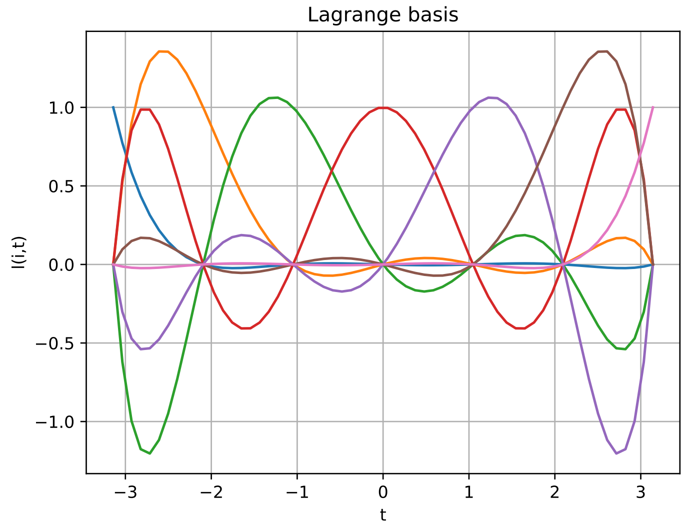

Figure 1. Lagrange basis for $n = 6$ equidistant subintervals over interval $[-\pi, \pi]$

- Note that at each $x_j = -\pi + 2\pi j / n$ all polynomials except one evaluate as 0. The remaining polynomial evaluates as 1.
- The polynomials can reach values greater or less than 1

Algorithm (Lagrange evaluation)

Input: $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^{n+1}, t \in \mathbb{R}$
Output: $p(t) = \sum_{i=0}^{n} y_i \prod_{j=0}^{n}{}' (t - x_j)/(x_i - x_j)$
$p = 0$
for $i = 0$ to n
    $w = 1$
    for $j = 0$ to n
        if $j \neq i$ then $w = w\,(t - x_j)/(x_i - x_j)$
    end
    $p = p + w \cdot y_i$
end
return p

- Operation count: for each i from 0 to n, for each j from 0 to n skipping one index, carry out 2 FLOPS (1 FLOP = 1 addition and 1 multiplication)

$$2n(n-1) = \mathcal{O}(2n^2)\ \text{FLOPs}$$

- The $\mathcal{O}(2n^2)$ operation count for evaluating the Lagrange interpolant can be reduced to $\mathcal{O}(2n)$ by using a different form, the *barycentric form*.
- Let $w(t) = \prod_{k=0}^{n} (t - x_k)$ and rewrite $\ell_i(t)$ as

$$\ell_i(t) = \prod_{j=0}^{n}{}' \frac{t - x_j}{x_i - x_j} = w(t) \frac{w_i}{t - x_i}, \; w_i = \prod_{j=0}^{n}{}' \frac{1}{x_i - x_j}.$$

- The interpolating polynomial is now

$$p(t) = \sum_{i=0}^{n} y_i \, \ell_i(t) = w(t) \sum_{i=0}^{n} y_i \frac{w_i}{t - x_i}.$$

- Interpolation of the function $g(t) = 1$ gives $1 = w(t) \sum_{i=0}^{n} \frac{w_i}{t - x_i}$.
- Take ratio to obtain barycentric form

$$p(t) = \frac{\sum_{i=0}^{n} y_i \dfrac{w_i}{t - x_i}}{\sum_{i=0}^{n} \dfrac{w_i}{t - x_i}},$$

Algorithm (Barycentric Lagrange evaluation)

Input: $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^{n+1}, t \in \mathbb{R}$

Output: $p(t) = \left(\sum_{i=0}^{n} y_i \, \frac{w_i}{t - x_i} \right) / \left(\sum_{i=0}^{n} \frac{w_i}{t - x_i} \right)$

for $i = 0$ to n

  $w_i = 1$

 for $j = 0$ to n

  if $j \neq i$ $w_i = w_i / (x_i - x_j)$

 end

end

$q = r = 0$

for $i = 0$ to n

 $s = w_i / (t - x_i)$; $q = q + y_i \, s$; $r = r + s$

end

$p = q / r$

return p

- Precompute w_i, $\mathcal{O}(n^2)$ FLOPs. Evaluation for given t costs only $\mathcal{O}(2n)$ FLOPs

- **Algorithm (Barycentric Lagrange evaluation)**

Input: $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^{n+1}, t \in \mathbb{R}$

Output: $p(t) = \left(\sum_{i=0}^{n} y_i \frac{w_i}{t - x_i} \right) / \left(\sum_{i=0}^{n} \frac{w_i}{t - x_i} \right)$

for $i = 0$ to n
    $w_i = 1$
  for $j = 0$ to n
    if $j \neq i$ $w_i = w_i / (x_i - x_j)$
  end
end
$q = r = 0$
for $i = 0$ to n
   $s = w_i / (t - x_i)$; $q = q + y_i s$; $r = r + s$
end
$p = q / r$
return p

```
∴ function BaryLagrange(t,x,y)
    n=length(x)-1; w=ones(size(x));
    for i=1:n+1
      w[i]=1
      for j=1:n+1
        if (i!=j) w[i]=w[i]/(x[i]-x[j]); end
      end
    end
    q=r=0
    for i=1:n+1
      d=t-x[i]
      if d≈0 return y[i]; end
      s=w[i]/d; q=q+y[i]*s; r=r+s
    end
    return q/r
  end;
```

```
∴ p2(t)=3*t^2-2*t+1;
```

```
∴ x=[-2 0 2]; y=p2.(x);
```

```
∴ t=-3:3; [p2.(t) BaryLagrange.(t,Ref(x),Ref(y))]
```

```
∴
```

- Precompute w_i, $\mathcal{O}(n^2)$ FLOPs. Evaluation for given t costs only $\mathcal{O}(2n)$ FLOPs

- **Algorithm (Barycentric Lagrange evaluation)**

  Input: $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^{n+1}, t \in \mathbb{R}$

  Output: $p(t) = \left(\sum_{i=0}^{n} y_i \dfrac{w_i}{t - x_i} \right) / \left(\sum_{i=0}^{n} \dfrac{w_i}{t - x_i} \right)$

  for $i = 0$ to n
    $w_i = 1$
   for $j = 0$ to n
    if $j \neq i$ $w_i = w_i / (x_i - x_j)$
   end
  end

$q = r = 0$
for $i = 0$ to n
 $s = w_i/(t - x_i)$; $q = q + y_i s$; $r = r + s$
end
$p = q/r$
return p

```
∴ function BaryLagrange(t,x,y)
    n=length(x)-1; w=ones(size(x));
    for i=1:n+1
      w[i]=1
      for j=1:n+1
        if (i!=j) w[i]=w[i]/(x[i]-x[j]); end
      end
    end
    q=r=0
    for i=1:n+1
      d=t-x[i]
      if d≈0 return y[i]; end
      s=w[i]/d; q=q+y[i]*s; r=r+s
    end
    return q/r
  end;
```

```
∴ p2(t)=3*t^2-2*t+1;
```

```
∴ x=[-2 0 2]; y=p2.(x);
```

```
∴ t=-3:3; [p2.(t) BaryLagrange.(t,Ref(x),Ref(y))]
```

$$\begin{bmatrix} 34 & 33.99999999999999 \\ 17 & 17 \\ 6 & 6.0 \\ 1 & 1 \\ 2 & 2.0 \\ 9 & 9 \\ 22 & 21.999999999999996 \end{bmatrix} \tag{1}$$

```
∴
```

- Precompute w_i, $\mathcal{O}(n^2)$ FLOPs. Evaluation for given t costs only $\mathcal{O}(2n)$ FLOPs