Interpolare Lagrange baricentrică

March 22, 2012

1 Interpolare Lagrange baricentrică

Considerăm $n+1$ noduri de interpolare distincte x_j, $j = 0, \ldots, n$, date, împreună cu numerele f_j, care pot să fie valori ale unei funcţii f sau nu. Vom presupune că nodurile sunt reale, deşi cele mai multe dintre rezultate se extind şi în complex. Fie $\mathbb{P}_n$ spaţiul polinoamelor de grad cel mult n. Considerăm polinomul de interpolare Lagrange şi polinoamele fundamentale

$$p(x) = \sum_{j=0}^{n} f_j \ell_j(x), \qquad \ell_j(x) = \frac{\prod\limits_{\substack{k=0\\k\neq j}}^{n} (x - x_k)}{\prod\limits_{\substack{k=0\\k\neq j}}^{n} (x_j - x_k)}. \tag{1}$$

Polinomul fundamental ℓ_j corespunzător nodului x_j are proprietatea

$$\ell_j(x_k) = \begin{cases} 1, & j = k \\ 0, & \text{altfel} \end{cases} \qquad j, k = 0, \ldots, n. \tag{2}$$

Dezavantaje:

1. fiecare evaluare a lui $p(x)$ necesită $O(n^2)$ adunări şi înmulţiri;
2. adăugarea unei noi perechi de date (x_{n+1}, f_{n+1}) necesită reluarea tuturor calculelor.
3. procesul de calcul este numeric instabil.

Spre deosebire de metoda directă, metoda lui Newton necesită doar $O(n)$ flops pentru fiecare evaluare a lui p odată ce anumite numere, independente de punctul de evaluare x au fost calculate. Metoda lui Newton constă din doi paşi. La primul se calculează tabela de diferenţe divizate

$$
\begin{array}{llllll}
f[x_0] & f[x_0,x_1] & f[x_0,x_1,x_2] & f[x_0,x_1,x_2,x_3] & \cdots & f[x_0,x_1,x_2,\dots,x_n] \\
f[x_1] & f[x_1,x_2] & f[x_1,x_2,x_3] & f[x_1,x_2,x_3,x_4] & & \\
f[x_2] & f[x_2,x_3] & & & & \\
\vdots & & & & & \\
\vdots & f[x_{n-1},x_n] & & & & \\
f[x_n] & & & & &
\end{array}
$$

cu formula recursivă

$$f[x_j, x_{j+1}, \dots, x_{k-1}, x_k] = \frac{f[x_{j+1}, \dots, x_k] - f[x_j, \dots, x_{k-1}]}{x_k - x_j}$$

Calculul tabelei necesită aproximativ n^2 scăderi şi $n^2/2$ împărţiri. La al doilea pas se evaluează $p(x)$ cu formula de interpolare a lui Newton

$$\begin{aligned} p(x) = f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) + \dots \\ + f[x_0, x_1, x_2, \dots, x_n](x - x_0)(x - x_1) \dots (x - x_{n-1}). \end{aligned}$$

1.1 O formulă Lagrange îmbunătăţită

Vom rescrie formula (1) astfel ca ea să poată fi evaluată şi actualizată cu $O(n)$ operaţii. Introducând

$$\ell(x) = (x - x_0)(x - x_1) \cdots (x - x_n) \tag{4}$$

ℓ_j se poate scrie ca $\ell_j(x) = \ell(x)/(x - x_j)$. Definind ponderile baricentrice prin

$$w_j = \frac{1}{\prod\limits_{k \neq j} (x_j - x_k)}, \qquad j = 0, \dots, n, \tag{5}$$

adică, $w_j = 1/\ell'(x_j)$, putem scrie ℓ_j sub forma

$$\ell_j(x) = \ell(x) \frac{w_j}{x - x_j}.$$

Acum PIL se scrie

$$p(x) = \ell(x) \sum_{j=0}^{n} \frac{w_j}{x - x_j} f_j. \tag{6}$$

Avantajul este că putem calcula interpolantul Lagrange cu o formulă ce necesită $O(n^2)$ flops pentru calculul unor cantităţi independente de x, numerele w_j, urmate de $O(n)$ flops pentru evaluarea lui p, odată ce aceste numere sunt cunoscute.

Din (6) rezultă că actualizarea polinomului de interpolare la inserţia unui nod nou necesită următoarele calcule:

- se împarte fiecare w_j, $j = 0..n$, prim $x_j - x_{n+1}$ (un flop pentru fiecare punct), cu un cost de $n + 1$ flops;
- se calculează w_{n+1} cu formula (5) cu alte $n + 1$ flops.

Algoritmul de actualizare

Algorithm 1 Calcul ponderi baricentrice

$w_0^{(0)} := 1$;
for $j := 1$ **to** n **do**
    for $k := 0$ **to** $j - 1$ **do**
        $w_k^{(j)} := (x_k - x_j) w_k^{(j-1)}$;
    end for
    $w_j^{(j)} := \prod_{j=0}^{n-1} (x_j - x_k)$;
end for
for $j := 0$ **to** n **do**
    $w_j^{(j)} := 1/w_j^{(j)}$;
end for

1.2 Metoda baricentrică

Interpolând funcţia constantă 1 obţinem

$$1 = \sum_{j=0}^{n} \ell_j(x) = \ell(x) \sum_{j=0}^{n} \frac{w_j}{x - x_j.} \tag{7}$$

Împărţind (6) cu expresia de mai sus şi simplificând cu $\ell(x)$, obţinând

$$p(x) = \frac{\displaystyle\sum_{j=0}^{n} \frac{w_j}{x - x_j} f_j}{\displaystyle\sum_{j=0}^{n} \frac{w_j}{x - x_j}}, \tag{8}$$

numită formula baricentrică.

La fel ca în (6), în (8) se poate adăuga o nouă pereche de date (x_{n+1}, f_{n+1}) şi actualiza w_j în $O(n)$ flops.

1.3 Distribuţii remarcabile

În cazul unor noduri particulare se pot da formule explicite pentru ponderile baricentrice w_j. Pentru noduri echidistante în intervalul $[-1, 1]$, la distanţa $h = 2/n$, se obţine $w_j = (-1)^n \binom{n}{j} / (h^n n!)$, care după anularea factorilor independenţi de j ne dă

$$w_j = (-1)^j \binom{n}{j}. \tag{9}$$

Acelaşi rezultat se obţine şi pentru un interval arbitrar $[a, b]$, deoarece formula originală pentru w_j se înmulţeşte cu $2^n(b-a)^{-n}$, dar acest factor poate fi înlăturat.

Familia de *puncte Cebîşev* se poate obţine proiectând puncte egal spaţiate pe cercul unitate pe intervalul $[-1, 1]$. Pornind de la formula

$$w_j = \frac{1}{\ell'(x_j)}, \tag{10}$$

se pot obţine formule explicite pentru ponderile w_j.

Punctele Cebîşev de speţa I sunt date de

$$x_j = \cos\frac{(2j+1)\,\pi}{2n+2}, \qquad j = 0, \dots, n.$$

Anulând factorii independenţi de j se obţine

$$w_j = (-1)^j \sin\frac{(2j+1)\,\pi}{2n+2}. \tag{11}$$

Punctele Cebîşev de speţa II sunt date de

$$x_j = \cos\frac{j\pi}{n}, \qquad j = 0, \dots, n,$$

iar ponderile corespunzătoare sunt

$$w_j = (-1)^j\,\delta_j, \qquad \delta_j = \begin{cases} 1/2, & j = 0 \text{ sau } j = n, \\ 1, & \text{altfel.} \end{cases}$$

Dăm codul MATLAB pentru interpolarea Lagrange baricentrică

```
function ff=baryLagrange(x,y,xx)
%BARYLAGRANGE - barycentric Lagrange interpolation
%call ff=baryLagrange(x,y,xi)
%x -  nodes
%y - function values
%xx - interpolation points
%ff - values of interpolation polynomial


%compute weights
n=length(x)-1;
[X,Y]=meshgrid(x);
M=X-Y+eye(n+1);
c=1./prod(M)';
numer = zeros(size(xx));
denom = zeros(size(xx));
exact = zeros(size(xx));
```

```
for j=1:n+1
    xdiff = xx-x(j);
    temp = c(j)./xdiff;
    numer = numer+temp*y(j);
    denom = denom+temp;
    exact(xdiff==0) = 1;
end
ff = numer ./ denom;
jj = find(exact); ff(jj) = y(exact(jj));
```

În cazul nodurilor Cebîşev de speţa a doua sursa MATLAB este

```
function ff=ChebLagrange(y,xx,a,b)
%CHEBLAGRANGE - Lagrange interpolation for Chebyshev points- barycentric
%call ff=ChebLagrange(y,xx,a,b)
%y - function values;
%xx - evaluation points
%a,b - interval
%ff - values of Lagrange interpolation polynomial

n = length(y)-1;
if nargin==2
    a=-1; b=1;
end
c = [1/2; ones(n-1,1); 1/2].*(-1).^((0:n)');
x = sort(cos((0:n)'*pi/n))*(b-a)/2+(a+b)/2;
numer = zeros(size(xx));
denom = zeros(size(xx));
exact = zeros(size(xx));
for j=1:n+1
    xdiff = xx-x(j);
    temp = c(j)./xdiff;
    numer = numer+temp*y(j);
    denom = denom+temp;
    exact(xdiff==0) = 1;
end
ff = numer ./ denom;
jj = find(exact); ff(jj) = y(exact(jj));
```