Aproximace funkcí

1 Úvod

Aproximace funkce - výpočet funkčních hodnot nejbližší (v nějakém smyslu) funkce v určité třídě funkcí (funkce s nějakými neznámými parametry)

Příklady funkcí používaných pro aproximaci

- Polynom $\Phi_m(x) = c_0 + c_1 x + c_2 x^2 + \ldots + c_m x^m$.
- Zobecněný polynom

  $$\Phi_m(x) = c_0 g_0(x) + c_1 g_1(x) + c_2 g_2(x) + \ldots + c_m g_m(x),$$

  kde $g_0(x),\ \ldots,\ g_m(x)$ je systém m+1 lineárně nezávislých jednoduchých a dostatečně hladkých funkcí.
- Racionální lomená funkce

  $$\Phi_m(x) = \frac{a_0 + a_1 x + a_2 x^2 + \ldots + a_k x^k}{b_0 + b_1 x + b_2 x^2 + \ldots + b_l x^l} \qquad (m = k + l).$$
- Jiné.

Důvody aproximace - různorodé

- Příliš náročný výpočet funkce (složitý funkční předpis, implicitně zadané funkce, . . .)
- Potřeba výpočtu dalších charakteristik funkce (derivace, integrál, . . .)
- Analytické vyjádření není známo - funkce daná tabulkou hodnot (spočtených či naměřených)

Typy aproximací

- Interpolační aproximace (interpolace a extrapolace) - hledáme takovou funkci $\Phi_m(x)$, která má v zadaných bodech $x_0, \ldots, x_n$ stejné hodnoty (případně i derivace nejnižších řádů) jako funkce $f(x)$ (prochází body zadanými v tabulce)
  - Globální interpolace - v celém intervalu jsou koeficienty interpolační funkce stejné (např. Lagrangeův či Newtonův interpolační polynom, Hermiteova interpolace)
  - Lokální interpolace - celý interval rozdělen na podintervaly a v každém podintervalu má interpolační funkce jiné koeficienty (např. spline)
- Čebyševovy aproximace - hledáme $\Phi_m(x)$ tak, aby se na zadaném intervalu $\langle a, b\rangle$ minimalizoval maximální rozdíl mezi $f(x)$ a $\Phi_m(x)$, tj. minimalizujeme

  $$\max_{x\in\langle a,b\rangle} |f(x) - \Phi_m(x)|.$$

  Nazývá se též nejlepší stejnoměrná aproximace. Používá se často pro výpočet hodnot funkcí. K takové interpolaci lze dospět, pokud zvolíme optimální hodnoty $x_0,\ x_1, \ldots,\ x_n$, ve kterých funkci tabelujeme.
- Aproximace metodou nejmenších čtverců - minimalizujeme

  $$\int_a^b w(x)\ \ [f(x) - \Phi_m(x)]^2\, \mathrm{d}x \qquad \text{nebo} \qquad \sum_{i=0}^{n} w(x_i)\ \ [f(x_i) - \Phi_m(x_i)]^2 .$$

  Jde o spojitý nebo diskrétní případ aproximace metodou nejmenších čtverců. U diskrétního případu je vždy $n > m$ a tak funkce $\Phi_m(x)$ neprochází $\forall$ zadanými body. Často se volí $w(x) = 1$, pokud $w(x) \neq const$, pak mluvíme o vážené metodě nejmenších čtverců a funkci $w(x)$ nazýváme vahou.
  Pozn. Pokud cílem aproximace výsledků měření metodou nejmenších čtverců není jen nalézt přibližné funkční hodnoty, ale jde především o určení hodnot koeficientů koeficientů $c_0, \ldots, c_m$, stanovení přesnosti určení těchto koeficientů a stanovení, zda je možno dobře aproximovat naměřené hodnoty v dané třídě funkcí, úlohu nazýváme regrese (vyrovnávací počet, modelování dat) a úloha patří do statistického zpracování dat.

2 Interpolace a extrapolace

Interpolační metody tedy dělíme na

- globální - ve všech podintervalech interpolujeme stejnou funkcí
- lokální - v různých podintervalech používáme různé funkce, příkladem lokální interpolace je spline, která je na hranicích podintervalu spojitá včetně alespoň první derivace.

2.1 Lagrangeův interpolační polynom

Polynom L_n stupně nejvýše n takový, že $L_n(x_0) = y_0$, $L_n(x_1) = y_1$, ..., $L_n(x_n) = y_n$.

Uzly interpolace – body $x_0, \ldots, x_n$.

Pomocné funkce $F_i(x)$ $\forall i \in 0, \ldots, n$ takové, že

$$F_i(x_j) = \begin{cases} 1 & j = i \\ 0 & j \neq i \end{cases}$$

spočteme podle vztahu

$$F_i(x) = \frac{(x - x_0)(x - x_1)\ldots(x - x_{i-1})(x - x_{i+1})\ldots(x - x_n)}{(x_i - x_0)(x_i - x_1)\ldots(x_i - x_{i-1})(x_i - x_{i+1})\ldots(x_i - x_n)} \quad .$$

Lagrangeův interpolační polynom n-tého stupně má tvar

$$L_n(x) = \sum_{i=0}^{n} y_i\, F_i(x) = \sum_{i=0}^{n} y_i \, \frac{\omega_n(x)}{(x - x_i)\, \omega'_n(x_i)} \quad ,$$

kde $\omega_n(x) = (x - x_0)(x - x_1)\ldots(x - x_n)$.

Pro **ekvidistantní uzly** s krokem $h = x_{i+1} - x_i$ lze užít proměnnou $t = (x - x_0)/h$.
Lagrangeův polynom stupně n lze pak zapsat

$$L_n(x) = L_n(x_0 + th) = \sum_{i=0}^{n} y_i \, \frac{t(t-1)\ldots(t-n)}{(t-i)\; i!(n-i)!(-1)^{n-i}} \quad .$$

Pozn. Lagrangeův vzorec se nehodí pro numerický výpočet interpolace.

Pozn. Koeficienty interpolačního polynomu lze spočítat řešením systému lineárních rovnic. Matice tohoto systému (Van der Mondova) je však často špatně podmíněná (pro ekvidistantní uzly), výpočet koeficientů tedy není přesný.

2.2 Nevillův algoritmus

Hodnotu $L_n(x)$ vypočteme přesněji (byť pomaleji) pomocí **Nevillova algoritmu** (používá se i termín "iterovaná interpolace").
Principem je interpolace pomocí postupně se zvětšujícího počtu uzlů. Postupná přiblížení L_{ik}

$$L_{ik}(x) = L_i(x, x_i, x_{i-1}, \ldots, x_{i-k}, y_i, \ldots, y_{i-k})$$

jsou interpolační polynomy k-tého stupně. Tedy polynomy 0-tého stupně jsou $L_{i0}(x) = y_i$.

Pro polynomy platí rekurentní vztah

$$L_{ik}(x) = \frac{(x_i - x)L_{i-1,k-1} - (x_{i-k} - x)L_{i,k-1}}{x_i - x_{i-k}}.$$

Hodnoty jednotlivých polynomů lze zapsat do tabulky (**Nevillovo schéma**)

x	y	$k=0$	$k=1$	$\ldots$	$k=i-1$	$k=i$	$\ldots$	$k=n$
x_0	y_0	L_{00}						
x_1	y_1	L_{10}	L_{11}					
$\vdots$				$\ddots$				
x_{i-1}	y_{i-1}	$L_{i-1,0}$	$L_{i-1,1}$	$\ldots$	$L_{i-1,i-1}$			
x_i	y_i	$L_{i,0}$	$L_{i,1}$	$\ldots$	$L_{i,i-1}$	Lii		
$\vdots$							$\ddots$	
x_n	y_n	$L_{n,0}$	$L_{n,1}$	$\ldots$	$L_{n,i-1}$	L_{ni}	$\ldots$	L_{nn}

Lagrangeův interpolační polynom n-tého stupně v bodě x je tedy $L_n(x) = L_{nn}(x)$.

Odhad chyby aproximace interpolačním polynomem je dán rozdílem interpolace polynomem řádu n a nejlepší interpolace řádu $(n-1)$.

Praktická implementace Nevillova algoritmu, zahrnující odhad chyby využívá vztahů

$$\begin{aligned}
C_{m,i} &\equiv L_{i+m,m} - L_{i+m-1,m-1}, \\
D_{m,i} &\equiv L_{i+m,m} - L_{i+m,m-1}, \\
C_{m+1,i} &= \frac{(x_i - x)(C_{m,i+1} - D_{m,i})}{x_i - x_{i+m+1}}, \\
D_{m+1,i} &= \frac{(x_{i+m+1} - x)(C_{m,i+1} - D_{m,i})}{x_i - x_{i+m+1}},
\end{aligned}$$

V 0-tém kroku zvolíme za aproximaci $y_0(x) = L_{i,0} = y_i = C_{0,i} = D_{0,i}$ hodnotu v uzlu x_i nejbližším k x, v každém dalším kroku $y_{m+1}(x) = y_m(x) + \delta y_{m+1}$, kde $\delta y_{m+1} = C_{m+1,k} \ \vee \delta y_{m+1} = D_{m+1,k}$ tak, aby se v každém kroku pokud možno symetricky rozšířila oblast použitých uzlů, δy_{m+1} je odhad chyby interpolace v daném kroku. Nakonec vypočteme hodnotu interpolačního polynomu $y(x) = y_n(x)$ v bodě x a odhad chyby této aproximace δy_n.

Hodnota chyby polynomiální interpolace je dána vztahem

$$R(x) = f(x) - L_n(x) = \omega_n(x) \frac{f^{(n+1)}(\xi)}{(n+1)!} \quad ,$$

kde $\xi \in I \equiv \langle \min(x, x_0, \ldots, x_n), \max(x, x_0, \ldots, x_n) \rangle$. Tuto chybu lze odhadnout

$$|R(x)| \leq \frac{\max_{\xi \in I} \ |f^{(n+1)}(\xi)|}{(n+1)!} \ |\omega_n(x)| \ .$$

Extrapolace ($x \notin \langle x_0, x_n \rangle$) - chyba rychle roste se zvětšováním vzdálenosti od krajního uzlu.

Vlastnosti interpolace Pro ekvidistantní uzly x_i má při vyšších n stupních n interpolační polynom tendenci obsahovat velké oscilace mezi uzly (pokud sama funkce není polynomem). Polynomiální interpolace s ekvidistantními uzly vede obvykle k problematickým výsledkům pro $n \simeq 7$.

Konvergence Uvažuji konstantní interval $\langle a, b\rangle$. Nechť počet ekvidistantních uzlů $(n+1) \to \infty$ ($h = (b-a)/n \to 0$). Pak konvergence $L_n(x) \to f(x)$ je zaručena pro funkce analytické (mající komplexní derivaci) v celé komplexní rovině s výjimkou ∞. Jinak není zaručena ani bodová konvergence interpolačního polynomu k funkci.

Pozn. Pro neekvidistantní uzly může být situace lepší. Například pro $x_i = \frac{a+b}{2} + \frac{b-a}{2} \cos\left(\frac{2i+1}{2(n+1)}\pi\right)$, kde $i = 0, 1, \ldots, n$, konverguje pro $n \to \infty$ interpolační polynom k funkci pro $\forall$ funkce třídy C^1 na intervalu $\langle a, b\rangle$. $\Rightarrow$ aproximace Čebyševovými polynomy

2.3 Newtonův interpolační polynom

Newtonův interpolační polynom je jen jiný zápis Lagrangeova interpolačního polynomu

$$\begin{aligned} N_n(x) = a_0 &+ a_1(x-x_0) + a_2(x-x_0)(x-x_1) + \ldots + \\ &+ a_n(x-x_0)(x-x_1)\ldots(x-x_{n-1}) \quad . \end{aligned}$$

Pro vyjádření neznámých koeficientů a_i definujeme poměrné a obyčejné diference.

Poměrné diference

Poměrná diference prvního řádu je definována

$$f(x_i, x_{i+1}) = \frac{y_{i+1} - y_i}{x_{i+1} - x_i} \quad .$$

Poměrnou diferenci k-tého řádu definujeme rekurentním vztahem

$$f(x_i, x_{i+1}, \ldots, x_{i+k}) = \frac{f(x_{i+1}, \ldots, x_{i+k}) - f(x_i, \ldots, x_{i+k-1})}{x_{i+k} - x_i} \quad .$$

Pozn. Pro poměrnou diferenci druhého řádu platí

$$f(x_i, x_{i+1}, x_{i+2}) = \frac{\frac{y_{i+2}-y_{i+1}}{x_{i+2}-x_{i+1}} - \frac{y_{i+1}-y_i}{x_{i+1}-x_i}}{x_{i+2} - x_i}$$

Pozn. Vzorec pro výpočet k-té poměrné diference lze zapsat

$$f(x_i, x_{i+1}, \ldots, x_{i+k}) = \sum_{j=0}^{k} \frac{y_{i+j}}{\prod_{m=0,\ m\neq j}^{k}(x_{i+j} - x_{i+m})} \quad .$$

Obyčejné diference

Obyčejná diference 1. řádu je dána

$$\Delta^1 f_i = y_{i+1} - y_i$$

Obyčejná diference k-tého řádu je dána

$$\Delta^k f_i = \Delta^{k-1} f_{i+1} - \Delta^{k-1} f_i$$

Newtonův interpolační polynom zapíšeme pomocí poměrných diferencí ve tvaru

$$\begin{aligned} N_n(x) &= f(x_0) + (x - x_0) f(x_0, x_1) + \ldots + \\ &+ (x - x_0)(x - x_1) \ldots (x - x_{n-1}) f(x_0, x_1, \ldots, x_n) \end{aligned}$$

Ekvidistantní uzly – Newtonův interpolační polynom má tvar

$$\begin{aligned} N_n(x) = N_n(x_0 + th) &= f(x_0) + \frac{t}{1!}\Delta^1 f_0 + \frac{t(t-1)}{2!}\Delta^2 f_0 + \ldots + \\ &+ \frac{t(t-1)\ldots(t-n+1)}{n!}\Delta^n f_0 \end{aligned}$$

Pozn. Newtonovy interpolační polynomy pro uzly předcházející uzlu x_0 (Newtonův interpolační polynom vzad)

$$\begin{aligned} N_n(x) &= f(x_0) + (x - x_0) f(x_{-1}, x_0) + \ldots + \ldots + \\ &+ (x - x_0)(x - x_{-1}) \ldots (x - x_{-n+1}) f(x_{-n}, \ldots, x_0) \end{aligned}$$

2.4 Interpolace racionální lomenou funkcí

Některé funkce se špatně aproximují pomocí polynomů, ale lze je dobře aproximovat jejich podílem.

Nechť je zadáno $m+1$ bodů $(x_i, y_i),\ i = 0, \ldots, m$, pak lze zadanou funkci aproximovat racionální lomenou funkcí

$$R_{\mu,\nu}(x) = \frac{P_\mu(x)}{Q_\nu(x)} = \frac{p_0 + p_1 x + \ldots + p_\mu x^\mu}{q_0 + q_1 x + \ldots + q_\nu x^\nu} \quad ,$$

kde $\mu + \nu + 1 = m + 1$ a bez újmy na obecnosti lze položit $q_0 = 1$.

Interpolační racionální lomenou funkci počítáme rekurentním algoritmem, který je obdobou Nevillova algoritmu, navrženým Stoerem a Bulirschem.

2.5 Hermiteova interpolace

Interpolace, která kromě funkčních hodnot má zadané některé **derivace**.

Nechť jsou dány hodnoty funkce $f(x_i) = y_i$ v uzlech x_i pro $i = 0, 1, \ldots, m$ a navíc pro některé hodnoty i jsou dány hodnoty jejích derivací $f'(x_i) = y_i'$, $f''(x_i) = y_i''$, ..., $f^{(\alpha_i)}(x_i) = y_i^{(\alpha_i)}$.

Všechny podmínky splní polynom $H_n(x)$ stupně $n = m + \alpha_0 + \ldots + \alpha_m$. Lze ho vyjádřit

$$H_n(x) = L_m(x) + \omega_m(x) H_{n-m-1}(x)$$

pomocí Lagrangeova polynomu m-tého stupně, který prochází všemi uzly, a polynomu ω_m stupně $(m+1)$.
Hermiteův polynom $H_{n-m-1}(x)$ je dán tak, aby byly splněny podmínky pro derivace v uzlech, tedy $y_i' = L_m'(x_i) + \omega_m'(x_i) H_{n-m-1}$. Odtud vyplývá, že pro $i = 0, 1, 2, \ldots, p_1$ ($p_1 + 1$ je počet zadaných prvních derivací) platí

$$H_{n-m-1}(x_i) = \frac{y_i' - L_m'(x_i)}{\omega_m'(x_i)} = z_i \quad .$$

Provádíme tedy Hermiteovu interpolaci hodnot z_i s tím, že maximální stupeň zadané derivace již máme o 1 nižší. Postupným opakováním uvedeného algoritmu nalezneme požadovanou interpolaci.

Pozn. Někdy se pod Hermiteovým interpolačním polynomem rozumí Hermiteova interpolace v užším smyslu, kdy jsou ve $\forall$ uzlových bodech zadány hodnoty funkce a její 1. derivace.

2.6 Interpolační spline

Interpolační spline je lokální interpolace taková, že kromě průchodu všemi uzlovými body $f(x_i) = y_i$ má spojitou alespoň první derivaci ve $\forall\ x_i$, tedy

$$\lim_{x \to x_{i-}} f'(x) = \lim_{x \to x_{i+}} f'(x) \quad .$$

V každém subintervalu $\langle x_i, x_{i+1} \rangle$ na interpolační spline klademe čtyři podmínky (y_i, y_{i+1}, $y'_{i+} = y'_{i-}$, $y'_{i+1-} = y'_{i+1+}$), proto užívané funkce musí mít alespoň 4 volitelné parametry.

Kubický spline používá kubické polynomy pro konstrukci interpolačního splinu.

Okraje - zde nemá smysl mluvit o spojitosti derivací, proto je třeba 2 podmínky dodefinovat.
Možnosti jsou

1. Pokud známe hodnoty derivací na okraji, pak zadáme $y'(x_0) = y'_0$ a/nebo $y'(x_n) = y'_n$, tedy spline má zadanou hodnotu derivace na okraji.

2. Pokud derivaci neznáme, zadáme $y''_0 = 0$ a/nebo $y''_N = 0$. **Přirozený spline** – $y'' = 0$ na obou okrajích.

Pozn. Zadání obou dodatečných podmínek na jednom okraji by sice zjednodušilo výpočet aproximace, ale takový algoritmus je nepoužitelný vzhledem k numerické nestabilitě (oscilace interpolačního splinu exponenciálně rostoucí od daného okraje).

Konstrukce kubického splinu
Druhá derivace kubického polynomu je lineární funkcí a tedy v intervalu $\langle x_j, x_{j+1} \rangle$

$$
\begin{aligned}
y'' &= A\, y''_j + B\, y''_{j+1} \\
A &= \frac{x_{j+1} - x}{x_{j+1} - x_j} \qquad\qquad B = 1 - A = \frac{x - x_j}{x_{j+1} - x_j}
\end{aligned}
$$

Po dvojím zintegrování této rovnice při uvážení podmínek
$y(x_j) = y_j \;\wedge\; y(x_{j+1}) = y_{j+1}$ dostaneme

$$
\begin{aligned}
y &= A\, y_j + B y_{j+1} + C y''_j + D y''_{j+1} \\
C &= \frac{1}{6}(A^3 - A)(x_{j+1} - x_j)^2 \qquad\qquad D = \frac{1}{6}(B^3 - B)(x_{j+1} - x_j)^2
\end{aligned}
$$

Derivace $y'(x)$ má tvar

$$
y'(x) = \frac{y_{j+1} - y_j}{x_{j+1} - x_j} - \frac{3A^2 - 1}{6}(x_{j+1} - x_j)\, y''_j + \frac{3B^2 - 1}{6}(x_{j+1} - x_j)\, y''_{j+1}
$$

Hodnoty y_j'' určíme z podmínky spojitosti první derivace splinu $(y'(x_j))_- = (y'(x_j))_+$ ve tvaru

$$\frac{x_j - x_{j-1}}{6}\, y_{j-1}'' + \frac{x_{j+1} - x_{j-1}}{3}\, y_j'' + \frac{x_{j+1} - x_j}{6}\, y_{j+1}'' = \frac{y_{j+1} - y_j}{x_{j+1} - x_j} - \frac{y_j - y_{j-1}}{x_j - x_{j-1}}$$

Jde tedy o soustavu lineárních rovnic pro koeficienty y_j'', kde $j = 0, 1, \ldots, n$ s **tridiagonální**, diagonálně dominantní maticí. Za znalosti $\forall\ y_j''$ pak snadno určíme hodnotu splinu pro libovolné $x \in \langle x_0, x_n \rangle$.

Konvergence kubického splinu

Nechť $f(x)$ je řádu C^q na intervalu $\langle a, b\rangle$ (tedy spojité derivace až do q-té včetně), kde $q = 0, 1, 2, 3, 4$. Dále nechť interval $\langle a, b\rangle$ je rozdělen na podintervaly délky h_i, kde $i = 1, \ldots, n$, a $\max_i h_i = h$. Mějme dále konstantu K, pro kterou platí $K \geq \max_i h_i / \min_i h_i$. Je-li $S(x)$ kubický interpolační spline, pak pro $p = 0, 1, 2, 3$ platí

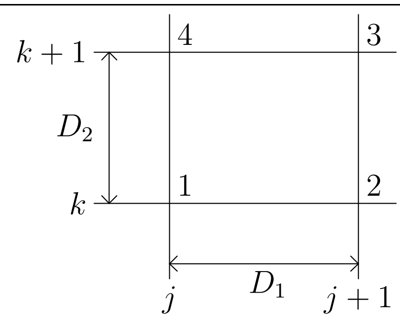
$$|f^{(p)}(x) - S^{(p)}(x)| \leq C\ K\ h^{q-p} \quad ,$$

kde konstanta C nezávisí na x ani na způsobu dělení intervalu $\langle a, b\rangle$.

3 Interpolace ve 2 (více) dimenzích

3.1 Spojitá lokální interpolace

Interval ve dvou proměnných (x_1, x_2) je znázorněn na obrázku



Obrázek 1: Interval ve 2 dimenzích

Hodnoty v jednotlivých bodech (viz obr. 1) jsou zadány takto $y_1 \equiv y[j,k]$, $y_2 \equiv y[j+1,k]$, $y_3 \equiv y[j+1,k+1]$ a $y_4 \equiv y[j,k+1]$.

Definujeme t a u vztahy

$$t = \frac{x_1 - x_1[j]}{D_1} \quad , \qquad u = \frac{x_2 - x_2[k]}{D_2} \quad .$$

Pak má lokální interpolace tvar

$$y(x_1, x_2) = (1-t)(1-u)y_1 + t(1-u)y_2 + tu\ y_3 + (1-t)u\ y_4 \quad .$$

Tato interpolace s bází $(1, x_1, x_2, x_1x_2)$ je lineární v každé z proměnných $x_1,\ x_2$. Spojitost na hranicích intervalu je zajištěna – na hranicích přímka, derivace ve směru kolmém k hranici ovšem nemusí existovat.

3.2 Globální interpolace

Má vyšší řád přesnosti. Lze ji sestrojit například následovně

1. Interpolujeme ve směru x_2 a pro $\forall j$ získáme $y(x_1[j], x_2)$.
2. Interpolujeme ve směru x_1 hodnoty $y(x_1[j], x_2)$ a získáme interpolační funkci $y(x_1, x_2)$.

3.3 Bikubická interpolace

Jde o lokální interpolace Hermiteova typu. V každém bodě $(x_1[j], x_2[k])$ jsou zadány hodnoty funkce a jejích parciálních derivací

$$y(x_1, x_2) \quad , \qquad \frac{\partial y}{\partial x_1} \quad , \qquad \frac{\partial y}{\partial x_2} \quad , \qquad \frac{\partial^2 y}{\partial x_1 \partial x_2}$$

Jde tedy o 4 podmínky v každém ze 4 uzlů 2-rozměrného intervalu. a tak lze určit 16 neznámých koeficientů c_{ij} bikubické interpolace

$$y(x_1, x_2) = \sum_{i=1}^{4} \sum_{j=1}^{4} c_{ij}\ t^{i-1}\ u^{j-1}.$$

3.4 Bikubický spline

Lokální interpolace se spojitými parciálními derivacemi v obou směrech na hranicích intervalů. Vypočteme spline v jednom směru (např. x_1) a uschováme hodnoty $y(x_1, x_2[k])$. Tyto hodnoty interpolujeme splinem v druhém směru x_2.

4 Čebyševovy aproximace

4.1 Čebyševova úloha

Hledá se **nejlepší stejnoměrná aproximace** funkce v daném intervalu. Jedná se o funkci $h(x)$, která v daném intervalu $\langle a, b\rangle$ minimalizuje maximální absolutní hodnotu chyby $\max\limits_{x\in\langle a,b\rangle} |f(x) - h(x)|$ v určité třídě funkcí.

Polynom, který je nejlepší stejnoměrnou aproximací funkce na daném intervalu $\langle a, b\rangle$ mezi polynomy daného stupně, se někdy nazývá polynom "minimax".

Polynom "minimax" existuje za velmi obecných podmínek, ale špatně se konstruuje. Používá se **Remezův algoritmus**, který postupně iteruje polohu bodů s extrémy $f(x) - P(x)$ (současně tedy mění polohu uzlů, kde se $y(x) = f(x)$) tak, že interpolační polynom konverguje k polynomu "minimax". Pro výpočet funkčních hodnot je však Remezův algoritmus příliš zdlouhavý.

Obdobně je zaručena existence nejlepší stejnoměrné aproximace v intervalu $\langle a, b\rangle$ mezi racionálními lomenými funkcemi typu $R_{\mu,\nu}$. Pro výpočet se opět používá iterační algoritmus výběru bodů s extrémy – **Remezův algoritmus**. (Podrobněji viz. Vospěl nebo Vitásek).

4.2 Čebyševovy polynomy

Aproximace Čebyševovými polynomy se lehce konstruuje a je téměř tak přesná jako nejlepší stejnoměrná aproximace. Často se používá pro výpočet funkcí.

Pro interpolaci Čebyševovými polynomy se libovolný interval lineárně transformuje na interval $\langle -1, 1\rangle$ Každému $t \in \langle a, b\rangle$ přiřadíme hodnotu $x \in \langle -1, 1\rangle$ funkčním předpisem

$$x = \frac{2\,t - (a + b)}{b - a} \quad .$$

Čebyševův polynom lze psát ve tvaru

$$T_n(x) = \cos(n \arccos x)$$

Polynomy 0-tého a 1-ního stupně jsou tedy $T_0(x) = 1$ a $T_1(x) = x$. Čebyševův polynom $n + 1$-tého stupně lze též vyjádřit pomocí rekurentního vztahu

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$$

Pomocí rekurentního vztahu lze odvodit tvar dalších Čebyševových polynomů

$$\begin{aligned} T_2(x) &= 2x^2 - 1, \\ T_3(x) &= 4x^3 - 3x, \\ T_4(x) &= 8x^4 - 8x^2 + 1 \quad . \end{aligned}$$

Kořeny a extrémy Čebyševův polynom $T_n(x)$ má n kořenů v intervalu $\langle -1, 1\rangle$ v bodech

$$x = \cos\left(\frac{\pi\ (k\ - \frac{1}{2})}{n}\right) \qquad k = 1, 2, \ldots, n$$

V intervalu $\langle -1, 1\rangle$ má $n+1$ extrémů $|T_n(x)| = 1$ v bodech $x = \cos(k\,\pi/n)$, kde $k = 0, 1, \ldots, n$.

Ortogonalita

Čebyševovy polynomy jsou ortogonální polynomy s vahou $\frac{1}{\sqrt{1-x^2}}$ na intervalu $\langle -1, 1\rangle$

$$\int_{-1}^{1} \frac{T_i(x)T_j(x)}{\sqrt{1-x^2}} dx = \begin{cases} 0 & i \neq j \\ \frac{\pi}{2} & i = j \neq 0 \\ \pi & i = j = 0 \end{cases}$$

Diskrétní ortogonalita

Nechť x_k, $k = 1, \ldots, m$ jsou kořeny Čebyševova polynomu $T_m(x)$. Pak pro $\forall i, j < m$ platí

$$\sum_{k=1}^{m} T_i(x_k)T_j(x_k) = \begin{cases} 0 & i \neq j \\ \frac{m}{2} & i = j \neq 0 \\ m & i = j = 0 \end{cases} .$$

Aproximace Čebyševovými polynomy

Funkci $f(x)$ aproximujeme

$$\begin{aligned} f(x) \approx T(x) &= \frac{1}{2}c_0 + \sum_{j=1}^{N-1} c_j T_j(x) \quad , \\ \text{kde} \qquad c_j &= \frac{2}{N}\sum_{k=1}^{N} f\left[\cos\left(\frac{\pi(k\ - \frac{1}{2})}{N}\right)\right]\cos\left(\frac{\pi j(k\ - \frac{1}{2})}{N}\right) . \end{aligned}$$

Hodnoty funkce $f(x)$ jsou rovny hodnotám funkce $T(x)$ ve všech N nulových bodech polynomu $T_N(x)$.

Výpočet funkce pomocí Čebyševových polynomů

Často volíme pro výpočet koeficientů relativně vysoký řád aproximace $N \simeq 30 - 50$ (procedura `CHEBFT` v knihovně *Numerical Recipies*). Koeficienty Čebyševova rozvoje jdou obvykle $\rightarrow 0$ relativně rychle. Vysoké N dovoluje přesně stanovit počet členů potřebných pro výpočet funkce se zadanou přesností ε. Pokud je $\sum_{k=m}^{N} |c_k| < \varepsilon$, potom pro výpočet funkce stačí použít prvních m členů rozvoje (procedura `CHEBEV` v knihovně *Numerical Recipies*).

Pozn. Počítání Lagrangeova polynomu s body danými nulovými body Čebyševova polynomu je možné, ale pro $N > 8$ je méně přesné a obtížnější.

Pozn. Procedura `CHEBEV` používá pro sumu funkcí zadaných rekurentně používá **Clenshawovy formule**.
Pokud je řada funkcí dána rekurentně

$$F_{n+1}(x) = \alpha(n,x)\ F_n(x) + \beta(n,x)\ F_{n-1}(x)$$

pak lze sumu

$$f(x) = \sum_{k=0}^{N} c_k F_k(x)$$

počítat tak, že postupně provádíme

$$\begin{aligned} y_{N+2} &= 0 \ , \qquad y_{N+1} = 0 \\ y_k &= \alpha(k,x) y_{k+1} + \beta(k+1,x) y_{k+2} + c_k \ , \qquad k = N, N-1, \ldots, 1 \\ f(x) &= \beta(1,x) F_0(x) y_2 + F_1(x) y_1 + F_0(x) c_0 \ . \end{aligned}$$

Clenshawova formule nemusí být vhodná pro $\forall$ řady (může vést i ke katastrofální ztrátě přesnosti).

Integrál a derivace pomocí Čebyševových polynomů

Koeficienty C_i Čebyševova rozvoje integrálu funkce $f(x)$ jsou dány

$$C_i = \frac{c_{i-1} - c_{i+1}}{2i} \quad , \qquad \text{kde } i > 0 \quad .$$

kde c_i jsou koeficienty Čebyševova rozvoje funkce $f(x)$.

Pro derivaci funkce $f(x)$ platí vzorec

$$c'_{i-1} = c'_{i+1} + 2ic_i \quad , \qquad \text{kde} \quad i = N-1, N-2, \ldots, 1 \quad .$$

5 Aproximace metodou nejmenších čtverců

Aproximační funkce neprochází zadanými body. Využívá se na příklad při aproximaci výsledků měření s nezanedbatelnými chybami. Rozlišujeme

- **diskrétní aproximaci** - funkce je zadána v diskrétních bodech x_i – hledáme diskrétní funkci $\phi_M(x_i)$, která v určité třídě funkcí minimalizuje funkcionál

$$\varrho_N = \sqrt{\sum_{i=1}^{N} w_i[f(x_i) - \phi_M(x_i)]^2}$$

- **spojitou aproximaci** – funkce je zadána v celém intervalu $\langle a, b \rangle$ – hledáme funkci $\phi_M(x)$, která v určité třídě funkcí minimalizuje funkcionál

$$\varrho = \sqrt{\int_a^b [f(x) - \phi_M(x)]^2 w(x) dx}$$

kde w je váhová funkce (často $w(x) = 1$).

Tvar funkce Φ_M je obvykle zadán až na M neznámých parametrů c_j. Aproximace metodou nejmenších čtverců může být

- **lineární** - funkce je lineární vzhledem k neznámým parametrům c_j, jde tedy o zobecněný polynom

$$\phi_M(x) = \sum_{j=1}^{M} c_j \, g_j(x) \quad ,$$

  bázové funkce g_j jsou zadané.

- **nelineární** - funkce není lineární vzhledem ke koeficientům c_j

$$\phi_M = \phi_M(x, c_1, \ldots, c_M)$$

5.1 Diskrétní aproximace

Ve statistice je často třeba aproximovat naměřená data, zatížená náhodnými chybami měření nebo náhodným šumem funkční závislostí. Uvedená závislost je nazývána modelem, který je znám až na M neznámých parametrů. V matematické statistice je uvedená úloha nazývána regresí, a k jejímu řešení se často používá metoda nejmenších čtverců. Hlavním cílem statistické analýzy je

1. vypočítat koeficienty c_j, kde $j = 1, \ldots, M$
2. určit směrodatné odchylky koeficientů δc_j, kde $j = 1, \ldots, M$
3. zjistit kvalitu modelu (zda je použitý model statisticky přijatelný)

Zde chceme nalézt vhodnou aproximaci. Úlohou může být nalezení hladké aproximace bez konstrukce modelu. Jednou z možností je použít **zvonové spliny** jako bázové funkce.
Nechť je dána síť bodů s krokem h. Konstruujeme zvonový spline (bell spline, B-spline) se středem v μ. Označme

$$p = h \; - |x - \mu| \;\; , \qquad q = 2h - |x - \mu| \;\; ,$$

potom je zvonový spline $B_\mu(x)$ dán vztahy

$$B_\mu(x) = \begin{cases} 0 & q \le 0 \\ q^3 & p \le 0 \\ q^3 - 4p^3 & p \ge 0 \end{cases}$$

Zvonový spline je funkce třídy $C^{(2)}$ nenulová pouze v intervalu $\langle \mu - 2h, \mu + 2h \rangle$. Hledání koeficientů u zvonových splinů vede na úlohu řešení systému lineárních rovnic s pásovou maticí.

Výpočet parametrů lineární aproximace metodou nejmenších čtverců

1. Položíme $f(x_i) = \sum_{j=1}^{M} c_j g_j(x_i)$, kde $M < N$, a získáme tak N rovnic pro M neznámých koeficientů c_j. Ve smyslu nejmenších čtverců úlohu řeší **SVD metoda**.
2. Funkce ϱ_N^2 má minimum, pokud pro $\forall l = 1, \ldots, M$ platí

$$\frac{\partial}{\partial c_l} \left[\sum_{i=1}^{N} w_i \left(f(x_i) - \sum_{j=1}^{M} c_j g_j(x_i) \right)^2 \right] = 0.$$

Řešíme pak soustavu M lineárních rovnic o M neznámých c_j nazývanou **normální rovnice**.

Pozn. Jestliže definujeme skalární součin $(f, g) \equiv \sum_{i=1}^{N} w_i f(x_i) g(x_i)$, pak lze soustavu normálních rovnic přepsat do tvaru

$$\sum_{j=1}^{M} c_j (g_j, g_l) = (f, g_l).$$

Výběr bázových (základních) funkcí

1. **Polynomy** - jako bázové funkce jsou často voleny $1, x, x^2, \ldots, x^{M-1}$. Tyto základní funkce jsou však vhodné jen pro malá M, protože při větších M jsou tyto polynomy přibližně rovnoběžné, soustava normálních rovnic je špatně podmíněná a chyba koeficientů c_j je značná.
2. **Ortogonalizované polynomy** - problémy, které vznikají v předchozím případě, zde odpadají. Tyto polynomy konstruujeme Gramm–Schmidtovým ortogonalizačním procesem.
3. **Trigonometrické polynomy** - mají základní funkce 1, $\cos x$, $\sin x$, $\cos 2x$, $\sin 2x$, $\ldots$. Jsou ortogonální pro všechny body $x_i = \frac{2\pi i}{2N+1}$, kde $i = 0, 1, \ldots, 2N$.

5.2 Spojitý případ

Skalární součin je pak definován vztahy

$$(f, g) \equiv \int_a^b f(x)\, g(x) dx \quad , \quad \text{příp.} \quad (f, g) \equiv \int_a^b w(x)\, f(x)\, g(x) dx$$

Normální rovnice mají tvar

$$\sum_{j=1}^{M} c_j (g_j, g_l) = (f, g_l).$$

Výběr bázových funkcí

1. **Ortogonální polynomy**

   (a) V případě váhy $w = 1$ v intervalu $\langle -1, 1\rangle$ užijeme **Legendreovy polynomy**,

   $$P_l(x) = \frac{1}{2^l\, l!}\,\frac{d^l}{dx^l}\left[x^2 - 1\right]^l$$

   Nejnižší Legendreovy polynomy jsou

   $$P_0 = 1\ ,\quad P_1 = x\ ,\quad P_2 = \frac{1}{2}(3x^2 - 1)\ ,\quad P_3 = \frac{1}{2}(5x^3 - 3x)\ .$$

   (b) Je-li interval $x \in \langle -1, 1\rangle$ a váhová funkce $w = \frac{1}{\sqrt{1-x^2}}$, užijeme **Čebyševovy polynomy** $T_n(x)$.

   (c) Při váhové funkci $w = e^{-x}$ pro všechna $x \in \langle 0, \infty)$ jsou ortogonální **Laguerrovy polynomy**.

   (d) Při váze $w = e^{-x^2}$ pro všechna $x \in (-\infty, \infty)$ jsou ortogonální **Hermiteovy polynomy**.

2. **Trigonometrické funkce** - volba trigonometrických funkcí 1, $\sin x$, $\cos x$, $\sin 2x$, $\cos 2x$, $\ldots$ jako základních funkcí vede k aproximaci **Fourierovu řadou**. Tyto funkce jsou ortogonální na $\langle a, a + 2\pi\rangle$, kde a je libovolné. Váhovou funkci pokládáme $w \equiv 1$.

6 Výpočet funkcí

Tato sekce obsahuje několik poznámek o metodách používaných při výpočtu funkcí.

6.1 Výpočet hodnoty polynomu

Počet operací potřebných pro výpočet hodnoty polynomu $P_n(x) = a_0 + a_1\, x + a_2\, x^2 + \ldots + a_n\, x^n$ lze zmenšit převedením do tvaru

$$P_n(x) = \{\ldots[(a_n\, x + a_{n-1})\, x + a_{n-2}]\, x + \ldots + a_1\}\, x + a_0$$

Tento postup je nazýván Hornerovo schéma. V jednom cyklu lze počítat kromě hodnoty polynomu i hodnoty jeho derivací.

6.2 Kořeny kvadratické rovnice

Klasický vzorec

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

vede při $|4ac| \ll b^2$ ke ztrátě přesnosti u jednoho z kořenů. Při $b < 0$ je to kořen x_2 a při $b > 0$ je to kořen x_1.

Pro $b < 0$ je pak přesnější postup

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}\,, \qquad x_2 = \frac{c}{a\, x_1}$$

6.3 Výpočet funkcí pomocí mocninných řad

Mnoho funkcí lze vyjádřit ve tvaru mocninné řady

$$f(x) = \sum_{k=0}^{\infty} a_k (x - x_0)^k \quad .$$

Často se však tato řada nehodí k výpočtu hodnot funkce nebo se hodí jen pro x blízká k x_0. Důvodem může být buď ztráta přesnosti a/nebo pomalá konvergence některých mocninných řad. Popis metod urychlujících konvergenci mocninných řad je mimo rozsah naší přednášky.

6.4 Nekonečné zlomky

Nekonečné zlomky jsou funkce ve tvaru

$$f(x) = b_0 + \cfrac{a_1}{b_1 + \cfrac{a_2}{b_2 + \cfrac{a_3}{b_3 + \dots}}} \quad .$$

Často se používá jejich zápis ve tvaru

$$f(x) = b_0 + \frac{a_1}{b_1+}\ \frac{a_2}{b_2+}\ \frac{a_3}{b_3+}\ \dots \quad .$$

Například pro funkci $\mathrm{tg}(x)$ existuje vyjádření pomocí nekonečného zlomku

$$\mathrm{tg}(x) = \frac{x}{1-}\ \frac{x^2}{3-}\ \frac{x^2}{5-}\ \frac{x^2}{7-}\ \dots \quad .$$

Přímý výpočet složeného zlomku je nepraktický, navíc zvýšení řádu aproximace vyžaduje nový výpočet od počátku. Proto je výhodný rekurentní postup

$$f_n = \frac{A_n}{B_n}\ , \quad \text{kde} \quad A_j = b_j A_{j-1} + a_j A_{j-2} \quad \text{a} \quad B_j = b_j B_{j-1} + a_j B_{j-2}.$$

Počáteční hodnoty klademe $A_{-1} \equiv 1$, $B_{-1} \equiv 0$, $A_0 \equiv b_0$ a $B_0 \equiv 1$.

6.5 Rekurentní vztahy pro výpočet funkcí

Mnoho funkcí je dáno rekurentními vztahy

$$f_{n+1}(x) = b_n(x)\ f_n(x) + c_n(x)\ f_{n-1}(x)$$

Vlastnosti takové rekurence jsou dány vlastnostmi kvadratické rovnice

$$y^2 - b_n y - c_n = 0 \quad \text{kořeny} \quad y_1,\ y_2$$

Označme y_1 ten kořen, který odpovídá výpočtu funkce. Pokud je $|y_1| > |y_2|$, rekurzi začínající od nejnižších n lze použít pro výpočet f_n s vysokým n. Pokud je ale $|y_1| < |y_2|$, pak je takový algoritmus numericky nestabilní.

Pozn. Pokud je rekurentní vztah nestabilní při růstu n, pak je stabilní při zmenšování n.

7 Numerické derivování

Pro výpočet numerické derivace je možno použít následujících aproximací funkcí

1. Interpolačního spline
2. Aproximace pomocí Čebyševových polynomů
3. Aproximace metodou nejmenších čtverců (u derivací funkce dané naměřenými hodnotami)
4. Interpolační polynom

   Pro stupeň polynomu $n = 1$ má derivace tvar

   $$f'(x) = \frac{1}{h}[-y_0 + y_1] + \frac{1}{2}\ h\ f''(\xi) \quad ,$$

   kde $\xi \in \langle \min(x, x_0, x_1), \max(x, x_0, x_1) \rangle$.

   Pro polynom stupně $n = 2$ s ekvidistantními uzly označme $t = (x - x_0)/h$. Pak má vzorec pro první derivaci tvar

   $$\begin{aligned} f'(x) &= \frac{1}{2h}[y_0(2t-3) - 2y_1(2t-2) + y_2(2t-1)] + R_2(x), \\ f'(x_0) &= \frac{1}{2h}[-3y_0 + 4y_1 - y_2] + \frac{1}{3}h^2 f'''(\xi), \\ f'(x_1) &= \frac{1}{2h}[-y_0 + y_2] - \frac{1}{6}h^2 f'''(\xi), \\ f'(x_2) &= \frac{1}{2h}[y_0 - 4y_1 + 3y_2] + \frac{1}{3}h^2 f'''(\xi). \end{aligned}$$

   Pro stupeň polynomu $n = 3$ je například

   $$f'(x_0) = \frac{1}{6h}[-11y_0 + 18y_1 - 9y_2 + 2y_3] - \frac{1}{4}h^3 f^{(4)}(\xi).$$

   Pro stupeň polynomu $n = 4$ získáme například symetrickou derivaci z 5 bodů

   $$f'(x_2) = \frac{1}{12h}[y_0 - 8y_1 + 8y_3 - y_4] + O(h^4).$$

   Pro $n = 6$ je například v bodě x_3

   $$f'(x_3) = \frac{1}{60h}[-y_0 + 9y_1 - 45y_2 + 45y_4 - 9y_5 + y_6] + O(h^6).$$