Predavanja 8

Osmi sklop izročkov

Fakulteta za računalništvo in informatiko
Univerza v Ljubljani

30. november 2021

Višanje stopnje aproksimacije

Višanje stopnje interpolacijskega ne izboljša vedno aproksimacije funkcije s polinomom. Znan je Rungejev primer, ko funkcijo $f(x) = \frac{1}{1+x^2}$ interpoliramo na intervalu $[-5, 5]$ z ekvidistantnimi točkami, tj.

$$x_0 = -5, \quad x_1 = -5+10\cdot\frac{1}{n}, \quad x_2 = -5+10\cdot\frac{2}{n}, \quad \ldots, \quad x_{n-1} = -5+10\cdot\frac{n-1}{n}, \quad x_n = 5.$$

Pričakujemo, da se bo interpolacijski polinom vse bolj prilegam naši funkciji. Izkaže pa se, da temu ni tako. Če interpoliramo v točkah Čebiševa

$$x_0 = 5\cos\left(\frac{\pi}{2(n+1)}\right), \quad x_1 = 5\cos\left(\frac{3\pi}{2(n+1)}\right), \quad \ldots, \quad x_n = 5\cos\left(\frac{(2n+1)\pi}{2(n+1)}\right),$$

pa z višanjem stopnje res dobimo boljše prileganje.

Toda: Za kateri koli nabor točk obstaja funkcija, za katero z višnjam stopnje interpolacijskega polinoma ne dobimo boljšega prileganja.

Rešitev problema višanja stopnje polinoma je uporaba zlepkov polinomov nizkih stopenj. Najboljši so kubični zlepki.

Primer:

https://zalara.github.io/Runge.m

Za delovanje zgornje skripte potrebujete še skripti:

https://zalara.github.io/vrednostIP.m

https://zalara.github.io/deljeneDif.m

Ti dve skripti namreč interpolacijski polinom in njegovo vrednost računata v stabilnejši Newtonovi bazi. Za radovedne je računanje Newtonovega interpolacijskega polinoma razloženo tu:

https://zalara.github.io/Newtonov-interpolacijski-polinom.pdf

Aproksimacija po metodi najmanjših kvadratov

Za funkcijo, podano v n točkah $(x_0, y_0), \ldots, (x_n, y_n)$, iščemo polinom p_k stopnje $k \leqslant n$, za katerega ima izraz

$$E_{LSQ} = \sqrt{\sum_{i=0}^{n} (p_k(x_i) - y_i)^2}$$

najmanjšo vrednost. Če zapišemo na dolgo:

$$E_{LSQ} = \sqrt{\sum_{i=0}^{n} (a_0 + a_1 x_i + \ldots + a_k x_i^k - y_i)^2}.$$

Torej iščemo ekstrem funkcije več spremenljivk. Iz analize vemo, da je potreben pogoj za ekstrem

$$\frac{\partial E_{LSQ}}{\partial a_0} = \frac{\partial E_{LSQ}}{\partial a_1} = \ldots = \frac{\partial E_{LSQ}}{\partial a_k} = 0.$$

Naj bo

$$s_1 = x_0 + \ldots + x_n, \quad s_2 = x_0^2 + \ldots + x_n^2, \quad \ldots, \quad s_{2k} = x_0^{2k} + \ldots + x_n^{2k}.$$

Dobimo normalni sistem:

$$\begin{bmatrix} n & s_1 & s_2 & \ldots & s_k \\ s_1 & s_2 & s_3 & \ldots & s_{k+1} \\ s_2 & s_3 & s_4 & \ldots & s_{k+2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ s_k & s_{k+1} & s_{k+2} & \ldots & s_{2k} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_k \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^n y_i \\ \sum_{i=0}^n y_i x_i \\ \sum_{i=0}^n y_i x_i^2 \\ \vdots \\ \sum_{i=0}^n y_i x_i^k \end{bmatrix}, \qquad (1)$$

ki pa je pri velikem številu točk lahko slabo pogojen.

Predoločeni sistemi

Problem iz prejšnje strani je poseben primer t.i. predoločenega sistema:

Za matriko $A \in \mathbb{R}^{n \times m}$, $n \geqslant m$, in vektor $b \in \mathbb{R}^n$, iščemo vektor $x \in \mathbb{R}^m$, ki zadošča:

$$Ax = b. \tag{2}$$

Kadar je $n > m$, točna rešitev sistema (2) najverjetneje ne obstaja. Zato nas navadno zanima rešitev po metodi najmanjših kvadratov:

$$\text{Poišči } x \in \mathbb{R}^m, \text{ ki minimizira } \|Ax - b\|_2. \tag{3}$$

Trditev

Naj bo $\operatorname{rank}(A) = m$. *Rešitev sistema* (2) *po metodi najmanjših kvadratov je* $x \in \mathbb{R}^m$, *ki reši t.i. normalni sistem:*

$$\underbrace{A^T A}_{m \times m} x = \underbrace{A^T b}_{m \times 1}. \tag{4}$$

Dokaz.

Vektor x, ki minimizira $\|b - Ax\|_2$, je tisti, za katerega je $b - Ax$ pravokoten na vsak stolpični prostor oz. sliko matrike A. To pa pomeni, da mora biti skalarni produkt vsakega stolpca A z vektorjem $b - Ax$ enak 0. To lahko v kompaktni obliki zapišemo kar kot $A^T(b - Ax) = 0$. S preoblikovanjem tega pogoja dobimo (4). Iz predpostavke $\operatorname{rank}(A) = m$ dobimo $\operatorname{rank}(A)^T A = m$ in zato je sistem (4) rešljiv. □

QR razcep

Težava, ki se pojavi pri reševanju normalnega sistema (4), je numerična stabilnost. V prvem poglavju smo spoznali, da računanje skalarnega produkta v splošnem ni stabilna operacija. Do težav pride v primeru skoraj pravokotnih vektorjev u, v, ko je relativna napaka $\frac{\mathrm{fl}(\mathrm{fl}(u)\cdot\mathrm{fl}(v))}{u\cdot v}$ zaradi deljenja z majhnim številom lahko zelo velika.

Rešitev tega problema je uporaba t.i. QR razcepa matrike A.

Definicija

QR razcep matrike $A \in \mathbb{R}^{n\times m}$ sta ortogonalna matrika $Q \in \mathbb{R}^{n\times m}$ ($Q^T Q = I_m$) in zgornjetrikotna matrika $R \in \mathbb{R}^{m\times m}$, ki zadoščata

$$A = QR. \tag{5}$$

Iz (5) sledi, da se sliki matrike A in Q ujemata. Pogoj ortogonalnosti matrike Q pa pomeni, da so njeni stolpci normirani (tj. dolžine 1) in paroma pravokotni. Če označimo z $a_1, \ldots, a_m$ in $q_1, \ldots, q_m$ stolpce matrik A in Q ter $R = [r_{i,j}]_{i,j}$, potem veljajo zveze:

$$\begin{aligned}
q_1 &= \frac{1}{r_{11}} a_1, \\
q_2 &= \frac{1}{r_{22}} (a_2 - r_{12} q_1), \\
&\vdots, \\
q_m &= \frac{1}{r_{mm}} (a_m - r_{1m} q_1 - \ldots - r_{m-1,m} q_{m-1}).
\end{aligned} \tag{6}$$

Iz (6) lahko izpeljemo enega od načinov za izračun *QR* razcepa, tj. z uporabo Gram-Schmidtove ortogonalizacije (GSO):

```
A = [a_1, ..., a_m] je n × m matrika s stolpci a_1, ..., a_m

r_{1,1} = ||a_1||_2
q_1 = (1/r_{1,1}) a_1
for i = 2, ..., m
    q_i = a_i
    for j = 1, ..., i
        r_{j,i} = q_j^T a_i
        q_i = q_i - r_{j,i} q_j
        r_{i,i} = ||q_i||_2
        q_i = (1/r_{i,i}) q_i
```

- Z natančnim štetjem števila računskih operacij se izkaže, da je računska zahtevnost GSO $\frac{4}{3}n^3 + \mathcal{O}(n^2)$, kar je dvakrat dražje od računanja LU razcepa z delnim pivotiranjem.

- Z natančno analizo numerične stabilnosti se izkaže, da je računanje *QR* razcepa stabilna operacija.

Uporaba QR razcepa za reševanje sistema (2)

Trditev

Naj bo $A \in \mathbb{R}^{n\times m}$ in $\operatorname{rank}(A) = m$. Rešitev sistema (2) po metodi najmanjših kvadratov je enaka rešitvi zgornjetrikotnega sistema

$$Rx = Q^T b, \tag{7}$$

kjer je $A = QR$ za zgornjetrikotno matriko R in ortogonalno matriko Q.

Dokaz.

Vemo, da je rešitev sistema (2) po metodi najmanjših kvadratov kar rešitev normalnega sistema (4). Velja:

$$\begin{aligned} A^T Ax &= (QR)^T(QR)x = (R^TQ^T)(QR)x = R^T(Q^TQ)Rx = R^TRx, \\ A^Tb &= (QR)^Tb = R^TQ^Tb. \end{aligned}$$

Sistem (4) je tako ekvivalenten sistemu $R^TRx = R^TQ^Tb$. Ker je po predpostavki $\operatorname{rank}(A) = m$, je tudi $\operatorname{rank}(R) = \operatorname{rank}(R^T) = m$. Zato je $R^T \in \mathbb{R}^{m\times m}$ obrnljiva in z množenjem zadnje enačbe z leve z $(R^T)^{-1}$, dobimo (7). □

Opomba

- *V Matlabu QR razcep matrike izračunamo z ukazom $[Q, R] = qr(A)$.*
- *V Matlabu sistem $Ax = b$ rešimo po metodi najmanjših kvadratov z ukazom $A\backslash b$. V ozadju operatorja $\backslash$ je uporaba QR razcepa.*

Linearna regresija

Primer

Iščemo premico, ki se najbolje prilega podatkom $(x_1, y_1), \ldots, (x_n, y_m)$ *po metodi najmanjših kvadratov. Premica je oblike* $y = a + bx$. *Torej sta spremenljivki* a *in* b. *Sistem lahko zapišemo v obliki*

$$\underbrace{\begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \\ 1 & x_n \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} a \\ b \end{bmatrix}}_{\vec{x}} = \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}}_{\vec{b}}. \tag{8}$$

Po zgoraj napisanem je rešitev (8) *enaka*

$$\vec{x} = (A^T A)^{-1} A^T \vec{b} = \begin{bmatrix} n & \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i & \sum_{i=1}^n x_i^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i y_i \end{bmatrix}.$$

Na isti način pridemo tudi do sistema (1), *ki smo ga navedli kot uvod v predoločene sisteme.*

Poglavje 4

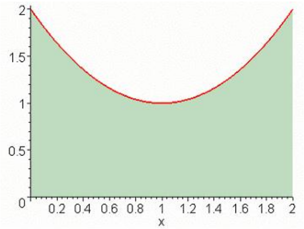
Numerična integracija

Numerična integracija

Naš cilj je izračunati določen integral

$$\int_a^b f(x)\, dx = F(b) - F(a)$$

funkcije $f(x)$. Tu je F nedoločen integral funkcije f.



Če ne znamo izračunati nedoločenega integrala F, smo v težavah. Npr. za $f(x) = e^{-x^2}$, $g(x) = \frac{\sin x}{x}$, $h(x) = x \tan x$.

Prav tako ne moremo točno izračunati vrednosti integrala, če imamo funkcijo podano samo na neki množici točk.