Poročilo projekta - Skoraj konstantne vsote trikotnih funkcij na zaprtem intervalu

Oskar Vavtar in Neža Kržan

6. 12. 2021

1 Uvod

1.1 Navodilo

Imamo nabor n trikotnih funkcij. Izmed njih želimo izbrati podmnožico največ sedmih, da bo njihova vsota čim bolj konstantna na intervalu $[0, 1]$ - s tem mislimo, da je razlika med maksimalno in minimalno vrednostjo vsote najmanjša možna. Ali je metoda celoštevilskega linearnega programiranja primerna za ta problem? Lahko uporabimo kakšne druge metode? Kako bi sami generirali smiselne vhodne podatke za ta problem? Bi lahko na enak način obravnavali tudi druge funkcije, npr. preproste stopničaste funkcije?

1.2 Razmislek in potek dela

Očitno je, da je več aspektov tega problema mogoče posplošiti. Trikotne funkcije lahko nadomestimo s katerimi drugimi preprostimi funkcijami (npr. stopničastimi, pravokotnimi). Namesto 7 funkcij, jih lahko izberemo r, $r < n$. Namesto intervala $[0, 1]$ pa lahko izberemo poljuben zaprt interval $[a, b]$, $-\infty < a \leq b < \infty$.

Problem sva se odločila rešiti za dva tipa trikotnih funkcij:

- simetrične,
- splošne.

Problem sva nameravala reševati z metodo celoštevilskega linearnega programiranja, ki se je kasneje izkazala za neustrezno. Namesto tega sva uporabila metodo mešano-celoštevilskega programiranja, ki v dovoli tudi realnoštevilske omejitve.

Kot opisano v navodilu je "čim bolj konstantno" mišljeno kot čim manjša razlika med maksimalno in minimalno vrednostjo vsote funkcij na intervalu. Iz nabora funkcij $\{f_1, \dots, f_n\}$ želimo torej izbrat tako podmnožico $\{g_1, \dots, g_r\}$, $r \leq n$, da bo dosežen

$$\min_{\{g_1,\dots,g_r\}\subseteq\{f_1,\dots,f_n\}} \left(\max_{x\in[a,b]} \sum_{i=1}^{r} g_i - \min_{x\in[a,b]} \sum_{i=1}^{r} g_i \right)$$

na želenem zaprtem intervalu $[a, b]$ (v navodilih $[a, b] = [0, 1]$, $r = 7$). Očitno je, da zgoraj zapisano ne formulira linearnega programa – funkciji max in max nista linearni. Formulacijo je bilo zato potrebno še malo spremeniti. Pri tem sva si pomagala z dejstvom, da so trikotne funkcije, s katerimi imava opravka, odsekoma linearne – ekstreme lahko dosežejo le v točkah, kjer se prelomijo. Ta lastnost se zaradi linearnosti prenese tudi na njihovo vsoto. Ta lahko ekstreme doseže le v točkah, kjer se prelomi neka funkcija, ki je del te vsote.

1.3 Programsko okolje in implementacija

Za implementacijo problema sva se odločila za uporabo programskega jezika `Sage`, ker ima že vgrajeno podporo za celoštevilsko linearno programiranje. V osnovi sloni na programskem jeziku `Python`, z dodatno podporo za matematiko, nastal pa je kot alternativa programskemu jeziku `Mathematica`. Programirala sva na platformi `CoCalc`, kjer sva datoteke pretvorila v `Jupyter Notebooks` obliko.

2 Ideja rešitve z linearnim programiranjem LP

Kot omenjeno v uvodu, je ideja naslednja: iz nabora funkcij $\{f_1, \ldots, f_r\}$ želimo torej izbrati tako podmnožico $\{g_1, \ldots, g_r\}$, $r \leq 7 < n$, da bo dosežen

$$\min_{\{g_1,\ldots,g_7\}\subseteq\{f_1,\ldots,f_n\}} \left(\max_{x\in[a,b]} \sum_{i=1}^{r} g_i - \min_{x\in[a,b]} \sum_{i=1}^{r} g_i \right)$$

na želenem zaprtem intervalu $[a, b]$. Glede na to, da imamo opravka s trikotnimi funkcijami, ki so odsekoma linearne, pa lahko problem poenostavimo. Iz odsekoma linearnosti tako funkcij kot njihove vsote sledi, da bo vsota lahko dosegla ekstrem le na robovih intervala ali na mestu prelomu ene izmed funkcij. Vsoto lahko zato namesto na celotnem intervalu $[a, b]$ ocenimo le na točkah preloma. Definiramo torej množico testnih točk $\mathcal{B} = \{x_1, \ldots, x_k\}$, v kateri so vsebovane točke preloma obravnavanih funkcij iz intervala $[a, b]$ ter robni točki $\{a\}$ in $\{b\}$. Da lahko v teh točkah ocenimo vsoto, moramo izračunati vrednosti vseh funkcij f_j, $j \in [n]$, v vseh točkah množice $\mathcal{B}$. Zgornjo formulacijo problema lahko prepišemo kot

$$\min \left(\max_{x_i\in\mathcal{B}} \sum_{j=1}^{n} f_j(x_i)v_j - \min_{x_i\in\mathcal{B}} \sum_{j=1}^{n} f_j(x_i)v_j \right),$$

kjer v_j definiramo kot

$$v_j = \begin{cases} 1\,; & f_j \in \{g_1, \ldots, g_r\}, \\ 0\,; & f_j \notin \{g_1, \ldots, g_r\}. \end{cases}$$

Zdaj se moramo znebiti še funkcij max in min. Izračunamo lahko vrednost $\sum_{j=1}^{n} f_j(x_i)v_j$ za vsak i in definiramo vrednosti $M, m \in \mathbb{R}$, taki, da M navzgor omeji dane vsote, m pa navzdol.

Zdaj lahko zapišemo sledeč linearni program:

$$
\begin{aligned}
&\min(M-m)\\
&\forall j \in \{1,\dots,n\}:\ 0 \le v_j \le 1,\ v_j \in \mathbb{Z}\\
&\forall i \in \{1,\dots,k\}:\ \sum_{j=1}^{n} f_j(x_i)v_j \le M\\
&\forall i \in \{1,\dots,k\}:\ \sum_{j=1}^{n} f_j(x_i)v_j \ge m\\
&\sum_{j=1}^{n} v_j \ge 1\\
&\sum_{j=1}^{n} v_j \le 7
\end{aligned}
$$

Ker sta M in m realnoštevilski omejitvi, ta linearen program ni celoštevilski ampak mešano-celoštevilski.

3 Implementacija

3.1 Generiranje trikotnih funkcij

Definicija trikotne funkcije ni pretirano striktno določena, ideja pa je naslednja. Funkcija je konstantno enaka 0 na intervalu $(-\infty, x_1)$, kjer x_1 predstavlja levi rob trikotnika. Na intervalu $[x_1, x_2]$, kjer je $(x_2, y(x_2))$ vrh trikotnika, je definirana kot linearna funkcija s pozitivnim naklonom, ki potuje skozi točki $(x_1, 0)$ ter $(x_2, y(x_2))$. Na intervalu od x_2 do x_3, kjer je x_3 desni rob trikotnika, je zopet definirana kot linearna funkcija, tokrat z negativnim naklonom ter potuje skozi $(x_2, y(x_2))$ in $(x_3, 0)$. Na intervalu (x_3, ∞) je zopet konstantno enaka 0. Najbolj preprost primer bi bil morda $f(x) = \max\{1 - |x|, 0\}$.

Pri samem generiranju trikotnih funkcij sva se odločila izkoristiti kar najin izbran način spopadanja s linearnim programom, pri katerem funkcijo oceniva le v končnem naboru točk preloma. Zato sva se odločila, da bova funkcije definirala preko njihovih točk preloma. Ideja je preposta: za dobljeno trojico točk $(x_1, 0), (x_2, y_2), (x_3, 0) \in \mathbb{R}^2$, $x_1, x_2, x_3, y_1 \in \mathbb{R}$, $x_1 < x_2 < x_3$, $y_2 > 0$, lahko trikotno funkcijo f definiramo kot

$$
f(x) \;=\; \begin{cases} 0\,; & x \in (-\infty, x_1) \cup [x_3, \infty),\\ \frac{y_2}{x_2 - x_1}x - \frac{y_2 x_1}{x_2 - x_1}\,; & x \in [x_1, x_2),\\ \frac{-y_2}{x_3 - x_2}x + \frac{y_2 x_3}{x_3 - x_2}\,; & x \in [x_2, x_3). \end{cases}
$$

Točke sva generirala naključno, s pomočjo `Python`-ove knjižnice `random`. Definirala sva razred `TrikotnaFunkcija`, ki generira vrednosti po enakomerni zvezni porazdelitvi, $x_1, x_2, x_3 \sim \mathcal{U}([0, 1])$, $y_2 \sim \mathcal{U}([0, 100])$, ter jih nato uredi v točke $(x_1, 0), (x_2, y_2), (x_3, 0)$.