Lineārā programmēšana IV

Šajā lekcijā tiks apskatīta iekšējā punkta metode priekš lineārās programmēšanas. Tā atrod atrisinājumu polinomiālā laikā un tā arī diezgan labi strādā praksē (atkarībā no konkrētās situācijas, dažreiz pārspējot simpleksa algoritmu, citreiz zaudējot tam).

1. Pamatidejas

Atšķirībā no simpleksmetodes, kas pārmeklē pieļaujamā apgabala stūrus, iekšējā punkta metode meklē labākus un labākus atrisinājumus pieļaujamā apgabala iekšienē, tuvojoties stūrim tikai algoritma beigās.

Tas var tikt realizēts divos veidos:

1. Modificējot mērķfunkciju tā, lai tās vērtība kļūtu sliktāka pieļaujamā apgabala malās. Piemēram, mērķfunkciju max $c_1 x_1 + c_2 x_2 + ... + c_n x_n$ var aizstāt ar

   $$\max (c_1 x_1 + c_2 x_2 + ... + c_n x_n + \ln x_1 + ... + \ln x_n).$$

   Tad, tuvojoties x_i=0 plaknēm, kas ierobežo pieļaujamo apgabalu, $\ln x_1$ tiecas uz $-\infty$ un mērķfunkcija arī tieksies uz $-\infty$.
2. Ieviešot papildus nosacījumus, kas mūs attur no pieļaujamā apgabala malām.

Ar katru soli, papildus nosacījumi tiek vājināti, ļaujot algoritmam pietuvoties stūrim, kur ir sākotnējas mērķfunkcijas maksimālā vērtība. Mērķis ir panākt, lai algoritms sākumā atrod optimālo vērtību pieļaujamā apgabala iekšienē un tad nonāk optimālajā stūrī.

Iekšējā punkta metodei ir trīs galvenie varianti:

1. Afīnā skalēšana – teorētiska ātrdarbības novērtējuma nav, bet praksē strādā diezgan labi;
2. Potenciāla redukcija – pierādāmi strādā laikā O(n L), n – mainīgo skaits, L – precizitāte bitos, ar kādu jāatrod atrisinājums;
3. Centrālā trajektorija - pierādāmi strādā laikā $O(L \sqrt{n})$.

Tālāk mēs sīkāk apskatīsim pirmo no šiem variantiem.

2. Pārskats par afīnās skalēšanas metodi

Tiks pieņemts, ka lineārās programmēšanas problēma ir standartformā, kurā vienīgās nevienādības ir $x_i \geq 0$ (tāpat kā tas bija simpleksalgoritma izklāstā). Šādu lineāru programmu ērti pierakstīt matricu formā.

Apzīmējam

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ \dots \\ x_n \end{pmatrix} \quad A = \begin{pmatrix} a_{11} & a_{12} & & a_{1n} \\ \dots & \dots & & \dots \\ a_{m1} & a_{m2} & & a_{mn} \end{pmatrix} \quad b = \begin{pmatrix} b_1 \\ \dots \\ b_m \end{pmatrix}$$

Tad lineārā programma izskatās šādi:

Maksimizēt $c_1 x_1 + c_2 x_2 + ... + c_n x_n$
pie nosacījumiem $A x = b$, $x_1 \geq 0$, $x_2 \geq 0$, ..., $x_n \geq 0$.

Tāpēc $A x = b$ ietilpst visas vienādības, kas bija starp simpleksalgoritma nosacījumiem.

Afīnās skalēšanas metode darbojas šādi:

1. Sākam ar kaut kādu lineārās programmas atrisinājumu $x = (x_1, x_2, ..., x_n)$.
2. Atkārto šādu darbību virkni:
   a. Novelk elipsoīdu ap tekošo atrisinājumu $x = (x_1, x_2, ..., x_n)$, kas pieskaras visām plaknēm $x_i=0$.
   b. Atrod, kurā elipsoīda punktā mērķfunkcija ir maksimāla. (Šajā solī tiek ignorēti nosacījumi $Ax=b$ un maksimums rēķināts pār visiem elipsoīda punktiem, ieskaitot tos, kur šie nosacījumi neizpildās.) Atrasto maksimumu apzīmē ar $x' = (x'_1, x'_2, ..., x'_n)$.
   c. Projicē vektoru $x' - x = (x'_1 - x_1, x'_2 - x_2, ..., x'_n - x_n)$ uz plakni $Ax=0$. Iegūto projekciju apzīmē ar $z = (z_1, z_2, ..., z_n)$.
   d. Tagad, jebkuram skaitlim a, vektors $x + a z$ apmierina nosacījumus $Ax = b$. Mēs aprēķinam maksimālo a, pie kura $x_i + a z_i \geq 0$.
   e. Mūsu jaunais atrisinājums būs $x' = (x'_1, x'_2, ..., x'_n)$, kur
   $$x'_i = x_i + 0.96\, a\, z_i.$$

3. Afīnās skalēšanas metodes realizācija pa soļiem

3.1. Elipsoīda novilkšana

Pašreizējo atrisinājumu apzīmēsim ar $t = (t_1, t_2, ..., t_n)$. Ja $x = (x_1, x_2, ..., x_n)$ ir patvaļīgs punkts, tad mēs varam aizstāt koordinātes x_i ar jaunām koordinātēm y_i, definējot

$$x_i = y_i + t_i.$$

Tad jaunajā koordinātu sistēmā lineārā programma pārtop par

Maksimizēt $c_1 y_1 + c_2 y_2 + ... + c_n y_n$
pie nosacījumiem $A y = 0$, $y_1 \geq -t_1$, $y_2 \geq -t_2$, ..., $y_n \geq -t_n$.

Elipsoīda, kas pieskarās visām $y_i \geq - t_i$ plaknēm, vienādojums ir:

$$\frac{y_1^2}{t_1^2} + ... + \frac{y_n^2}{t_n^2} = 1 .$$

Tagad veicam vēl vienu koordinātu transformāciju, aizstājot y_i ar z_i, tā lai

$$y_i = t_i z_i.$$

Tad mēs iegūstam lināru programmu

Maksimizēt $c_1 t_1 z_1 + c_2 t_2 z_2 + ... + c_n t_n z_n$

pie nosacījumiem $A\begin{pmatrix} t_1 z_1 \\ ... \\ t_n z_n \end{pmatrix} = \begin{pmatrix} 0 \\ ... \\ 0 \end{pmatrix}$, $z_1 \geq -1$, $z_2 \geq -1$, ..., $z_n \geq -1$.

Elipsoīds tagad pārtop par n dimensiju sfēru ar vienādojumu

$$z_1^2 + ... + z_n^2 = 1.$$

3.2. Maksimuma noteikšana uz elipsoīda

Maksimums uz sfēras tiek sasniegts, ja vektors $(z_1, z_2, ..., z_n)$ ir paralēls $(c_1 t_1, c_2 t_2, ..., c_n t_n)$, t.i., ja

$$z_i = c\, c_i t_i,$$

kur c ir pozitīvs skaitlis, kas izvēlēts tā, lai vektoram $(z_1, z_2, ..., z_n)$ būtu pareizais garums (tas ir, 1).

3.3. Projicēšana

Ja mēs gribam projicēt vektoru $z = (z_1, z_2, ..., z_n)$ uz plakni $Az=0$, tad projekciju var aprēķināt pēc šādas formulas:

$$p = (I-B^T(BB^T)^{-1}B)\, z.$$

Sarežģītākais solis šeit ir apgrieztās matricas aprēķināšana. No tā var izvairīties šādi:

1. Ņem vienādojumu $Bz = (B B^T)\, w$, kur w – nezināmo vektors. Atrisina šo vienādojumu. Iegūtajais vektors w būs vienāds ar

$$w = (BB^T)^{-1}B\, z.$$

2. Projekcija tad būs vienāda ar $p = z - B^T w$.

3.4. Jaunā punkta noteikšana

Pēc 3.1. nodaļā izdarītajām koordinātu transformācijām esošais atrisinājums pārvērtās par (0, 0, ..., 0) jaunajās koordinātēs un nosacījumi – nevienādības par $z_1 \geq -1$, $z_2 \geq -1$, ..., $z_n \geq -1$.

Novelkam taisni caur (0, 0, ..., 0) un $p = (p_1, ..., p_n)$. Šo taisni var aprakstīt ar vienādojumu

$$z_1 = a\, p_1,\ z_2 = a\, p_2,\ ...,\ z_n = a\, p_n,$$

kur t-parametrs. Nosakām mazāko pozitīvo t, kuram z_i=-1. Tas būs pirmais krustpunkts starp mūsu nupat novilkto taisni un pieļaujamā apgabala robežām. Mūsu jaunais atrisinājums būs

$$z_1 = 0.96\, a\, p_1,\ z_2 = 0.96\, a\, p_2,\ ...,\ z_n = 0.96\, a\, p_n.$$

3.5. Pāreja uz veco koordinātu sistēmu

Atrisinājums (z_1, z_2, ..., z_n) jāpārveido uz sākotnējo koordinātu sistēmu (x_1, x_2, ... x_n).

4. Piemērs afīnas skalēšanas metodei

Apskatām lineārās programmēšanas uzdevumu:

maksimizēt $-\frac{x_1}{3} + x_2$, pie nosacījumiem

$x_1 + x_2 \leq 4$

$2x_1 - x_2 \leq 3$

$x_1 \geq 0$, $x_2 \geq 0$

Pielietosim afīnās skalēšanas metodi ar sākumpunktu x_1=1, x_2 =1.

4.1. Pārveide standartformā

Sākumā lineārās programmēšanas problēma jāpārveido uz standartformu, kurā vienīgās nevienādības ir $x_i \geq 0$. To var izdarīt līdzīgi kā simpleksalgoritma gadījumā. Mūsu konkrētajai lineārajai programmai standartforma būs:

maksimizēt $-\frac{x_1}{3} + x_2$, pie nosacījumiem

$x_1 + x_2 + x_3 = 4$

$2x_1 - x_2 + x_4 = 3$

$x_1 \geq 0$, $x_2 \geq 0$, $x_3 \geq 0$, $x_4 \geq 0$

Mūsu sākumpunkts tagad ir x_1=1, x_2 =1, x_3 =2, x_4 =2.

4.2. Pirmā koordinātu pārbīde

Aizstājam koordinātas ar x_1=1+y_1, x_2 =1+y_2, x_3 =2+y_3, x_4 =2+y_4, tā, lai sākumpunkts kļūtu par $y_1 = y_2 = y_3 = y_4 = 0$. Tad lineārā programma pārvēršas par

maksimizēt $-\frac{y_1}{3} + y_2$, pie nosacījumiem

$y_1 + y_2 + y_3 = 0$

$2y_1 - y_2 + y_4 = 0$

$y_1 \geq -1,\ y_2 \geq -1,\ y_3 \geq -2,\ y_4 \geq -2$

Piezīme. Sākotnēji, maksimizējamā izteiksme ir

$$-\frac{x_1}{3} + x_2 = -\frac{1 + y_1}{3} + 1 + y_2 = -\frac{y_1}{3} + y_2 + \frac{2}{3}.$$

Saskaitāmo 2/3 var ignorēt, jo tas ir nemainīgs. Tāpēc, $-\frac{y_1}{3} + y_2$ un $-\frac{y_1}{3} + y_2 + \frac{2}{3}$ maksimumu sasniedz pie vienām un tām pašām y_i vērtībām.

4.3. Otrā koordinātu pārbīde

Aizstājam koordinātas ar $y_1 = z_1$, $y_2 = z_2$, $y_3 = 2z_3$, $y_4 = 2z_4$, tā, lai sākumpunkts kļūtu par $z_i \geq -1$. Tad lineārā programma pārvēršas par

maksimizēt $-\frac{z_1}{3} + z_2$, pie nosacījumiem

$z_1 + z_2 + 2z_3 = 0$

$2z_1 - z_2 + 2z_4 = 0$

$z_1 \geq -1,\ z_2 \geq -1,\ z_3 \geq -1,\ z_4 \geq -1$

Elipsoīds tagad ir vienkārši sfēra

$$z_1^2 + z_2^2 + z_3^2 + z_4^2 = 1.$$

4.4. Maksimums

Maksimums uz sfēras tiek sasniegts virzienā

$$(z_1, z_2, z_3, z_4) = \left(-\frac{1}{3}, 1, 0, 0\right).$$

4.5. Projekcijas aprēķins

Tagad mums jāprojicē šis virziens uz plakni

$z_1 + z_2 + 2z_3 = 0$

$2z_1 - z_2 + 2z_4 = 0$

z_i koeficientu matrica mums ir

$$B = \begin{pmatrix} 1 & 1 & 2 & 0 \\ 2 & -1 & 0 & 2 \end{pmatrix}$$

Tās transponētā matrica ir $B^T = \begin{pmatrix} 1 & 2 \\ 1 & -1 \\ 2 & 0 \\ 0 & 2 \end{pmatrix}$

Aprēķinam

$$B \cdot B^T = \begin{pmatrix} 6 & 1 \\ 1 & 9 \end{pmatrix}.$$

Tagad mums jāatrisina vienādojums

$$\begin{pmatrix} 6 & 1 \\ 1 & 9 \end{pmatrix}\begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = B\begin{pmatrix} -\frac{1}{3} \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 2/3 \\ -5/3 \end{pmatrix}$$

Tas atbilst vienādojumu sistēmai

$$\begin{cases} 6w_1 + w_2 = \dfrac{2}{3} \\ w_1 + 9w_2 = -\dfrac{5}{3} \end{cases}$$

Šīs vienādojumu sistēmas atrisinājums ir

$$w_1 = \frac{23}{159}, w_2 = -\frac{32}{159}.$$

Projekcija ir vienāda ar

$$p = z - B^T w = \begin{pmatrix} -1/3 \\ 1 \\ 0 \\ 0 \end{pmatrix} - \begin{pmatrix} 1 & 2 \\ 1 & -1 \\ 2 & 0 \\ 0 & 2 \end{pmatrix}\begin{pmatrix} 23/159 \\ -32/159 \end{pmatrix} = \begin{pmatrix} -12/159 \\ 104/159 \\ -46/159 \\ 64/159 \end{pmatrix}$$

4.5. Krustpunkts ar robežnosacījumiem

Novelkam taisni virzienā p. Šo taisni var uzdot šādi:

$z_1 = -12\ t$, $z_2 = 104\ t$, $z_3 = -46\ t$, $z_4 = 64\ t$, kur t-parametrs.

Tagad jānosaka pirmais krustpunkts starp šo taisni un plaknēm $z_i \geq -1$ (virzienā $t \geq 0$). Tas ir $t = 1/46$, kur mūsu taisne krusto $z_3 \geq -1$. Tātad jaunais atrisinājums būs

$$z_1 = -0.96 \times 12 \times \frac{1}{46};$$

$$z_2 = 0.96 \times 104 \times \frac{1}{46};$$

$$z_3 = -0.96 \times 46 \times \frac{1}{46};$$

$$z_4 = 0.96 \times 64 \times \frac{1}{46};$$

Pārveidojot atpakaļ uz sākotnējās koordinātēm, iegūstam $x_1=0.75$, $x_2=3.17$.