Robusztus Optimalizálás

Mályusz Attila Edmund
2022.05.10

Bevezetés

Lineáris és egészértékű programozás tanulmányaim során a megoldandó feladatok estén mindig feltételeztük, hogy az adott elemek pontosan kimért adatok. Ez viszont hibára adhat lehetőséget, hiszen az adatok legtöbbször mért adatok, melyek így pontatlanok lehetnek. Mért adatokkal kapcsolatban felmerülő problémák alatt olyanra kell gondolni, hogy egyzerűen mérési hiba (fizikai méréseknél ez szinte mindig előfordul), vagy az adott adat eleve csak közelítve van statisztikai módszerekkel, mint példáúl kereslet adott termékekre. Így természetesen merül fel, hogy változó adatok mellett a feladatra adott megoldásunk még mindig kielégítse a feltételeket és ezeken belül optimális legyen, azaz robusztus.

Régebbi modellek

Az első próbálkozás ezen feladat megoldására Soystertől jött [5]. Soyster egy olyan lineáris optimalizálási modellt adott melyben a megoldásnak megengedettnek kellett lennie minden olyan adatra amely egy konvex halmazból származik. A kapott modellel viszont az volt a probléma, hogy túlságosan messze került a robusztus optimális megoldás a kapott adatokból számolt optimális megoldástól.

$$\begin{aligned} \text{maximize} \quad & c'x \\ \text{subject to} \quad & \sum_{j=1}^{n} A_j x_j \leq b, \forall A_j \in K_j, j = 1, \ldots, n, \\ & x \geq 0 \end{aligned}$$

Ahol K_j konvex halmazok a bizonytalansági halmazok és A_j pedig az oszlopai a mátrixnak. Soyster megmutatta, hogy a fenti feladat megegyezzik a következő lineáris programmal.

$$\begin{aligned} \text{maximize} \quad & c'x \\ \text{subject to} \quad & \sum_{j=1}^{n} \bar{A}_j x_j \leq b, \\ & x \geq 0 \end{aligned}$$

, ahol $\bar{a_{ij}} = sup_{A_j \in K_j}(A_{ij})$.
Egymástól függetlenül Ben-Tal és Nemirovski [1], El-Ghaoui [4], is azon dolgozott, hogy

a robusztus feladat optimális megoldás értéke ne legyen túl távol a kapott értékekből számolt optimumtól. Az ő megoldásukban ellipszioid bizonytalanságot vizsgáltak amit visszavezettek kvadratikus optimalizálási problémára. Ezen módszer problémája az volt, hogy lassabb lett a megoldás kiszámolása a kvadratikus feladat miatt.

Berstimas-Sim modell

A félév során a Bertsimas-Sim [3], [2] féle modellel foglalkoztam. Ehhez tekintsük a következő lineáris optimalizálási feladatot.

$$\begin{aligned} \text{maximize} \quad & c'x \\ \text{subject to} \quad & Ax \leq b \quad , \\ & l \leq x \leq u \end{aligned}$$

1. Állítás. *[3] Feltehetjük, hogy a bizonytalanság egyedül az A mátrixot érinti.*

Bizonyítás. Tegyük fel, hogy b_i bizonytalan, ekkor $a_i x - b_i z \leq$ 1-et hozzádva és z-t 1-re rögzítve kiküszöböljük a bizonytalanságot.
Tegyük fel, hogy c-ben szerepelnek bizonytalan eggyüthatók, ekkor $z - cx \leq 0$ és a maximalizálandó kifejezést z-re cserélve megint kiköszöbültük a bizonytalanságot.

A következő féleképpen fogjuk modellezni a bizonytalanságot a mátrixban. Vegyük A egy sorát és legyen J_i azon eggyüthatók halmaza az i. sorban melyek bizonytalanok. Minden a_{ij}, $j \in J_i$ eggyüttható egy $\bar{a}_{ij}$ random változó, mely a $[a_{ij} - \hat{a}_{ij}, a_{ij} + \hat{a}_{ij}]$ intervallumból vesz felértéket és szimmetrikus az adott intervallumra. Ezen kívűl még definiáljuk a $\eta_{ij} = (\bar{a}_{ij} - a_{ij})/\hat{a}_{ij}$ random változót mely szimmetrikus és $[0, 1]$ intervallumból veszi fel értékeit.
Minden i-re bevezetünk egy Γ_i változót, mely a $[0, |J_i|]$ intervallumból vesz fel értékeket. Γ_i az feladat robusztusságát alakítja, mégpedig azt jelenti, hogy az adott sorban $\lfloor \Gamma_i \rfloor$ maximum mennyi együttható változtatja meg értékét és egy a_{ij} együttható $(\Gamma_i - \lfloor \Gamma_i \rfloor)\hat{a}_{ij}$-vel.
A következőkben megmutatjuk, hogy az így kapott problémát megtudjuk oldani lineáris programozással. Ezen kívűl az is igazolható, hogy ha több mint $\lfloor \Gamma_i \rfloor$ együttható változik, akkor is még megengedett a kapott megoldásunk nagy valószínűséggel.
Tekintsük a következő jelenleg még nem lineáris formalizációját a problémának (1).

$$\begin{aligned} \text{maximize} \quad & c'x \\ \text{subject to} \quad & \sum_j a_{ij} x_j + \max_{\{S_i \cup t_i | S_i \subset J_i, |S_i| = \lfloor \Gamma_i \rfloor, t_i \in J_i \setminus S_i\}} \{\textstyle\sum_{j \in S_i} \hat{a}_{ij} y_j + (\Gamma_i - \lfloor \Gamma_i \rfloor)\hat{a}_{ij} y_t\} \leq b_i, \forall i, \\ & -y_j \leq x_j \leq y_j, \\ & l \leq x \leq u, \\ & y \geq 0 \end{aligned}$$

Legyen $\beta_i(x,\Gamma_i) = max_{\{S_i|S_i\subset J_i, S_i=|J_i|\}} \sum_{j\in S_i} \hat{a}_{ij}|x_j|$. Ha $\Gamma_i = 0$, akkor $\beta_i(x,\Gamma_i) = 0$. Ha pedig $\Gamma_i = |J_i|$, akkor Soyster modelljét kapjuk vissza.
Azért, hogy a feladatot LP feladatként tudjuk megfogalmazni a következő állításra van szükségünk.

2. Állítás. *[3] Legyen adva egy x^* vektor, ekkor $\beta_i(x,\Gamma_i) = max_{\{S_i|S_i\subset J_i, S_i=|J_i|\}} \sum_{j\in S_i} \hat{a}_{ij}|x_j|$ megeggyezik a következőz lineáris programozási feladat optimum értékével*

$$\begin{aligned} \beta_i(x,\Gamma_i) = \text{maximize} \quad & \sum_{j\in S_i} \hat{a}_{ij}|x_j^*|z_{ij} \\ \text{subject to} \quad & \sum_{j\in J_i} z_{ij} \le \Gamma_i, \\ & 0 \le z_{ij} \le 1 \ \ \forall j \in J_i \end{aligned}$$

Bizonyítás. Az LP feladat megoldása úgy fog kinézni, hogy $\lfloor\Gamma_i\rfloor$ db 1-es és egy változó $\Gamma_i - \lfloor\Gamma_i\rfloor$. Ez pedig ekvivalens a β_i definíciójában lévő halmaz kiválasztásához.

3. Tétel. *[3] (1)-nek a következő lineáris programizási feladat felel meg:*

$$\begin{aligned} \text{maximize} \quad & c'x \\ \text{subject to} \quad & \sum_j a_{ij}x_j + z_i\Gamma_i + \sum_{j\in J_i} p_{ij} \le b_i, \forall i, \\ & z_i + p_{ij} \ge \hat{a}_{ij}y_j \qquad && \forall i, j \in J_i, \\ & -y_j \le x_j \le y_j && \forall j, \\ & l_j \le x_{jj} && \forall j, \\ & p_{ij} \ge 0 && \forall i, j \in J_i, \\ & y_j \ge 0 && \forall j, \\ & z_i \ge 0 && \forall i \end{aligned}$$

A kapott lineáris programozási feladatnak $n+k+1$ változója és $m+k+n$ feltétele van, ahol $k = \sum_i |J_i|$.

Egészértékű modell

Ha esetleg egészértékű változós az x vektorunk vagy csak néhány változója egész. Akkor egy nagyon hasonló formalizációt tudunk felírni (továbbiakért lásd. [2]).
A továbbiakban kombinatorikus problémák robusztus változatait nézzük meg. Adott $X \subseteq \{0,1\}^n$ halmaz és a következő opimalizlálsi feladat

$$\begin{aligned} \text{maximize} \quad & \widetilde{c}x \\ \text{subject to} \quad & x \in X \end{aligned}$$

, ahol $\widetilde{c}$ minden $\widetilde{c}_j$ eleme a $[c_j, c_j + d_j]$, $d_j \geq 0$, $j \in N = \{1, 2, \ldots n\}$ intervallumból veszi fel értékeit. Szeretnénk egy $x \in X$ megoldást találni amely minimalizálja a cx költséget, ha legfeljebb Γ együttható változhat meg. Azaz (2)

$$\begin{aligned} Z^* = \text{maximize} \quad & \widetilde{c}x + max_{\{S|S\subseteq N, |S|\leq\Gamma\}} \sum_{j\in S} d_j x_j \\ \text{subject to} \quad & x \in X \end{aligned}$$

A továbbiakban feltesszük, hogy d_i csökkenő sorrendbe vannak rendezve, azaz $d_1 \geq d_2 \geq \cdots \geq d_n$ és legyen $d_{n+1} = 0$. Ezen feladattal olyan problémakat vettünk egy kalap alá, mint legrövidebb út, legolcsóbb feszítőfa, matroid metszet, utazó ügynök probléma. A következő tétel azt mutatja meg, hogy ha polinomiális az alap problémánk, akkor a robusztus változatot is polinomiális időben tudjuk megoldani.

4. Tétel. *A (2) feladat megoldása megkapható $n+1$ feladat megoldásából:*
$Z^* = min_{l=1,\ldots,n+1} G^l$, *ahol*

$$\begin{aligned} G^l = \text{maximize} \quad & \Gamma d_l + min(\widetilde{c}x + \sum_{j=1}^{l} (d_j - d_l) x_j) \\ \text{subject to} \quad & x \in X \end{aligned}$$

Van mség egy kiváló tulajdonsága a (2)-es robusztus optimalizálási problémának mégpedig, hogy ha az eredeti feladatnak van egy α közelítő algoritmusa, akkor tudunk készíteni egy polinomiális algoritmust amely a robusztus feladatra ad α közelítő megoldást (továbbiakért lásd [2]).

Mérés

A legrövidebb út problémát vettük a [2] cikkből. Adott egy $D = (V \cup \{s, t\}, A)$ gráfunk, c_{ij} $(i, j) \in A$ nem negatív költségekkel az éleken. A legrövidebb utat keressük s-ből t-be. A gráfot és az élköltségeket a következőképpen generáltuk. Vegyük az 2 dimenziós síkot. Legyen az s a $(0, 0)$, t pedig $(1, 1)$ ezek után generáljunk pontokat a $[0, 1] \times [0, 1]$ tégalalban. Legyen c_{ij} az euklideszi távolság i és j által meghatározott pontok között, ha fut él köztük. Legyen d_{ij} pedig γc_{ij} ahol γ egyenletes eloszlású a $[0, 8]$ intervallumon.
$\Gamma = 0, 3, 6, 10$-re mérjük meg a robusztus legrövidebb utat. Azt is mérjük, hogy ha minden élen 0.1 valószínűséggel $c_{ij} + d_{ij}$-re változtatjuk a költséget, akkor az így kapott gráfban mennyi a legrövidebb út.
A kapott méréseink alapján a robusztus legrövidebb út mindig megegyezik $\Gamma = 3, 6, 10$-re, és csak ritkán érhető el az, hogy különbözzön a robusztus megoldás értéke $\Gamma = 6, 10$-ra.
Ha valaki szeretné futtatni a programot, az letöltheti innen `https://github.com/atimaly/Robust_Shortest_Path`.

Irodalomjegyzék

[1] A. Ben-Tal and A. Nemirovski. Robust solutions of linear programming problems contaminated with uncertain data. *Mathematical Programming*, 88:411–424, 01 2000.

[2] D. Bertsimas and M. Sim. Sim, m.: Robust discrete optimization and network flows. math. prog. 98, 49-71. *Mathematical Programming*, 98:49–71, 09 2003.

[3] D. Bertsimas and M. Sim. The price of robustness. *Operations Research*, 52:35–53, 02 2004.

[4] L. El Ghaoui and H. Lebret. Robust solutions to least-squares problems with uncertain data. *SIAM Journal on Matrix Analysis and Applications*, 18(4):1035–1064, 1997.

[5] A. L. Soyster. Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations Research*, 21(5):1154–1157, 1973.