MEGHIÚSULÁSOK KOMPENZÁLÁSA LAKOSSÁGI FELVÉTELEKBEN: EGY SPECIÁLIS LINEÁRIS INVERZ PROBLÉMA[1]

MIHÁLYFFY LÁSZLÓ
Központi Statisztikai Hivatal

1. Bevezetés

A dolgozatban vizsgált probléma matematikai szempontból a lineáris inverz problémák körébe tartozik, amelyekről Csiszárnak a Szigmában nemrégen megjelent dolgozata [2] nyújt részletes áttekintést. A javasolt megoldás Darroch és Ratcliff „SMART" algoritmusával rokon – amelyről az említett dolgozat ugyancsak beszámol –, és annál valamivel egyszerűbb. Formális szempontból adva van egy nemnegatív elemű $m \times n$-es $A = (a_{ij})$ mátrix, egy pozitív elemekből álló n-dimenziós $x^0 = (x_1^0, x_2^0, \ldots, x_n^0)^T$ vektor, továbbá egy ugyancsak pozitív elemekből álló m-dimenziós $b = (b_1, b_2, \ldots, b_m)^T$ vektor, és egy olyan pozitív (vagy nemnegatív) elemekből álló $x = (x_1, x_2, \ldots, x_n)^T$ vektort keresünk, amelyre

$$Ax = b \tag{1.1}$$

és amely valamilyen értelemben „jól közelíti" az x^0 vektort. A megoldás triviális akkor, ha x^0 kielégíti ezt az egyenletet, a gyakorlatban azonban rendszerint nem ez a helyzet.

Valószínűségi mintákból, különösképpen lakossági mintákból való becslés esetén a szóban forgó probléma a következőképpen vetődik fel. Egy ilyen minta alapján értékösszeget, más szóval létszámadatot általában a

$$\sum_i w_i Y_i \tag{1.2}$$

alakban becslünk, ahol Y_i a vizsgált ismérv értéke a minta i-edik elemére (egységére) vonatkozóan, w_i az ehhez tartozó mintasúly (rendszerint az egység kiválasztási valószínűségének a reciproka), és az összegzést a minta összes

[1]Beérkezett 1994. október 10. A szerző köszönettel tartozik Budavári Péternek, akinek észrevételei lehetővé tették a kézirat néhány pontatlanságának kijavítását.

elemére vonatkozóan kell elvégezni. Mintán ebben az összefüggésben egyaránt érthetünk országos mintát, vagy annak egy jól meghatározott részét, például egy megyei részmintát. A mintavételi hibát is figyelembe véve az (1.2) összefüggés torzítatlan becslést eredményezne, ha nem lenne meghiúsulás, azaz, ha a minta minden egységére vonatkozóan sikerülne megszerezni a szükséges információt. A lakossági mintákra azonban világszerte jellemző az esetenként kisebb, máskor meg éppen jelentős mértékű válaszkimaradás, meghiúsulás, amit a w_i mintasúlyok korrekciójával szoktak ellensúlyozni. Általánosan elterjedt az a megoldás, miszerint a (súlyozott) minta nemek és korcsoportok szerinti megoszlását valamilyen területi részletezésben a megfelelő sokaságbeli megoszlás(ok)hoz igazítják, rendszerint a népszámlálás továbbvezetett adatainak a felhasználásával. Bizonyos esetekben a korrekciót nem a megoszlások, hanem a létszámadatok szintjén végzik; ez azt jelenti, hogy a nemek, korcsoportok, illetve földrajzi egységek által meghatározott cellákban a létszámadatok korrigált becslése megegyezik a továbbvezetett népszámlálási adatokkal. Az ilyen korrekciós eljárásoknak általában az a hátránya, hogy a minta háztartásain belül az egyes személyekhez különböző korrigált súlyokat rendelnek, ami rendkívül megnehezíti a személyekre, illetve a háztartásokra vonatkozó adatok/táblázatok összhangjának biztosítását. Ezen a következőképpen lehet segíteni.

Egyszerűség kedvéért tegyük fel, hogy egy Nógrád megyei mintával van dolgunk, és a korrekcióban a 0–4, 5–9, 10–14, ..., 70–74, 75–X öt évenkénti korcsoportokat fogjuk alkalmazni. A minta minden egyes j háztartásához egy

$$Y_{\cdot j} = (Y_{1j}, \ldots, Y_{16j}, Y_{17j}, \ldots, Y_{32j})^T$$

vektort rendelünk úgy, hogy $1 \leq i \leq 16$ esetén Y_{ij} az i-edik korcsoporthoz tartozó férfiak (fiúk), Y_{i+16j} pedig az i-edik korcsoporthoz tartozó nők (lányok) száma ebben a háztartásban. Legyen w_j^0 a j-edik háztartáshoz tartozó *korrigálatlan* mintasúly (ez a háztartás minden tagjára nézve azonos), és $1 \leq i \leq 16$ esetén jelölje N_i, illetve N_{i+16} az i-edik korcsoporthoz tartozó férfiak, illetve nők számát az adott megyében,

$$\mathbf{N} = (N_1, \ldots, N_{32})^T .$$

Jelölje J azoknak a háztartásoknak a számát, amelyekről legalább egy kitöltött kérdőív készült (Nógrádban $J \approx 600$ a munkaerőfelmérésben); a meghiúsulások miatt

$$\sum_{j=1}^{J} w_j^0 Y_{\cdot j} \neq \mathbf{N}$$

vagy

$$\mathbf{Y} w^0 \neq \mathbf{N} , \tag{1.3}$$

ahol $Y = (Y_1, \ldots, Y_J)$, $w^0 = (w_1^0, \ldots, w_J^0)$, és itt az eltérés jóval meghaladja a mintavételi hibával magyarázható mértéket. Értelemszerűen olyan korrigált $w_1, w_2, \ldots, w_J$ súlyokat keresünk, amelyek pozitívok, az eredeti $w_1^0, w_2^0, \ldots, w_J^0$ súlyokat jól közelítik, és kielégítik az (1.3) egyenletet. Az $\mathbf{Y} \sim A$, $w \sim x$, $w^0 \sim x^0$, $\mathbf{N} \sim b$ helyettesítésekkel a korrekciós probléma a standard lineáris inverz problémába megy át. A dolgozatban felváltva használjuk a kétféle jelölésrendszert aszerint, hogy mikor melyik célravezető.

2. Algoritmus

A bevezetésben leírt feladat megoldásának kézenfekvő eszköze lehetne a kvadratikus programozás, pl. a

$$\sum_{j=1}^{J}(w_j - w_j^0)^2$$

célfüggvénnyel, amikor azonban a feladat felvetődött, a megfelelő szoftver nem volt kéznél, s ugyanakkor gyors megoldásra volt szükség. Így jött létre az alábbi algoritmus.

Tekintsük az (1.1) egyenletet. $t = 0$ esetén legyen $x(t) = x^0$, és $t = 0, 1, 2, \ldots$ esetén tegyük a következőket.

$i = 1, 2, \ldots, m$ esetén legyen

$$r_i(t) = \frac{b_i}{\sum_{j=1}^{n} a_{ij} x_j(t)} \; ; \tag{2.1}$$

$j = 1, 2, \ldots, n$ esetén legyen

$$u_j(t) = \frac{\sum_{k=1}^{m} a_{kj} r_k(t)}{\sum_{k=1}^{m} a_{kj}} \; , \tag{2.2}$$

az $r_i(t)$-k súlyozott számtani közepe, és legyen végül $j = 1, 2, \ldots, n$ esetén

$$x_j(t+1) = x_j(t) u_j(t) \; . \tag{2.3}$$

Ellenőrizzük az eljárás konvergenciáját, és szükség esetén folytassuk az eljárást (2.1)-nél. A (2.1)–(2.3) összefüggésekre az alábbiakban úgy is fogunk hivatkozni, mint az algoritmus 1., 2., illetve 3. lépésére.

Az általánosság korlátozása nélkül feltehetjük, hogy

$$\sum_{k=1}^{m} a_{kj} = 1 \tag{2.4}$$

minden j-re, vagyis, hogy az A mátrix oszlopösszegei egységnyiek. (2.1)–(2.3) ekkor a következő kompakt alakba írható:

$$x_j(t+1) = x_j(t) \sum_{k=1}^{m} a_{kj} \frac{b_k}{\sum_{h=1}^{n} a_{kh} x_h(t)} . \tag{2.5}$$

A Darroch-Ratcliff féle SMART algoritmnus ([2], (5.16) összefüggés) jelöléseinkkel az

$$x_j(t+1) = x_j(t) \prod_{k=1}^{m} {b_k}^{a_{kj}} \Big(\sum_{h=1}^{n} a_{kh} x_h(t) \Big)^{-a_{kj}}$$

alakba írható; eszerint javasolt eljárásunk ennél annyival egyszerűbb, amennyivel könnyebb a súlyozott számtani átlagok kiszámítása a súlyozott mértani átlagok kiszámításánál. A kétféle átlag közötti kapcsolat alapján arra is számíthatunk, hogy eljárásunk valamivel gyorsabb lesz a SMART-nál, hiszen minden egyes iterációban valamivel nagyobbat lép.

A SMART algoritmus az (1.1) egyenletnek olyan nemnegatív x megoldását eredményezi – amennyiben ilyen létezik –, amelyre az

$$I(x \| x^0) = \sum_j \big(x_j \log \frac{x_j}{x_j^0} - x_j + x_j^0 \big)$$

I-divergencia minimális. A (2.1)–(2.3) algoritmus által szolgáltatott x megoldással kapcsolatban egyenlőre nincs ilyen eredményünk, tehát nem tudjuk megmondani, hogy x milyen értelemben közelíti az x^0 induló vektort. Erre a kérdésre a 4. fejezetben még visszatérünk.

3. Az eljárás konvergenciája

Egyszerűség kedvéért az iterációs lépés t sorszámát az esetek többségében el fogjuk hagyni; ilyenkor a változók aktuális értékét egyszerűen r_i-vel, x_j-vel, illetve u_j-vel jelöljük, míg a következő iterációhoz tartozó megfelelő értékeket vesszővel különböztetjük meg: r_i', x_j', illetve u_j'. Vezessük be a következő jelöléseket: $i = 1, 2, \ldots, m$ esetén legyen

$$f_i = y_i - b_i + b_i \log(b_i / y_i) = y_i - b_i + b_i \log r_i .$$

Könnyen belátható, hogy az f_i függvények rendelkeznek az alábbi tulajdonságokkal:

- szigorúan konvexek és akárhányszor deriválhatók az $\mathbb{R}_+ \setminus \{0\}$ halmazon (a pozitív valós számok halmazán);

- $f_i(y_i) \to \infty$, ha $y_i \to +0$ vagy ha $y_i \to +\infty$;
- $y_i \in \mathbb{R}_+ \setminus \{0\}$ esetén $f_i(y_i) \geq 0$, és $f_i(y_i) = 0$ akkor és csak akkor, ha $y_i = b_i$.

Az m-változós

$$F(y) = F(y_1, y_2, \ldots, y_m) = f_1(y_1) + f_2(y_2) + \ldots + f_m(y_m)$$

függvény a $D = \{ y \mid y_1 > 0, y_2 > 0, \ldots, y_m > 0 \}$ halmazon ugyancsak nemnegatív, szigorúan konvex – a második parciális deriváltakból álló Hesse-féle mátrixa $H = \operatorname{diag}(b_1/y_1^2, b_2/y_2^2, \ldots, b_m/y_m^2)$ –, plusz végtelenhez tart továbbá, ha legalább egy i-re $y_i \to +\infty$, s végül, $F(y) = 0$ pontosan akkor, ha $f_i(y_i) = 0 \;\; i = 1, 2, \ldots, m$ esetén. Megjegyezzük, hogy $F(y)$ az m-dimenziós b és y vektorok közötti I-divergencia: $F(y) = I(b||y)$.

A $G(x) = G(x_1, x_2, \ldots, x_n)$ függvényt a következőképpen definiáljuk:

$$G(x) = F(Ax)\,.$$

Komponensekre nézve ez azt jelenti, hogy

$$y_i = \sum_{j=1}^{n} a_{ij} x_j\,, \qquad i = 1, 2, \ldots, m\,. \tag{3.1}$$

$G(x)$ konvex – általában nem szigorúan konvex – függvény, a

$$G(x) = \min, \qquad x \in \mathbb{R}_+^n \tag{3.2}$$

feladat tehát konvex programozási feladat. Bár $G(x)$ értelmezési tartománya nem a teljes $\mathbb{R}_+^n = \{ x \in \mathbb{R}^n \mid x \geq 0 \}$ halmaz, hanem annak csupán

$$D_G = \{ x \geq 0 \mid y = Ax \in D \}$$

részhalmaza, (3.2) megfogalmazása korrekt, ugyanis minden olyan esetben, amikor $x \in D_G \to x^* \in \mathbb{R}_+^n \setminus D_G$, $G(x)$ a plusz végtelenhez tart.

Ki fogjuk mutatni, hogy a 2. pontban leírt algoritmus olyan $x(t)$, $t = 0, 1, 2, \ldots$ pontsorozatot definiál, amelynek bármely x^* torlódási pontja a (3.2) feladat optimális megoldása. A feladat speciális voltából adódóan a Kuhn-Tucker feltételek ([1], 107. p.) most a következő alakot öltik:

$$\frac{\partial G(x)}{\partial x_j} = \sum_{k=1}^{m} a_{kj} - \sum_{k=1}^{m} b_k \frac{a_{kj}}{y_k} = 1 - \sum_{k=1}^{m} b_k \frac{a_{kj}}{y_k} \geq 0\,, \qquad j = 1, \ldots, n\,, \tag{3.3}$$

$$\sum_{j=1}^{n} x_j \frac{\partial G(x)}{\partial x_j} = \sum_{j=1}^{n} x_j \Big(1 - \sum_{k=1}^{m} b_k \frac{a_{kj}}{y_k}\Big) = 0\,, \tag{3.4}$$

ahol y_k értékét a (3.1) összefüggés határozza meg, $k = 1, 2, \ldots, m$ esetén. x^* optimalitásából következni fog annak egyértelműsége is, abban az értelemben, hogy az $x(t)$ sorozatnak csak egy torlódási pontja van, tehát konvergens. A Kuhn-Tucker feltételeket célszerű az alábbi alakba átírni, ahol $r_1, r_2, \ldots, r_m$, $u_1, u_2, \ldots, u_n$ értékét a (3.3)–(3.4)-at kielégítő x-re vonatkozóan (2.1)–(2.2) alapján határozzuk meg:

$$\frac{\partial G(x)}{\partial x_j} = 1 - \sum_{k=1}^{m} r_k a_{kj} = 1 - u_j \geq 0 \,, \tag{3.3'}$$

$$\sum_{j=1}^{n} x_j \frac{\partial G(x)}{\partial x_j} = \sum_{j=1}^{n} x_j \Big(1 - \sum_{k=1}^{m} r_k a_{kj}\Big) = \sum_{j=1}^{n} x_j (1 - u_j) = 0 \,. \tag{3.4'}$$

3.1 Tétel. *Legyen A $m \times n$-es, nemnegatív elemekből álló mátrix, x^0 és b pozitív komponensekből álló, n-, illetve m-dimenziós vektor. Ha A minden sora különbözik nullától, akkor a (2.1)–(2.3) algoritmussal meghatározott $x(t)$ sorozat ($t = 0, 1, 2, \ldots$; $x(0) = x^0$) a (3.2) konvex programozási feladat egy optimális megoldásához tart.*

Bizonyítás. Az általánosság korlátozása nélkül feltehetjük, hogy (2.4) fennáll, azaz A oszlopösszegei egységnyiek. Ekkor (2.5) mindkét oldalát j-re összegezve ($j = 1, \ldots, n$)

$$x_1(t) + x_2(t) + \ldots + x_n(t) = b_1 + b_2 + \ldots + b_m \tag{3.5}$$

adódik minden nullánál nagyobb t-re. Legyen x és x' két egymás utáni pont az algoritmus által meghatározott sorozatban; kimutatjuk, hogy $G(x) > G(x')$. Legyen $0 \leq \lambda \leq 1$, és tekintsük a $\phi(\lambda) = G\big((1-\lambda)x + \lambda x'\big)$ függvényt. (3.5) miatt $G(x)$ és ezzel együtt $\phi(\lambda)$ lineáris része nullával egyenlő, és ezért

$$\phi(\lambda) = \sum_{i=1}^{m} b_i \log b_i - \sum_{i=1}^{m} b_i \log \sum_{j=1}^{n} a_{ij} x_j (\lambda u_j + 1 - \lambda)$$

és

$$\phi'(\lambda) = \frac{\mathrm{d}\phi(\lambda)}{\mathrm{d}\lambda} = -\sum_{i=1}^{m} b_i \frac{\sum_{j=1}^{n} a_{ij} x_j (u_j - 1)}{\sum_{j=1}^{n} a_{ij} x_j (\lambda u_j + 1 - \lambda)} \,;$$

itt (2.3)-nak megfelelően x' komponenseit $x_j u_j$-vel helyettesítettük, $j = 1, 2, \ldots, n$. (2.1)–(2.2) valamint (3.5) felhasználásával az utóbbi összefüggésből a következőkhöz jutunk:

$$\phi'(0) = -\sum_{j=1}^{n} x_j u_j^2 + \sum_{j=1}^{n} x_j \,, \qquad \phi'(1) = -\sum_{j=1}^{n} x_j + \sum_{i=1}^{m} b_i \frac{\sum_{j=1}^{n} a_{ij} x_j}{\sum_{j=1}^{n} a_{ij} x_j u_j} \,;$$

kimutatjuk, hogy $\phi'(0) < 0$ és $\phi'(1) \leq 0$. Mivel defníció szerint $x_j = x_j(t)$ pozitív minden (véges) t-re,

$$0 \leq \sum_{j=1}^{n} x_j(u_j - 1)^2 = \sum_{j=1}^{n} x_j u_j^2 - \sum_{j=1}^{n} x_j \ , \tag{3.6}$$

ahol az egyenlőség a (3.5) összefüggés következménye. Ha a "$\leq$" jelnél is az egyenlőség lenne érvényes, akkor szükségképpen $u_j = 1$ teljesülne minden j-re, tehát az aktuális x pont a (3.3')–(3.4') Kuhn-Tucker feltételek értelmében optimális megoldása lenne a (3.2) feladatnak. Feltehetjük tehát, hogy most a "$<$" érvényes, ami éppen a $\phi'(0)$-ra vonatkozó állításunkat igazolja. A $\phi'(1)$ kifejezés második részösszegében $u_1, u_2, \ldots, u_n$ súlyozott átlagainak reciprokai szerepelnek, s ezért $\phi'(1)$ a $z = 1/u$ függvény konvexitásának figyelembevételével a következőképpen becsülhető:

$$\phi'(\lambda) \leq -\sum_{j=1}^{n} x_j + \sum_{i=1}^{m} \frac{b_i}{\sum_{k=1}^{n} a_{ik} x_k} \sum_{j=1}^{n} a_{ij} x_j \frac{1}{u_j} = -\sum_{j=1}^{n} x_j + \sum_{i=1}^{m} r_i \sum_{j=1}^{n} a_{ij} x_j \frac{1}{u_j}$$

$$= -\sum_{j=1}^{n} x_j + \sum_{j=1}^{n} u_j x_j \frac{1}{u_j} = 0$$

Mivel $\phi(\lambda)$ konvex, első deriváltja növekvő, s így szükségképpen nem pozitív a $0 \leq \lambda \leq 1$ intervallumon, és ennek következtében

$$\phi(1) - \phi(0) = \int_0^1 \phi'(\lambda)\, d\lambda < 0 \ .$$

A $G\big(x(0)\big)$, $G\big(x(1)\big)$, $G\big(x(2)\big)$, ... függvényértékek sorozata tehát monoton fogyó, következésképpen egy $G_{\min}$ határértékhez tart, amely $G(x) = F(Ax)$ tulajdonságai miatt nemnegatív. Itt jegyezzük meg, hogy az $x = x(t)$ sorozat (vagy annak bármely részsorozata) nem tarthat a D_G értelmezési tartomány határához, akkor ugyanis $G\big(x(t)\big)$ felülről nem lehetne korlátos. Másfelől a pozitív komponensekből álló $x = x(t)$ vektorsorozat (3.5) miatt korlátos, tehát van torlódási pontja. Ha x^* egy ilyen torlódási pont, akkor arra egyrészt $G(x^*) = G_{\min}$;, másrészt a (3.6) egyenlőtlenség egyenlőség formájában teljesül:

$$\sum_{j=1}^{n} x_j^*(u_j - 1)^2 = 0 \ , \tag{3.6'}$$

egyébként ugyanis x^* -ból kiindulva, az algoritmus szerint egy a $G_{\min}$-nél kisebb függvényértékhez lehetne eljutni, ellentmondásban a $G\big(x(t)\big)$ sorozatról mondottakkal. (3.6') szerint tetszőleges j-re $(1 \leq j \leq n)$ az

$$x_j^* = 0 \qquad \text{és az} \qquad u_j = 1$$

egyenlőségek közül legalább az egyik teljesül. A (3.3')–(3.4') Kuhn-Tucker feltételek teljesüléséhez elegendő azt kimutatnunk, hogy $x_j^* = 0$ esetén szükségképpen $u_j \leq 1$.

Ha $x_j^* > 0$ minden j-re, akkor (3.6') miatt $u_1 = u_2 = \ldots = u_n = 1$, és (3.3')–(3.4') teljesülése triviális. Tegyük fel, hogy $x_j^* = 0$ valamilyen j indexre, és bontsuk fel a nemnegatív egész számok $S = \{0, 1, 2, \ldots\}$ sorozatát két részsorozatra a következőképpen:

$$S_1 = \{\, t \in S \mid u_j(t) \geq 1 \,\}\,, \qquad S_2 = \{\, t \in S \mid u_j(t) < 1 \,\}\,.$$

S_2 nem lehet véges, mert különben $x_j(t+1) = x_j(t)u_j(t)$ miatt $x_j(t)$-nek nem lenne $x_j^* = 0$-hoz tartó (rész)sorozata. Ha S_1, véges, akkor nyilván

$$\lim_{t\to\infty} x_j(t) = 0 \qquad \text{és} \qquad \limsup_{t\to\infty} u_j(t) \leq 1\,.$$

Ha viszont S_1 is végtelen, akkor könnyen belátható, hogy minden olyan $S' = \{\, t_1, t_2, \ldots \}$ részsorozathoz, amelyre

$$\lim_{k\to\infty} x_j(t_k) = 0\,,$$

található olyan $S'' = \{\, t_1', t_2', \ldots \} \subset S_2$ részsorozat, hogy $k = 1, 2, \ldots$ esetén $t_k' < t_k$ és $x_j(t_k') \leq x_j(t_k)$, tehát

$$\lim_{k\to\infty} x_j(t_k') = 0 \qquad \text{és} \qquad \limsup_{k\to\infty} u_j(t_k') \leq 1\,.$$

Ezzel kimutattuk, hogy $x(t)$ bármely torlódási pontja teljesíti a Kuhn-Tucker feltételeket, tehát optimális megoldása a (3.2) feladatnak. Konvex programozási feladatról lévén szó, tetszőleges két optimális megoldás, x^* és x'' esetén $\frac{1}{2}(x^* + x'')$ is optimális megoldás. Az $F(y)$ függvény szigorú konvexitása miatt

$$F\big(\frac{1}{2}(y^* + y'')\big) = F\big(\frac{1}{2}(Ax^* + Ax'')\big) < \frac{1}{2}\big(F(y^*) + F(y'')\big)\,,$$

hacsak $Ax^* \neq Ax''$; x^* és x'' optimalitása miatt ebből szükségszerűen az következik, hogy $Ax^* = Ax''$. Eszerint $y(t) = Ax(t)$ konvergens, és hasonlóképpen konvergálnak az $r_i(t) = b_i/y_i(t)$ sorozatok is, $i = 1, 2, \ldots, m$ esetén. Ebből következik az $u_j(t) = \sum_{k=1}^{m} a_{kj} r_k(t)$ valamint az $x_j(t+1) = x_j(t)u_j(t)$ sorozat konvergenciája, és ezzel állításunkat bizonyítottuk.

3.2 Korollárium. *Ha az (1.1) egyenletnek van nemnegatív x megoldása, akkor a (2.1)–(2.3) algoritmus egy ilyen megoldást eredményez, feltéve, hogy az x^0 induló vektor minden komponense pozitív; x^0 egyébként tetszőleges lehet.*

A korollárium az előző tétel egyszerű következménye.

4. Alkalmazások

A módszer eddigi alkalmazásai a KSH lakossági felvételeiből származó problémákra korlátozódtak, tehát bizonyos pozitív $w_1^0, w_2^0, \ldots, w_J^0$ súlyok, mint induló értékek birtokában a

$$\sum_{j=1}^{J} w_j Y_{\cdot j} = \mathbf{N} \tag{4.1}$$

egyenlet $w = (w_1, w_2, \ldots, w_J)^T$ megoldását kerestük – a jelölések értelmezését lásd a bevezetésben. A konkrét alkalmazások közül megemlítjük az 1992. évi jövedelem felvétel, valamint az 1993. évi nemzetiségi felvétel feldolgozását. Mivel a (2.1)–(2.3) algoritmus bizonyos változók (súlyok) korrigált értékeként nullát eredményezhet, a (4.1) feladatot a

$$v_j = w_j - 0.15 w_j^0$$

transzformációval a

$$\sum_{j=1}^{J} v_j Y_{\cdot j} = \mathbf{N}^1 \tag{4.2a}$$

feladatba vittük át, ahol

$$\mathbf{N}^1 = \mathbf{N} - 0.15 \sum_{j=1}^{J} w_j^0 Y_{\cdot j}\,, \qquad v \geq 0\,, \tag{4.2b}$$

és v_j^0 kezdeti értéke

$$v_j^0 = 0.85 w_j^0 \tag{4.2c}$$

volt. A "0.15" tényezőt heurisztikus alapon választottuk, éspedig azzal a céllal, hogy b^1 minden komponense pozitív legyen, továbbá, hogy (4.1')-nek legyen nemnegatív megoldása. Ez a választás azt is biztosította, hogy a korrigált $w_j = v_j + 0.15 w_j^0$ súlyok szélsőséges esetben sem lehettek kisebbek induló értékük 15%-ánál.

1. Táblázat: Számítástechnikai tapasztalatok a korrekciós eljárással kapcsolatban (Minta: nemzetiségi felvétel, 1993. szeptember-november)

Megye	Háztartások száma	Személyek száma	Iterációk száma		Gépidő (mp)	
			RAS	SMART	RAS	SMART
Budapest	2872	6802	19	19	4,74	5,32
Baranya	911	2512	34	33	1,85	2,09
Bács-Kiskun	1416	3674	35	35	2,86	3,16
Békés	1002	2574	41	40	2,09	2,34
Borsod-A.-Z.	1719	4754	34	34	3,39	3,83
Csongrád	978	2408	36	36	1,91	2,16
Fejér	952	2614	31	31	1,83	2,10
Győr-M.-S.	956	2687	34	34	1,91	2,18
Hajdú-Bihar	1215	3239	30	30	2,29	2,60
Heves	888	2371	20	20	1,53	1,74
Komárom-E.	732	1988	22	23	1,29	1,49
Nógrád	599	1608	31	31	1,16	1,31
Pest	2088	5781	44	44	4,34	5,12
Somogy	785	2134	29	30	1,47	1,72
Szabolcs-Sz.-B	1299	3878	25	25	2,40	2,75
J.-N.-Szolnok	1059	2892	26	26	1,97	2,28
Tolna	689	1856	34	35	1,34	1,57
Vas	617	1722	30	30	1,20	1,37
Veszprém	914	2561	27	27	1,75	1,97
Zala	687	1942	29	30	1,32	1,53

A (4.1) feladatnak a (4.2a-c) átfogalmazása azzal a kérdéssel függ össze, amelyet a 2. fejezet végén említettünk, miszerint nem ismeretes, hogy a (2.1)-(2.3) algoritmus milyen értelemben eredményezi az induló x^0 (w^0) vektor „jó" közelítését. Nem sokat segítene, ha – mint a Darroch-Ratcliff féle SMART esetében – ki lehetne mutatni, hogy w és w^0 eltérése I-divergenciával mérve minimális, ha ettől még egy-két korrigált súly értéke zérus lenne, ami a megfelelő megfgyeléseknek a vizsgálatból való kirekesztését jelentené.

Célszerű itt még azt is megjegyezni, hogy a konkrét alkalmazások során – amelyek tapasztalatainak egy részét az 1. táblázatban mutatjuk be – módszerünk és a SMART algoritmus gyakorlatilag ugyanazt a megoldást eredményezte. Ez arra utal, hogy a kutatást ebben az irányban folytatni kellene.

A táblázat a nemzetiségi felvétel feldolgozása során szerzett tapasztalatainkat mutatja be. A korrekciós számítások FORTRAN-77 programja a KSH HP9000-867-es szerverén futott. Az egyes megyei szintű feladatok méretéről

a háztartások, illetve a személyek száma ad felvilágosítást. Budapest esetén például a feladat mátrixa 32 × 2872-es volt (a nemek-korcsoportok szerinti keresztosztályok száma ugyanis 32), és ebben a mátrixban csak 6802 elem volt nullától különböző. A program a mátrixnak csak a nullától különböző elemeit tárolta, pozíció szerint. A táblázat utolsó négy oszlopa az iterációs lépések számát, illetve a gépidő-felhasználást mutatja; itt mód nyílik eljárásunknak a Darroch-Ratcliff féle SMART algoritmussal való összehasonlítására (a RAS módszerhez való hasonlósága miatt a táblázatban eljárásunkat a "RAS" címkével azonosítottuk). Az iterációk számát az a feltétel szabta meg, hogy a (3.2) összefüggésben szereplő $G(x)$ célfüggvény értéke 1-nél kisebb legyen, ami Budapest esetén például azt jelenti, hogy a város magánháztartásokban élő népességének 1 982 708 létszámát (továbbvezetett, becsült adat, 1993. január 1.) egységnyi pontossággal sikerült megközelíteni.

Mint látható, az egyes feladatokban eljárásunk és a SMART eljárás gyakorlatilag ugyanannyi iteráció után ért célhoz, az elhasznált gépidő azonban, amelynek értékét a libU77 könyvtár DTIME függvényeljárásának segítségével határoztuk meg, a SMART módszernél mindig nagyobb, mint a dolgozatban közölt eljárás alkalmazásánál. Ennek nyilvánvalóan az az oka, hogy eljárásunk a változók értékét az egyes iterációkban az egyenletek hibájának (a bal és jobb oldal eltérésének) súlyozott számtani átlaga segítségével javítja, míg a SMART algoritmus ugyanezeknek a mennyiségeknek a mértani átlagával operál.

Az 1. táblázat megfelelőjét összeállítottuk a munkaerőfelmérés 1994. júliusi adatai alapján is, mely esetben a mintanagyság kb. egyharmada a nemzetiségi felvételben megfigyelt minta nagyságának. Gyakorlatilag ugyanaz a tendencia érvényesült ott is, mint az 1. táblázatban, a felhasznált gépidő mindenütt közel egyharmada volt annak, amire a háromszor akkora minta esetén volt szükség. Ami meglepő volt, az az iterációk számának oszlopában mutatkozott, ott ugyanis semmiféle szabályszerűséget sem lehetett felfedezni. Egy-egy példa (nemzetiségi felvétel – munkaerőfelmérés sorrendben): Budapest 19-24, Bács-Kiskun 35-18, Borsod 34-30, Komárom 22-25, Nógrád 31-47, Veszprém 27-40.

Irodalom

1. Collatz, L. – W. Wetterling: *Optimierungsaufgaben.* Springer Verlag, Berlin / Heidelberg / New York 1971.

2. Csiszár, I.: Entrópiamaximalizálás és rokon módszerek: axiomatika, algoritmusok. *Szigma* XXIV(1993), 111–137.

ADJUSTMENT FOR NON-RESPONSE IN HOUSEHOLD SURVEYS: A SPECIAL LINEAR INVERSE PROBLEM

Data from household surveys are often adjusted for non-response by updated census counts. In such cases adjustment means re-weighting the observations in such a way that estimated totals for the adjustment cells defined as cross-classes by age, sex and geographical domains comply with the corresponding updated census counts. If in any sample household each member must have the same (adjusted) sample weight, the problem becomes a special linear inverse problem, i.e. we are given a system of linear equations with an initial unsatisfactory solution, and have to find an acceptable solution to this system, which is close to the initial solution in some sense. The initial and the final solutions consist of the original and the adjusted sample weights, respectively, and all quantities occurring in the problem are non-negative. In the paper a new method for solving the special linear inverse problem is given, and its application in the household surveys of the Hungarian Central Statistic Office is discussed.