O DOPOLNJEVANJU IZGUBLJENIH VREDNOSTI V PODATKOVNI MATRIKI S POMOČJO TESTOV (S KOMBINATORNO LOGIČNIMI SREDSTVI)

Valerij Kudrjavcev, Moskovska državna univerza Lomonosova, Moskva
Žiga Knap, Institut za sociologijo pri Univerzi v Ljubljani

Avtorja razvijata postopek dopolnjevanja manjkajočih vrednosti podatkovnih matrik na osn testnega pristopa (kombinatorno logične metode prepoznavanja vzorcev). Obravnavata pet tip problemov in predlagata deterministično in stohastično varianto reševanja zastavljene naloge.

Ključne besede: podatkovne matrike, izgubljene vrednosti, testi, prepoznavanje vzorcev

COMPLETION OF MATRIX DATA BASIS BY TEST APPROACH (COMBINATORIA LOGICAL APPROACH). The autors of this article developed an approach to possible completio of matrix data bases, where the caracteristics of objects are either columns or rows or both. Five typ of such completion are being analysed and the authors suggest two kind of solution - deterministic well as stochastic variations.

Key words: data matrix, missing value, tests, pattern recognition

0. Uvod

V prispevku proučujemo vprašanje o nadomeščanju izgubljenih vrednosti v pravokotn podatkovnih matrikah. Predlagamo postopek (algoritem) nadomeščanja izgubljenih vrednosti, je osnovan na uporabi metod prepoznavanja vzorcev. Študiramo nekaj logično možnih situacij izhajajoč iz predpostavke, da so konstitutivni elementi matrike (entitete, opisane z matriko) stolpci, vrstice, ali oboje hkrati. Predlagani pristop je učinkovit, ko je znano, da posamezni stolpec ali vrstica opisuje lastnosti, in pa, ko o tem ni ničesar znanega.

1. Formulacija problema

Predpostavljamo, da imamo dani dve množici (abecedi)

$$\mathbf{A}=(a_1, a_2, ..., a_n) \quad \text{in} \quad \mathbf{B}=(*).$$

Študiramo matriko $\mathbf{T}=(d_{ij})$ reda $s \times t$, kjer nam s pomeni število stolpcev, t pa število vrstic. V matriki nastopajo elementi, ki so iz unije množic **A** in **B**, pri tem pomeni $d_{ij}=*$, da je ta element nedoločen (izgubljena vrednost).

Preglejmo vse možnosti, ki lahko nastopijo:

(1) V matriki **T** predstavljajo stolpci parametre-lastnosti, vrstice pa so enote (entitete).
(2) V matriki **T** predstavljajo vrstice parametre-lastnost, stolpci pa so enote (entitete).
(3) V matriki **T** so tako vrstice kot stolpci parametri lastnosti.
(4) Za matriko **T** ni znano ali so stolpci ali vrstice parametri-lastnosti.
(5) En del stopcev in en del vrstic predstavlja parametre-lastnosti.

Zanimalo nas bo dopolnjevanje nepopolne matrike v situacijah, ki so opisane v (1) do (5). Da bomo izražanje poenostavili, bomo nalogo, ki se nanaša na možnost oštevilčeno z (i), na kratko poimenovali problem-(i) rekonstrukcije matrike za i=1, 2, 3, 4, 5.

Takoj vidimo, da sta problema-(i) za i=1 in 2 med sabo ekvivalentna, medtem ko problem-(5) vključuje vse probleme-(i) za i=1, 2, 3. Podrobno bomo obdelali problem-(1) in problem-(3), potem pa bomo še pokazali, kakšno strategijo reševanja problema-(i) vidimo pri i=3 in 5.

2. Rešitev problema-(1)

V matriki **T** poiščemo povsod določeno podmatriko **T'** (torej tako, ki ne vsebuje nobenega elementa *), ki jo sestavlja nekaj zapored izbranih vrstic in nekaj zapored izbranih stolpcev. Tako podmatriko imenujmo blok. Blok **T'** imenujemo maksimalni blok, če v matriki **T** ne eksistira blok $\mathbf{T^0}$, ki bi vseboval **T'** in bi bil različen od **T'**.

Množico vseh maksimalnih blokov matrike **T** zaznamujmo z **B(T)**. Izberimo poljuben maksimalen blok iz **B(T)**. Da bomo imeli opravka z enostavnejšimi označbami, predpostavimo, da je izbrani blok v zgornjem levem kotu matrike **T** in naj je njegov red $s' x t'$, kjer je s' število vrstic in t' število stolpcev. Oglejmo si poljubno, ne povsod opredeljeno vrstico v, ki naj bo oštevilčena z $l > t'$, iz podmatrike $\mathbf{T_1}$, ki je sestavljena iz prvih s' stolpcev matrike **T**. Pokazali bomo, kako dopolnimo (rekonstruiramo) tiste elemente te vrstice, ki so označeni z "*".

Konstruirajmo v matriki **T'** podmatriko **T"**, ki jo tvorijo tisti stolpci, v katerih ima naša vrstica v elemente iz abecede **A**. Pripomnimo, da je lahko ta podmatrika tudi prazna matrika, v tem slučaju se odpovemo rekonstrukciji manjkajočih elementov v tem delu vrstice v.

Če matrika **T"** ni prazna, tedaj na osnovi matrike **T"** konstruiramo matriko **T'"**, ki sestoji iz vseh paroma različnih vrstic matrike **T"** (iz matrike **T"** izločimo tiste vrstice, ki več kot enkrat nastopajo, in pustimo samo po enega predstavnika takih vrstic). Vsaka vrstica matrike **T"** torej sovpada z vrstico matrike **T'"**.

Po matriki **T'"** izračunamo množico vseh njenih testov, ki jo označimo s **F(T'")**. Povejmo, da imenujemo test matrike **T'"** vsak nabor (številke) stolpcev, ki imajo to lastnost, da izberejo iz **T'"** tako podmatriko, ki ima vse vrstice paroma različne.

Opišimo v nekaj korakih postopek *C*, ki najprej določa 'podobnost' vrstice v s kakšno vrstico iz matrike **T'"**, potem pa z upoštevanjem večkratnega nastopanja vrstice iz **T'"** v matriki **T"** določa *'podobnost'* vrstice v z vrsticami matrike **T"** in **T'**, kar nam omogoča dopolniti vrstico v.

1.-korak.

Vzemimo test S iz množice **F(T''')** in primerjajmo podvrstice iz **T'''**, ki jih izreže test S iz matrike **T'''**, z ustreznim delom podvrstice v, ki jo dobimo prav tako s pomočjo testa S (v obeh primerih imamo opravka s tistimi elementi vrstic, ki imajo isti indeks stolpca, kot so stolpični indeksi v testu S).

Nastopita lahko dve situaciji:

. ali podvrstica v sovpada z eno od podvrstic (denimo s tisto oštevilčeno z i) matrike **T'''**,
. ali pa je različna od vseh takšnih podvrstic.

V prvem slučaju štejemo, da je test S kot "ekspert" ugotovil (glasoval za) podobnost vrstice v z i-to vrstico matrike **T'''**, v drugem slučaju pa ugotavljamo, da "ekspert" S ni sprejel nobene odločitve (se je vzdržal, odpovedal glasovanju).

Na ta način postopamo z vsakim testom S iz množice **F(T''')**. Kot rezultat tega postopka dobimo vektor $\mathbf{P}=(p_1, p_2, ..., p_{t'''}, q)$, kjer s p_j označujemo število "glasov", ki so bili zbrani za podobnost vrstice v z j-to vrstico matrike **T'''**, s številom q pa označujemo skupno število situacij, ko ni bila sprejeta nobena odločitev za podobnost; števila so še normirana s številom vseh testov, torej deljena z močjo množice **F(T''')**; s t''' smo označili število vseh vrstic v matriki **T'''**.

Opozorimo, da je lahko v matriki **T''** več vrstic, kot pa jih je v matriki **T'''**. V tem primeru pripišemo vsem vrsticam v matriki **T''**, ki sovpadajo z j-to vrstico matrike **T'''**, kar iste normirane "*glasove*" iz **P** kot za ustrezne vrstice iz matrike **T'''**, le da so še pomnoženi z večkratnostjo dane vrstice v matriki **T''**, in jih seveda delimo s celotnim številom vrstic v matriki **T''**. To količino imenujmo utež vrstice v matriki **T''**.

Tako postopamo za vsak j. Prav te uteži pripišemo tudi vrsticam v matriki **T'**, iz katerih smo dobili vrstice iz **T''**. Te uteži potem določajo "*podobnost*" vrstice v z vrstico iz **T'**. In tak postopek izvršimo za vsak j.

2.-korak.

Naš postopek C pripiše vsaki nedoločeni vrednosti v vrstici v (torej vsakemu simbolu "*") neko virtuelno vrednost na tale način. Vzamemo stolpec v matriki **T'** in to takšen, da se bo v nadaljevanju tega stolpca nahajal v vrstici v znak "*". Oglejmo si vse vrednosti $a_{i1}, a_{i2}, ..., a_{ik}$, ki jih nahajamo v tem stolpcu; seveda se lahko nekatere med njimi ponavljajo.

Vsaki od teh k vrednosti a_{ir} pripišemo določeno utež na naslednji način. Seštejemo vse tiste uteži, ki so pripisane vrsticam matrike **T'**, ki imajo to vrednost (črka iz abecede **A**), pa označimo to število s h_{ir}. Zapišimo to dejstvo z $a_{ir}(h_{ir})$.

Če zberemo vse tako dobljene izraze, dobimo niz ($a_{i1}(h_{i1})$, $a_{i2}(h_{i2})$, ..., $a_{ik}(h_{ik})$, o(q)), v tem nizu o(q) označuje skupno število *q* primerov, ko odločitev o "podobnosti" ni bila sprejeta.

3.-korak.

Zdaj postopek *C* apliciramo k naslednji vrstici *v'*, ki je različna od *v* in ni iz matrike **T'**, hkrati pa leži v pasu, ki ga tvorijo stolpci oštevilčeni od *1* do *s*. Tako postopamo potem z vsemi vrsticami *v'*.

4.-korak.

Ko je izvršen tretji korak, izberemo nov maksimalni blok $\mathbf{T_1'}$. Zdaj ne upoštevamo rezultatov dopolnjevanja simbolov *, ki smo jih dobili pri 3.-koraku pri aplikaciji k bloku **T'**, in uporabimo postopek A_1 zaporedoma od 1. do 3.-koraka, vendar zdaj apliciran na blok $\mathbf{T_1'}$. Ta proces nadaljujemo potem za vse maksimalne bloke.

5.-korak.

Ko je izvršen 4.-korak, je lahko vsaki vrednosti pripisan svoj lasten kod dopolnitev, ki ga generira konkreten maksimalni blok. Lahko privzamemo, da imamo kodo vedno zapisano v obliki ($a_1(h_1)$, $a_2(h_2)$, ..., $a_n(h_n)$, o(h)), če le privzamemo, da označba $a_i(0)$ pomeni, da v tem primeru element $a_i(h_i)$ ne nastopa.

Zdaj pa vzamemo povprečno vrednost koda za tisto vrednost, ki naj izraža končni izračun virtuelnih možnosti za simbol * na konkretnem mestu v matriki **T**. To povprečno vrednost računamo takole: za vsak element a_i se seštejejo njegove uteži iz vseh kodov, ki ustrezajo upoštevanim maksimalnim blokom, in nato vsoto še delimo s številom vseh takih blokov; analogno izračunamo srednjo vrednost za o(q).

Na ta način dobljeno matriko dopolnitev označimo s $\mathbf{T^*}$. To matriko proglasimo za rešitev problema-(1).

Pripomba 1. Opisani postopek za konstrukcijo matrike $\mathbf{T^*}$ je zelo kompleksen, čemur se lahko izognemo s približno rešitvijo problema-(1). To dosežemo na naslednji način:

> *Prvič*, ne upoštevamo vseh maksimalnih blokov, ampak nek sistem maksimalnih blokov, ki "*pokriva*" vse elemente "*" matrike **T**, ki jih je potrebno dopolniti, in ni abundanten v zgoraj opisanem smislu.
>
> *Drugič*, namesto determiniranih postopkov iskanja testov in uteži lahko uporabimo razne vrste stohastičnih postopkov.

Pripomba 2. Če imamo fiksne prage verjetnosti vrednosti nedoločenosti v matriki **T**, ki jih sporoči naročnik glede na matriko $\mathbf{T}^*$, lahko potem določamo meje verjetnosti, da se taka matrika pojavi.

3. Rešitev problema-(3), -(4), -(5)

Oglejmo si najprej problem-(3). V tem primeru gledamo na nalogo, kot da rešujemo problem-(1) in najdemo vse možne dopolnitve elementov matrike **T**, ki jim je pripisan znak *. Nato rešujemo problem-(1) za matriko **T**, v kateri smo zamenjali vrstice s stolpci, in prav tako najdemo vse možne dopolnitve za elemente "*".

Kot rezultat dobimo pripisan vsakemu elementu "*" matrike **T** najprej en kod dopolnitev, potem pa še drug. Vzamemo srednjo vrednost, ki jo dobimo iz obeh kodov, kot smo to že prej počeli, in dobimo matriko $\mathbf{T}^{**}$. Ta matrika predstavlja rešitev problema-(3). Tudi problem-(3) dopušča približno rešitev na račun dvakratnega približnega reševanja problema-(1) v postopku reševanja problema-(3).

Oglejmo si problem-(4). Pri tej nalogi imamo opravka s tremi možnostmi:

prvič, lahko so stolpci lastnosti, vrstice pa to niso,
drugič, vrstice so lastnosti, stolpci pa to niso in
tretjič, tako stolpci kot vrstice so lastnosti.

V prvem primeru rešujemo problem-(1), v drugem primeru rešujemo problem-(2), dualen k problemu-(1), in v tretjem primeru rešujemo problem-(3). Te tri rešitve predstavljajo tudi rešitve problema-(4). Njegovo približno rešitev dobimo kot rezultat približnih rešitev problemov-(i) za i=1, 2, 3. Ko naročnik precizira svojo postavitev problema, lahko izbere kakšno izmed teh treh rešitev ali pa njihovo kombinacijo.

Oglejmo si problem-(5) in začnimo s situacijo, ko del stolpcev predstavlja lastnosti, medtem ko vrstice niso lastnosti.

V tem primeru iz matrike **T** izrežemo podmatriko, ki jo tvorijo tisti stolpci, ki predstavljajo lastnosti, in za to podmatriko rešujemo problem-(1), ki rezultira v opisu možnih dopolnitev elementov "*". Nato pa obravnavamo matriko **T**, kjer štejemo vrstice za stolpce, izrežemo iz nje podmatriko, ki jo tvorijo stolpci-lastnosti in zanjo rešujemo problem-(1), ki se zaključi z opisom dopolnjevanj elementov "*".

Iz dobljenih dveh kodov uteži izražunamo za vsak element "*" srednjo vrednost in dobimo kot odgovor virtuelno matriko, v kateri vsi elementi "*" v splošnem niso dopolnjeni. Tudi tu so možne približne rešitve nalog, ki jih dobimo kot rezultat uporabe približnih rešitev pri reševanju nastopajočih problemov-(i) za i=1, 2, 3.

Našteti postopki dopuščajo algoritmično realizacijo, v kateri se uporabljajo bloki podprogramov kombinatornega, logičnega in stohastičnega tipa. Ti bloki so do določene

mere že izdelani in z določeno modifikacijo jih lahko preberemo v Jablonskij (1958), Konstantinov, Koroleva, Kudrjavcev (1976) in Nejman (1968).

Naš pristop dopušča bistveno posplošitev za slučaj, ko je na vrsticah in stolpcih matrike T vnaprej postulirana določena ekvivalentnost (posebej za vrstice, posebej za stolpce). V tem primeru se naš problem tesno prilega k problemom prepoznavanja vzorcev, kar pa je že predmet posebne obravnave.

LITERATURA

Jablonskij, S. V. (1958). *Funkcional'nye postroenija v k-značnoj logike*. Trudy MIAN SSSR im. Steklova, tom LI

Konstantinov, R. P., Koroleva, Z. E., Kudrjavcev, V. B. (1976). "*O kombinatorno-logičeskom podhode k zadačam prognoza rudonosnosti*". sb. Problemy kibernetiki, vypusk 31, str. 5-33

Nejman, Ju. (1968). *Vvodnyj kurs teorii verojatnostej i matematičeskoj statistiki*. Moskva: "Nauka".