PŘESNÝ TEST NEZÁVISLOSTI V KONTINGENČNÍ TABULCE R x C

Miroslav Hartmann, Josef Bukač

Výpočetní středisko Lékařské fakulty UK
v Hradci Králové, Šimkova 870

1. ÚVOD

Tento postup byl vyvinut pro případy, kdy χ^2-test dobré shody, používaný při testování nezávislosti v kontingenčních tabulkách, nevyhovuje pro velmi malé pozorované četnosti v některých políčkách. Metodu přesného výpočtu hladiny významnosti vyvinuli Freeman a Halton [1]. V této práci z roku 1951 se ještě nepředpokládá použití počítače a není ani zaměřena na jednoznačné stanovení algoritmu výpočtu. První program s přesným popisem postupu výpočtu publikoval March [2]. Při podrobném rozboru se ukazuje, že postup nevyužívá speciálních vztahů v kontingenčních tabulkách, které lze odvodit elementárními úvahami. Program je velmi jednoduchý a krátký, ale ne dostatečně účinný pro praktické použití. Některých speciálních vztahů pro tabulky typu 2xC využívají Stucky a Vollmar [3]. Jimi sestrojený program pro tento typ tabulek je podstatně efektivnější, než Marchův.

Naše práce je zaměřena na obecné tabulky typu RxC. Vycházíme z Marchovy myšlenky, postup jsme podstatně upravili a tím urychlili několikanásobně vlastní výpočet.

2. STANOVENÍ PROBLÉMU

Kontingenční tabulkou typu RxC rozumíme matici celých nezáporných čísel

$$\mathbb{X} = \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,C} \\ \vdots & & & \\ x_{R,1} & x_{R,2} & \cdots & x_{R,C} \end{pmatrix} \tag{1}$$

Marginální součet i-tého řádku tabulky je definován vztahem

$$r_i = \sum_{j=1}^{C} x_{i,j} , \tag{2}$$

obdobně marginální součet j-tého sloupce je definován vztahem

$$c_j = \sum_{i=1}^{R} x_{i,j} . \tag{3}$$

Celkový počet pozorování v tabulce je dán výrazem

$$N = \sum_{i=1}^{R} \sum_{j=1}^{C} x_{i,j} = \sum_{i=1}^{R} r_i = \sum_{j=1}^{C} c_j . \tag{4}$$

Pravděpodobnost výskytu tabulky $\mathbb{X}$ pro předem dané marginální součty r_i a c_j je při platnosti hypotézy nezávislosti dána výrazem

$$P(\mathbb{X}) = \frac{\prod_{i=1}^{R} (r_i!) \prod_{j=1}^{C} (c_j!)}{N! \prod_{i=1}^{R} \prod_{j=1}^{C} (x_{i,j}!)} . \tag{5}$$

Hladinu významnosti $P_s(\mathbb{X}_0)$ testu nezávislosti v dané tabulce definujeme součtem

$$P_s(\mathbb{X}_0) = \sum_{P(\mathbb{X}) \leq P(\mathbb{X}_0)} P(\mathbb{X}) . \tag{6}$$

Tuto definici vyjádříme slovně takto:

a. zadaná tabulka $\mathbb{X}_0$ určuje konkrétní hodnoty marginálních součtů r_i a c_j a hodnotu N.

b. stanovíme pravděpodobnost $P(\mathbb{X}_0)$ zadané tabulky.

c. vytvoříme všechny možné tabulky $\mathbb{X}$, které vyhovují daným marginálním součtům a pro každou z nich vypočteme její pravděpodobnost $P(\mathbb{X})$.

d. hladina významnosti je pak dána jako součet všech pravděpodobností $P(\mathbb{X})$, pro které platí $P(\mathbb{X}) \leq P(\mathbb{X}_o)$.

Vzhledem k tomu, že hodnoty r_i, c_j, N jsou ve všech vytvářených tabulkách totožné, je vhodné přepsat vzorec (5) do tvaru

$$P(\mathbb{X}) = Q(\mathbb{X}_o)/R(\mathbb{X}), \qquad (7)$$

kde

$$Q(\mathbb{X}_o) = \prod_{i=1}^{R}(r_i!) \prod_{j=1}^{C}(c_j!) \;/\; (N!) \qquad (8)$$

a

$$R(\mathbb{X}) = \prod_{i=1}^{R} \prod_{j=1}^{C} (x_{i,j}!) \quad . \qquad (9)$$

3. GENEROVÁNÍ TABULEK

3.1 Marchova metoda

Jelikož jednotlivé hodnoty $x_{i,j}$ musí vyhovovat rovnicím (2) a (3), je tabulka typu RxC dána (R-1)x(C-1) prvky, přičemž zbývající prvky jsou z těchto rovnic dopočítány. Přípustnou tabulkou nazveme takovou tabulku, která splňuje rovnice (2), (3) a jejíž všechny prvky jsou nezáporné. První tabulku vytváří March tak, že pro (R-1)x(C-1) prvků vpravo dole, tj. s vynecháním prvního sloupce a prvního řádku, dosadí minimum z příslušných sloupcových a řádkových marginálních součtů. Po vypočtení prvků v prvním sloupci a řádku podle rovnic (2) a (3) zjistí, zda tabulka je přípustná. Generování další tabulky vychází vždy z vygenerované tabulky předchozí. V této tabulce hledá postupně v prvcích $x_{2,2},\ldots,x_{2,C}, x_{3,2},\ldots,x_{3,C},\ldots, x_{R,C}$ první kladný prvek. Od tohoto prvního odečte jedničku a do všech předcházejících, které byly nulové, dosadí minimum z příslušných marginálních součtů. Opět vypočítá prvky v prvním sloupci a řádku a zjistí, zda vzniklá tabulka je přípustná. Následuje opakovaný návrat do vyhledávání kladného prvku. Celý postup je ukončen, jestliže není nalezen žádný další prvek.

V podstatě tento postup představuje (R-1)x(C-1) násobný cykl s pevnými mezemi a ukazuje se, že obecně je při něm generován velký počet nepřípustných tabulek.

3.2 Nová metoda

Jde o způsob, podobný metodě severozápadního rohu, používané pro nalezení základního řešení lineárního dopravního problému. Z hlediska dopravního problému lze hodnoty marginálních součtů r_i, resp. c_j zadané tabulky $\mathbb{X}_o$ považovat za kapacity řádků, resp. sloupců tabulky. Toho využijeme následujícím způsobem:

Před započetím generování naplníme první řádek tabulky hodnotami c_j z tabulky $\mathbb{X}_o$, tj. na prvním řádku jsou umístěny kapacity sloupců ($x_{1,j} = c_j$). Obdobně do prvého sloupce, počínaje druhým řádkem, umístíme hodnoty r_i z tabulky $\mathbb{X}_o$, tj. kapacity řádků ($x_{i,1} = r_i$ pro $i=2,3,\ldots,R$). Ostatní prvky tabulky nechť jsou nulové. Takto vzniklá tabulka splňuje rovnice (2) pro všechny řádky s výjimkou prvního a rovnice (3) pro všechny sloupce s výjimkou prvního.

Při generování jakéhokoliv prvku $x_{i,j}$ ($i > 1$, $j > 1$) v tabulce dbáme vždy této zásady: Jestliže prvek zvětšíme, resp. zmenšíme o určitou hodnotu, pak o tutéž hodnotu zmenšíme, resp. zvětšíme i prvky $x_{i,1}$ a $x_{1,j}$. Tím je tedy neustále zachována platnost rovnic (2) a (3) kromě prvního řádku a prvního sloupce.

První tabulku generujeme tak, že do $x_{R,C}$ dosadíme minimum z $x_{R,1}$ a $x_{1,C}$. Po dosazení okamžitě hodnoty $x_{R,1}$ a $x_{1,C}$ upravíme. Obdobně do prvku $x_{R,j}$ ($j=C-1, C-2,\ldots,2$) dosadíme minimum z hodnot $x_{R,1}$ a $x_{1,j}$. Po tomto procesu již nebudeme považovat hodnotu $x_{R,1}$ za volnou kapacitu R-tého řádku, ale za hodnotu, která byla v tomto prvku vygenerována. O tuto hodnotu $x_{R,1}$ snížíme prvek $x_{1,1}$ /tj. kapacitu prvního sloupce/. Stejně jako v R-tém řádku postupujeme i v řádcích vyšších /tj. $i=R-1,R-2,\dots,2$/. Po vygenerování celého druhého řádku budeme také hodnoty, které zůstaly v prvním řádku, považovat za vygenerované. Tato první vygenerovaná tabulka je vždy přípustná /analogie metody severozápadního rohu u dopravního problému/ a navíc je totožná s první přípustnou tabulkou, generovanou Marchovou metodou.

Při generování dalších tabulek vycházíme podobně jako March vždy z předcházející vygenerované tabulky. V prvcích $x_{2,2},\dots,x_{2,C}, x_{3,2},\dots,x_{3,C},\dots,x_{R,2},\dots,x_{R,C}$ hledáme první kladný prvek. Navíc před započetím prohledávání i-tého řádku přičteme hodnotu $x_{i,1}$ k hodnotě $x_{1,1}$. Od prvního kladného prvku, řekněme $x_{i,j}$, odečteme jedničku a navíc tuto jedničku přičteme k prvkům $x_{i,1}$ a $x_{1,j}$. V pořadí opačném k pořadí prohledávání znovu generujeme již popsaným způsobem /jako u první tabulky/ hodnoty prvků, předcházejících prvku $x_{i,j}$.

Ze způsobu generování plyne, že přípustnost vygenerované tabulky je závislá pouze na hodnotě prvku $x_{1,1}$. Tabulka je přípustná právě tehdy, když $x_{1,1} \geqq 0$.

Vzhledem k tomu, že hodnota $x_{1,1}$ během generování tabulky představuje volnou kapacitu prvního sloupce, potom jestliže po vygenerování i-tého řádku $(i \geqq 2)$ je hodnota $x_{i,1}$ větší než hodnota $x_{1,1}$ a tedy po odečtení by hodnota $x_{1,1}$ byla záporná, pak ať je ve vyšších řádcích tímto postupem generováno cokoliv, výsledná tabulka nebude nikdy přípustná. V takovém případě není tedy třeba zbytek tabulky generovat a hodnotu $x_{i,1}$ od hodnoty $x_{1,1}$ neodečteme. Můžeme vyhledat první kladný prvek, počínaje prvkem $x_{i,2}$ a od něj odečíst jedničku. Jestliže však odečteme jedničku od prvku $x_{i,2}$ /je-li to možné/, zvýšíme tím hodnotu $x_{i,1}$ a po odečtení této hodnoty od $x_{1,1}$ /kapacity 1. sloupce/ získáme ještě menší hodnotu $x_{1,1}$ a tudíž ani tato tabulka nebude přípustná. Analogicky můžeme uvážit:

Byla-li po vygenerování i-tého řádku přečerpána kapacita 1. sloupce a byly-li vyčerpány kapacity 2. až j-tého sloupce, pak jakékoliv snížení hodnot v 2. až $(j+1)$. prvku i-tého řádku nevede k přípustné tabulce. V důsledku toho můžeme před vyhledáváním prvního kladného prvku vrátit hodnoty prvků $x_{i,2}$ až $x_{i,j+1}$ do jim příslušných kapacit a tyto prvky vynulovat. Po odečtení jedničky v prvním kladném prvku a úpravách patřičných kapacit opět začneme generovat předcházející prvky dříve popsaným způsobem.

Efektivnost těchto úvah vyplývá z následujícího tvrzení: Jestliže po vygenerování i-tého řádku nebyla přečerpána kapacita 1. sloupce, pak jistě existuje alespoň jedna tabulka, která je přípustná a prvky v řádcích i-tém až R-tém má totožné s dosud vygenerovanými. Toto tvrzení plyne z analogie s dopravním problémem.

Vytváření všech tabulek je ukončeno tehdy, když všechny prvky $x_{i,j}$ $(i > 1,\ j > 1)$ jsou nulové a nelze tedy nikde provést odečtení jedničky.

3.3 Postupy užitečné při programování nové metody

3.3.1 Odlišné generování u prvku $x_{2,2}$

Mějme vygenerovanou přípustnou tabulku a nechť $x_{2,2} > 0$. Při generování další tabulky bychom tedy odečetli jedničku od prvku $x_{2,2}$. Musíme pak přičíst jedničku k prvkům $x_{1,2}$ a $x_{2,1}$ a odečíst jedničku od prvku $x_{1,1}$. Opakované použití tohoto postupu nenaruší přípustnost tabulky v tom případě, kdy $x_{1,1} \geqq x_{2,2}$. Toho lze využít k naprogramování těchto změn v jednoduchém cyklu. Hodnoty v prvcích o indexech $(1,1)$, $(1,2)$, $(2,1)$ a $(2,2)$ budou dány výrazy

$$x_{1,1} - K,\quad x_{1,2} + K,\quad x_{2,1} + K,\quad x_{2,2} - K \tag{10}$$

kde $K = 0,1,\dots,\min(x_{1,1},\ x_{2,2})$.

Po tomto cyklu přičteme hodnotu x_2 k x_1 a hodnotu x_2 přičteme k x_1

a $x_{2,1}$. Prvek $x_{2,2}$ vynulujeme a teprve nyní začneme hledat kladný prvek.

3.3.2 Uspořádání marginálních součtů

Že vzorců (5) a (7) vyplývá, že jsou invariantní k záměně řádků a sloupců zadané tabulky. Tím tedy ani výpočet hladiny významnosti (6) nezávisí na uspořádání marginálních součtů, přesto, že přípustné tabulky mohou být generovány v jiném pořadí. Vzhledem k algoritmu generování tabulek a vzhledem k odstavci 3.3.1 se jeví nejvýhodnější uspořádání marginálních součtů sestupně podle velikosti. To je výhodné zejména proto, že algoritmus zajišťuje rychleji změny v prvcích o nižších řádkových a v řádku o nižších sloupcových indexech.

3.4 Zhodnocení nové metody

Nová metoda zaručuje generování všech přípustných tabulek, navíc jsou generovány tyto přípustné tabulky ve stejném pořadí, v jakém jsou generovány Marchovou metodou. Proti Marchově metodě zde vzniká podstatně menší počet nepřípustných tabulek, navíc nepřípustnost generované tabulky můžeme zjistit dříve, než March a proces generování zastavit.

Není třeba vypočítávat prvky v 1. řádku, resp. 1. sloupci pomocí rovnic (2), resp. (3) a během jejich výpočtu kontrolovat přípustnost, jako to dělá March. Zde se stačí omezit na prvek $x_{1,1}$ a to v libovolné fázi výpočtu.

Sestupné uspořádání marginálních součtů snižuje podstatně dobu výpočtu a při praktických výpočtech se ukázalo, že vede i ke snížení počtu nepřípustných tabulek. Toto uspořádání by mohlo urychlit i Marchovu metodu.

4. VÝPOČET HLADINY VÝZNAMNOSTI

4.1 Výpočet pravděpodobnosti tabulky

Pravděpodobnost vygenerované tabulky se počítá podle vzorců (7), (8) a (9). Při praktickém využití je vhodné používat logaritmů, což zabrání přetečení. V Marchově metodě je výraz $\ln(R(\mathbb{X}))$ počítán po každé vygenerované přípustné tabulce celý znovu. Ukazuje se, že je účelné zavést tabulku $\mathbb{S}$ součtu logaritmů faktoriálů, jejíž využití urychlí výpočet $\ln(R(\mathbb{X}))$. Tabulka $\mathbb{S}$ má také rozměr R x C a je svázána s tabulkou $\mathbb{X}$ tak, že prvek $s_{i,j}$ obsahuje součet logaritmů faktoriálů všech prvků, lexikograficky následujících za prvkem $x_{i,j}$. Např. prvek $s_{3,C}$ obsahuje součet logaritmů faktoriálů všech prvků tabulky $\mathbb{X}$ od čtvrtého řádku počínaje.

Výpočet prvků tabulky $\mathbb{S}$ se provádí současně s generováním tabulky $\mathbb{X}$. Jestliže před generováním další tabulky $\mathbb{X}$ byla odečtena jednička na prvku $x_{i,j}$, potom tabulka $\mathbb{S}$ bude přepočítána jen v prvcích lexikograficky předcházejících prvku $s_{i,j}$.

4.2 Výpočet hladiny významnosti pomocí doplňku

Pravděpodobnosti $P(\mathbb{X})$ přípustných tabulek, které byly vypočteny v průběhu generování, jsou vždy porovnány s pravděpodobností $P(\mathbb{X}_0)$ zadané tabulky $\mathbb{X}_0$ a jestliže platí $P(\mathbb{X}) \leq P(\mathbb{X}_0)$, jsou přičteny do proměnné, která po skončení generování dává $P_s(\mathbb{X}_0)$.

Tímto způsobem počítá hladinu významnosti March. Rovněž počítá doplněk k hladině významnosti podle vzorce

$$P_s^D(\mathbb{X}_0) = \sum_{P(\mathbb{X}) > P(\mathbb{X}_0)} P(\mathbb{X}) \tag{11}$$

Součet $P_s(\mathbb{X}_0) + P_s^D(\mathbb{X}_0)$ by se měl, až na zaokrouhlovací chyby rovnat jedné. Tohoto vztahu používá pro kontrolu přesnosti výpočtu. Praktické výpočty na ODRE 1204 ukázaly, že přesnost je dostatečně vysoká a tato kontrola je zbytečná. Lze tedy bez podstatné chyby počítat hladinu významnosti i ze vztahu

$$P_s(\mathbb{X}_0) = 1 - P_s^D(\mathbb{X}_0) \tag{12}$$

V nové metodě nepočítáme $P_s(\mathbb{X}_0)$ podle vztahu (6) jako March, ale podle vztahu (12). Omezili jsme se tedy jen na ty tabulky $\mathbb{X}$, pro které $P(\mathbb{X}) > P(\mathbb{X}_0)$. Ze vztahu (7) vyplývá, že u těchto tabulek platí $R(\mathbb{X}) < R(\mathbb{X}_0)$.

5. PRAVDĚPODOBNOSTNÍ OMEZENÍ PŘI GENEROVÁNÍ

5.1 Pravděpodobnostní omezení

Jestliže se při výpočtu hladiny významnosti řídíme vztahem (12), omezíme se jen na tabulky $\mathbb{X}$, pro které $R(\mathbb{X}) < R(\mathbb{X}_0)$. $R(\mathbb{X}_0)$ je tedy horní mez hodnot $R(\mathbb{X})$ pro všechny tabulky $\mathbb{X}$, které je nutné generovat. Ostatní tabulky, které mají $R(\mathbb{X}) \geq R(\mathbb{X}_0)$, nejsou k výpočtu hladiny významnosti potřebné. Je tedy možno je z generování vypustit.

Vzhledem k tomu, že $\ln(R(\mathbb{X}))$ je v nové metodě načítán průběžně v tabulce S, je možné po vygenerování jakéhokoliv prvku $x_{i,j}$ zjistit, zda $s_{i,j} + \ln(x_{i,j}!)$ je větší než $\ln(R(\mathbb{X}_0))$. Jestliže ano, potom již není nutné takovou tabulku dále generovat, neboť pro ni bude jistě platit $R(\mathbb{X}) > R(\mathbb{X}_0)$, ať do zbývajících prvků vygenerujeme cokoliv.

5.2 Funkce MSF

Pro potřeby dalšího výkladu udělejme tuto úvahu. Máme-li rozdělit K jednotek do N políček tak, aby součin faktoriálů /resp. součet logaritmů faktoriálů/ počtu jednotek v těchto políčkách byl minimální, lze jednoduše ukázat, že to bude v tom případě, když budou rozděleny co nejrovnoměrněji /tj. počty v libovolných dvou políčkách se budou lišit nejvýše o jednotku/. K tomu si definujme funkci MSF (K,N), jejíž hodnotou bude minimální součet logaritmů faktoriálů při K jednotkách, rozdělených do N políček.

5.3 Řádková a sloupcová pravděpodobnostní omezení

Podle úvahy v předchozím odstavci 5.2 můžeme pro jednotlivé řádky tabulek určit předem dolní mez součtů logaritmů faktoriálů prvků v těchto řádcích. Tato dolní mez je pro i-tý řádek dána hodnotou funkce MSF (r_i, C). Můžeme tedy pro i-tý řádek určit horní mez RB_i součtů logaritmů faktoriálů prvků v tomto řádku a v řádcích s indexem vyšším pomocí vztahu

$$RB_i = \ln(R(\mathbb{X}_0)) - \sum_{k=1}^{i-1} MSF(r_k, C) \tag{13}$$

Hodnoty RB_i je možno vypočítat před započetím generování, neboť závisí pouze na $R(\mathbb{X}_0)$, r_i a C. Omezení stanovené v bodě 5.1 je možné ještě zostřit tímto způsobem:

Jestliže po vygenerování prvku $x_{i,j}$ zjistíme, že

$$s_{i,j} + \ln(x_{i,j}!) > RB_i , \tag{14}$$

není nutné tabulku dále generovat. Ať je do zbytku tabulky dogenerováno cokoliv, bude vždy $R(\mathbb{X}) > R(\mathbb{X}_0)$ pro každou přípustnou tabulku $\mathbb{X}$, která má v prvku $x_{i,j}$ a v prvcích lexikograficky následujících hodnoty dosud vygenerované.

Obdobně jako pro řádky je možno uplatňovat během výpočtu i omezení pro sloupce, neboť zbývající sloupcové kapacity je nutno určitým způsobem do dosud nevygenerovaných prvků rozdělit. Je zřejmé, že pokud

$$a_{i,j} + \ln(x_{i,j}!) > \ln(R(\mathbb{X}_0)) - \sum_{k=1}^{j-1} MSF(x_{i,k}, i) - \sum_{k=j}^{C} MSF(x_{i,k}, i-1), \quad (15)$$

je zbytečné tabulku dogenerovávat.

Tato omezení však již závisí na volných kapacitách a na indexech i,j a není proto možné je vypočítat předem.

Průběžná kontrola podle nerovnosti (15) by časově příliš zatěžkávala proces generování a proto jsou předchozí úvahy prakticky realizovány tak, že nerovnost (14) je kontrolována při generování každého prvku. Teprve když je splněna, jsou prováděny kontroly jak pro sloupcové omezení (15), tak i pro zostřené řádkové omezení, dané nerovností

$$a_{i,j} + \ln(x_{i,j}!) > RS_i - MSF(x_{i,1}, j-1) . \quad (16)$$

Pokud je pro prvek $x_{i,j}$ splněna alespoň jedna z nerovností (15) a (16) a $x_{i,j} > 0$, odečteme od $x_{i,j}$ jedničku a provedeme příslušné úpravy kapacit $x_{i,1}$ a $x_{1,j}$. Opět testujeme splnění nerovností (15) a (16). Jestliže již $x_{i,j} = 0$ a dosud alespoň jedna z nerovností (15) a (16) platí, pokračujeme v obdobném testování prvku lexikograficky následujícího. Tento zpětný proces je ukončen, jestliže obě nerovnosti nejsou splněny /pak se pokračuje v generování/, případně tehdy, když jsme v tomto procesu dospěli k poslednímu prvku tabulky $x_{R,C}$ a tento je již nulový, přičemž alespoň jedna z nerovností (15) a (16) platí /pak je generování tabulek u konce/. Při přechodu na nižší řádek je v tomto procesu samozřejmě nutné upravit i kapacitu 1. sloupce /k hodnotě $x_{1,1}$ přičíst hodnotu $x_{i,1}$ /.

Tato strategie generování vyžaduje složitější program, ale nezatěžká zbytečnými časovými ztrátami výpočty pro tabulky s nízkou hladinou významnosti.

5.4. Úprava generování u prvku $x_{2,2}$

Postup uvedený v bodě 3.3.1 můžeme při využívání pravděpodobnostních omezení v generování upravit následujícím způsobem:

Množinu těch K, která vystupují v (10) můžeme rozdělit na dvě podmnožiny, pro které se $R(\mathbb{X})$ chovají monotonně.

$$M_1 = \{K;\ K \leq L\},$$
$$M_2 = \{K;\ K > L\},$$

kde

$$L = \frac{x_{1,1} \cdot x_{2,2} - (x_{1,2} + 1) \cdot (x_{2,1} + 1)}{x_{1,1} + x_{1,2} + x_{2,1} + x_{2,2} + 2} \quad (17)$$

V množině M_1 se pro rostoucí K $R(\mathbb{X})$ snižuje, v M_2 naopak zvyšuje. Je možné, že jedna z množin je prázdná.

Výpočet pak budeme provádět raději ve dvou cyklech, odpovídajících množinám M_1 a M_2, v M_1 postupně pro klesající K, v M_2 pro rostoucí. Nemusíme pak zbytečně počítat s tabulkami, s $R(\mathbb{X}) > R(\mathbb{X}_0)$.

V případě, že nebyla v obou cyklech nalezena ani jedna tabulka $\mathbb{X}$, pro kterou $R(\mathbb{X}) \leq R(\mathbb{X}_0)$, nepokračujeme přímo v generování dalších tabulek, ale počínaje prvkem $x_{2,2}$ provádíme testování vzhledem k pravděpodobnostním omezením podle bodu 5.3.

5.5 Zhodnocení metody generování při pravděpodobnostních omezeních

Metoda popsaná v kapitole 5 má proti metodě generování všech přípustných tabulek z kapitoly 3 tu přednost, že z procesu generování vynechává tabulky pravděpodobnostně nevhodné, tj. tabulky, pro které $P(X) \leq P(X_0)$. Z toho plyne, že efektivnost této metody se projeví podstatně především v těch případech, kdy $P(X_0)$ bude co největší vzhledem k pravděpodobnostem ostatních přípustných tabulek. Algoritmus je ovšem navržen tak, aby nezdržoval výpočet hladiny významnosti u tabulek, pro které $P(X_0)$ je malé.

Vezmeme-li v úvahu výsledky kapitoly 3 a této kapitoly, je zřejmé, že požadovanou hladinu významnosti $P_s(X_0)$ získáme v podstatně kratším čase, než March.

6. PROCEDURA CONP

6.1 Popis procedury

Nová metoda, jejíž popis byl uveden v předcházejících kapitolách 3 až 5, byla zpracována do algolské procedury. Hlavička procedury má tento tvar:

```
PROCEDURE CONP (MATRIX, R, C, PT, PS);
VALUE MATRIX, R, C;
INTEGER R, C;
REAL PT, PS;
INTEGER ARRAY MATRIX;
```

Parametry mají tento význam:

MATRIX	Dvourozměrné celočíselné pole s mezemi [1:R, 1:C], které obsahuje zadanou kontingenční tabulku X_0.
R	Počet řádků tabulky X_0
C	Počet sloupců tabulky X_0
PT	Výstupní parametr; při návratu z procedury obsahuje pravděpodobnost $P(X_0)$
PS	Výstupní parametr, při návratu z procedury obsahuje hladinu významnosti $P_s(X_0)$

Procedura pracuje s nelokálními proměnnými typu ARRAY. Je to jednorozměrné pole FL $[0:\alpha]$, jehož prvky obsahují logaritmy faktoriálů, tedy FL $[I] = \ln(I!)$ a dvourozměrné pole MSFL $[0:\beta, 1:\mu]$, kde jednotlivé prvky tohoto pole obsahují minimální součet logaritmů faktoriálů, tedy MSFL [K,N] = MSF (K,N), viz fce MSF, definovaná v bodě 5.2. Horní meze α, β, μ v deklaracích těchto polí je nutno volit s ohledem na rozměry tabulky a počty pozorování v tabulce. Hodnota α musí být větší nebo rovna celkovému počtu pozorování v tabulce X_0, hodnota β větší nebo rovna největšímu z řádkových a sloupcových marginálních součtů, hodnota μ větší nebo rovna maximu z hodnot R a C. Pokud tyto podmínky nejsou dodrženy, může dojít k chybě při práci procedury /SUBSCRIPT/. Hodnoty do polí FL a MSFL je nutno pochopitelně dosadit před vyvoláním procedury CONP. Prvky nelokálních polí FL a MSFL je možno samozřejmě nahradit funkčními procedurami, které poskytnou požadované hodnoty, to ale podstatně prodlouží dobu výpočtu.

6.2 Doba výpočtu

Odhad doby výpočtu nelze v jednotlivých případech stanovit. Lze jen uvést určitá fakta, ze kterých je možné usuzovat na složitost a tedy i délku výpočtu příkladu. Délka výpočtu závisí obecně na počtu přípustných tabulek, u kterých $P(X) > P(X_0)$. Obtížnost je zde dána především neznalostí celkového počtu přípustných tabulek pro dané marginální součty a neznalostí rozvrstvení jejich pravděpodobností. Vzorec pro výpočet počtu přípustných tabulek nám není znám, ze zkušenosti však víme, že tento počet závisí především na velikosti tabulky /jejích rozměrech/ a dále na velikosti a rozdělení marginálních součtů v řádcích a sloupcích. Čím rovnoměrněji jsou marginální součty v řádcích, resp. sloupcích rozděleny, tím větší počet přípustných tabulek můžeme očekávat. Také se vzrůstajícím počtem pozorování v tabulce počet přípustných tabulek narůstá.

Z praktického hlediska, jak ukázaly výpočty zkušebních příkladů, je vhodné zadávat tabulku X_0 vždy tak, aby $R \geqq C$. V tomto případě dochází k lepšímu využití pravděpodobnostních omezení RB_i v jednotlivých řádcích, proces generování pravděpodobnostně nevhodných tabulek $(P(X) \leqq P(X_0))$ je zastavován dříve.

7. APLIKACE METODY

7.1 Náhrada χ^2-testu dobré shody

Je známo, že není vhodné používat χ^2-testu, jestliže napozorované četnosti jsou nízké. Při srovnání s přesným testem se ukazuje, že testové kriterium χ^2 není vhodné užívat ani jako přibližnou míru závislosti v tabulce. Proti jeho použití svědčí dvě skutečnosti. Jednak to, že výsledky získané χ^2-testem se mohou značně lišit od výsledků získaných přesným testem a to tím více, čím menší jsou napozorované četnosti. Druhá skutečnost je ta, že výsledek přesného testu závisí jen na pravděpodobnosti a marginálních součtech zadané tabulky, zatímco χ^2 je závislý ne na pravděpodobnosti $P(X_0)$, ale konkrétních napozorovaných četnostech. Pro dvě různé tabulky se stejnou pravděpodobností a se stejnými marginálními součty dává přesný test tutéž hladinu významnosti, zatímco χ^2-test může dát hodnoty $P_s(X_0)$ značně odlišné. To se projevuje podstatně u tabulek s nízkými četnostmi.

S ohledem na to, že doba výpočtu hladiny významnosti $P_s(X_0)$ při použití přesného testu rychle roste s rostoucími rozměry tabulky a s rostoucími četnostmi pozorování, nelze bohužel počítat s jeho hromadným použitím jako náhradou za běžně používaný χ^2-test. Přesného testu lze však použít v těch případech, kdy podmínky pro aplikaci χ^2-testu nejsou dobře splněny.

7.2 Dopravní problém

Generování všech přípustných tabulek, které je součástí metody popsané v kapitole 3, je z hlediska matematického programování výčtem všech přípustných řešení celočíselného dopravního problému. Toho lze využít při minimalizaci nákladů v dopravních problémech, jejichž účelová funkce není lineární. Je-li účelová funkce dopravního problému taková, že její předpis je shodný pro všechna přípustná řešení a příspěvek každého prvku matice přepravovaných množství je nezávislý na hodnotách jiných prvků této matice a přitom je nezáporný, můžeme při minimalizaci použít obdobných úvah, jako v bodě 4.1 a kapitole 5 této práce. Navíc omezení, analogické pravděpodobnostnímu omezení uvedenému v bodě 5.1 lze upravovat během výpočtu podle dosud zjištěného minima účelové funkce.

8. PŘÍKLADY

Pro ilustraci uvádíme dva příklady, na nichž bude demonstrováno srovnání uvedených metod výpočtu hladiny významnosti. Jednotlivé metody označme takto:

1. Výpočet pomocí χ^2-testu.
2. Výpočet Marchovou metodou.
3. Jako 2, ale pro transponovanou tabulku.
4. Výpočet novou metodou generování tabulek bez pravděpodobnostních omezení /generovány všechny přípustné tabulky/.
5. Jako 4, ale pro transponovanou tabulku.
6. Výpočet novou metodou generování tabulek s pravděpodobnostními omezeními.
7. Jako 6, ale pro transponovanou tabulku.

Příklad 1.

Tabulka typu 3 x 3			metoda	hladina významnosti	počet generovaných připust. tabulek	doba výpočtu s
2	3	1	1	0.53831	-	-
0	1	2	2	0.71218	278	37
6	5	5	3	0.71218	278	39
			4	0.71218	278	8
			5	0.71218	278	9
			6	0.71218	13	1
			7	0.71218	13	2

Příklad 2.

Tabulka typu 3 x 4				metoda	hladina významnosti	počet generovaných připust. tabulek	doba výpočtu s
3	7	4	8	1	0.46631	-	-
5	2	3	3	2	?	?	3600
2	1	0	3	3	?	?	3600
				4	0.50790	31892	762
				5	0.50790	31892	768
				6	0.50790	818	70
				7	0.50790	818	53

Výpočet metodou 2 a 3 byl zastaven po jedné hodině bez získání výsledků.

LITERATURA:

1. Freeman, G. H., Halton, J. H.:
Note on an exact treatment of contingency, goodness of fit, and other problems of significance.
Biometrika, 38(1951), 141-149.

2. March, D. L.:
Exact probabilities for R x C contingency tables. Algorithm 434. Communications of the Association for Computing Machinary 15(1972) 11, 991-992.

3. Stucky, W., Vollmar, J.:
Ein Verfahren zur exakten Auswertung von 2 x C - Häufigkeitstafeln.
Biometrische Zeitschrift 17(1975) 3, 147-162.