

Algebraiczna metoda próbkowania z dyskretnych rozkładów warunkowych

Opracował: Piotr Zwiernik

Streszczenie

Omawiamy algorytm Diaconisa–Sturmfelsa generowania próbek z przestrzeni skończonej \mathcal{X} z rozkładu warunkowego względem statystyki dostatecznej T . Algorytm Metropolisa-Hastingsa pozwala nam zmodyfikować dowolny nieprzywiedlny, nieokresowy łańcuch Markowa tak, aby jego rozkład stacjonarny był odpowiedni. Problemem jednak jest konstrukcja samego łańcucha. Rozwiązuje to właśnie algorytm Diaconisa-Sturmfelsa.

Algorytm ten wykorzystuje ciekawą zależność pomiędzy nieprzywiedlnym łańcuchem Markowa na tej przestrzeni, a pewnym ideałem torycznym w pierścieniu wielomianów, w którym zmienne reprezentują elementy przestrzeni \mathcal{X} . Jest to jedno z kilku ciekawych zastosowań algebry przemiennej w statystyce.

W opracowaniu tym zaprezentujemy również, jak wykorzystywać programy do obliczeń symbolicznych w znajdowaniu ruchów konstruowanego łańcucha Markowa. Zaprezentowany zostanie również specjalistyczna aplikacja `4ti2`, która w dużym stopniu automatyzuje analizę dowolnych modeli hierarchicznych dla danych dyskretnych.

Spis treści

1	Wprowadzenie	2
2	Algorytm wykorzystujący bazy Markowa	6
2.1	Bazy Markowa, algorytm próbkowania.	7
2.2	Szybkość zbieżności	9
3	Algebraiczne metody szukania baz Markowa	9
3.1	Preliminaria algebraiczne	9
3.2	Bazy Markowa i ideały	12
4	Wykorzystanie komputera do znajdowania generatorów ideału \mathcal{I}_T	14
4.1	Dwa sposoby opisu ideału \mathcal{I}_T	15
4.2	Obliczenia w CoCoA	16
4.3	Obliczenia w <code>4ti2</code>	18

5	Przykłady	20
5.1	Trójwymiarowe tablice kontyngencji	20
5.1.1	Modele tablic kontyngencji	20
5.1.2	Model bez interakcji trójstronnych.	22
5.2	Regresja logistyczna	23
6	Funkcja napisana w programie Matlab	25

1 Wprowadzenie

W ostatnich latach coraz popularniejsze staje się wykorzystanie metod algebry obliczeniowej w statystyce. Miało to swój wyraz w wydanej w 2000 roku monografii Pistone, Wynn i Riccomagno [PRW01]. Pierwsze ważne prace pojawiły się nie dużo wcześniej. Jedną z ważniejszych jest artykuł Diaconisa i Sturmfelsa [DS98]. W artykule tym autorzy przedstawili konstrukcje algorytmu, który pozwala z warunkowych rozkładów, pod warunkiem statystyk dostatecznych, dla dyskretnych rodzin wykładniczych. Metoda ta stanowi ciekawe uzupełnienie dla tradycyjnej teorii asymptotycznej oraz testów dokładnych, które użyteczne są dla małych problemów.

Przedstawimy w pierw najprostszy możliwy przypadek: badanie niezależności dwóch zmiennych dyskretnych na podstawie danych z dwuwymiarowych *tablic kontyngencji* X^1 . Przy założeniu niezależności rozkłady brzegowe (czyli sumy po wierszach r i po kolumnach c) tej tablicy stanowią statystyki dostateczne analizowanego modelu. Przez \hat{X} będziemy oznaczać macierz będącą estymatorem największej wiarygodności tablicy X . Założmy, że X jest wymiaru $m \times n$ oraz jej rozkłady brzegowe wynoszą odpowiednio $r = (r_1, \dots, r_m)$ oraz $c = (c_1, \dots, c_n)$. Zatem $\hat{X} = \frac{1}{N} c \cdot r$, gdzie $N = \sum_{i=1}^n r_i = \sum_{i=1}^m c_i$ oznacza wielkość próby. Badanie niezależności dla tablicy X to badanie jak daleko są od siebie tablice X i \hat{X} . Odpowiednia metryka zadana jest na przestrzeni $\mathcal{F}(r, c)$ wszystkich tablic $B \in \mathbb{N}^{m \times n}$ o takich samych rozkładach brzegowych jak X :

$$\mathcal{F}(r, c) = \{B \in \mathbb{N}^{m \times n} : \text{sumy po wierszach}(B) = r, \text{sumy po kolumnach}(B) = c\}.$$

Przykład 1.1 Niech $m = 2$ $n = 3$ i rozważmy X jako macierz złożoną z jedynek. Mamy:

$$\mathcal{F}((3, 3), (2, 2, 2)) = \left\{ \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 2 & 1 \\ 2 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 2 & 0 & 1 \\ 0 & 2 & 1 \end{bmatrix}, \right. \\ \left. \begin{bmatrix} 0 & 1 & 2 \\ 2 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 2 & 1 & 0 \\ 0 & 1 & 2 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 2 \\ 1 & 2 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 2 & 0 \\ 1 & 0 & 2 \end{bmatrix} \right\}.$$

W tym prostym przypadku zatem weryfikacja hipotezy o niezależności dwóch zmiennych sprowadza się obliczeń bezpośrednich, które tablice są bardziej prawdopodobne przy założeniu niezależności obu zmiennych.

¹W dalszej części tekstu z reguły macierz kontyngencji będziemy utożsamiać z funkcją, która każdej komórce przyporządkowuje pewną liczbę naturalną. Tablicę kontyngencji będziemy wówczas oznaczać przez f . Dobrym wprowadzeniem do tematyki tablic kontyngencji jest zbiór wykładów Steffena L. Lauritzena [Lau02], które dostępne są w Sieci.

Bardzo prosty przykład tego rodzaju podał Fisher w swoim tekście z 1935 roku „*The Design of Experiments*” (przykład podajemy za Agresti [Agr96]). Fisher opisał następujący eksperyment: Koleżanka Fishera twierdziła, że pijąc herbatę (z mlekiem) potrafi rozróżnić, czy to mleko zostało dodane do herbaty, czy herbata do mleka. Aby sprawdzić, czy to nie czcze przechwałki, Fisher zaprojektował następujący eksperyment. Koleżance podał osiem filiżanek, z których cztery były jednego typu, a cztery drugiego. Koleżanka miała zaklasyfikować każdą filiżankę do odpowiedniej grupy, wiedząc, że w każdej grupie powinny się znaleźć dokładnie cztery filiżanki. Filiżanki zostały podane w losowej kolejności. Wyniki eksperymentu zawiera poniższa tabela:

	zaklasyfikowano		
wlane w pierw	mleko	herbata	razem
mleko	3	1	4
herbata	1	3	4
razem	4	4	

Hipoteza zerowa mówiła, że wybór przyporządkowania przez koleżankę jest niezależny. Skoro wiadomo było, że jest dokładnie cztery filiżanek w każdej z grup, rozkłady brzegowe możliwych wyników są takie same, zatem istotnie można porównać tylko macierze o tych samych rozkładach brzegowych.

W statystyce stosuje się zwykle jedną z wielu popularnych metryk, np. statystykę χ^2 , lub statystykę ilorazu wiarygodności G^2 (patrz Agresti [Agr02]) aby wyliczyć odległość $d(X, \hat{X})$. Statystykę χ^2 tą wylicza się według wzoru:

$$\chi^2 = \sum \frac{(n_{ij} - m_{ij})^2}{m_{ij}},$$

gdzie n_{ij} to elementy tablicy X a m_{ij} to elementy macierzy \hat{X} .

Twierdzenie 1.2 *Jeśli dla każdego i, j zachodzi $n_{ij} > 5$ wówczas statystyka χ^2 Pearsona dla testowania niezależności w tablicy kontyngencji X ma asymptotycznie rozkład χ^2 z $(m - 1)(n - 1)$ stopniami swobody.*

Jeśli zachodzi warunek z twierdzenia 1.2, możemy stosować testy asymptotyczne, taki jak np. test Pearsona. Problem pojawia się w przypadku macierzy rzadkich, które pojawiają się bardzo często w konkretnych zastosowaniach, kiedy to statystyka χ^2 ma nieznaną rozkład, więc nie możemy przy testowaniu odwoływać się do wartości krytycznych testu z tablic statystycznych. Zauważmy również, że w ogólności tablica największej wiarygodności nie ma elementów będących liczbami naturalnymi i w przypadku rzadkich macierzy w ogóle może nie istnieć tablica o elementach naturalnych dostatecznie bliska macierzy \hat{X} .

W ogólności jesteśmy zainteresowani odsetkiem tablic B , które są bliżej do \hat{X} niż X . Formalnie, chcemy policzyć

$$\frac{\#\{B \in \mathcal{F}(r, c) : d(B, \hat{X}) < d(X, \hat{X})\}}{\#\mathcal{F}(r, c)}$$

		Kolor włosów				Suma
		Ciemne	Brazowe	Rude	Blond	
Kolor oczu	Brazowe	68	119	26	7	220
	Niebieskie	20	84	17	94	215
	Zielone	15	54	14	10	93
	Piwne	5	29	14	16	64
	Suma	108	286	71	127	592

Tabela 1: Prosty przykład tablicy kontyngencji

Tablicę X będziemy uważali za statystycznie niezależną, jeśli powyższy odsetek nie przekracza pewnej z góry ustalonej liczby α (z reguły $\alpha = 0,05$). W przypadku, gdy rozkład statystyki nie jest znany możemy zastosować markowowskie metody Monte Carlo, umożliwiającą nam losowanie z tego nieznanego rozkładu, a w konsekwencji znalezienie dobrego przybliżenia analizowanego ilorazu.

W konkretnych przypadkach rozważanych przez statystyków, włókna $\mathcal{F}(r, c)$ zawierają znacznie więcej elementów niż w podanym wcześniej przykładzie. Na przykład, dla tablicy 1 mamy

$$\#\mathcal{F}((108, 286, 71, 127), (220, 215, 93, 64)) = 1.225.914.276.768.514$$

(patrz <http://www.math.ucdavis.edu/~deloera/contingency.html>).

Jeżeli moc zbioru $\mathcal{F}(r, c)$ nie jest zbyt duża - powiedzmy - mniej niż milion, wówczas można wypisać wszystkie elementy włókna. Z drugiej strony, w wielu zastosowaniach, takie wypisanie staje się niemożliwe z uwagi na wysokie koszty. W takim przypadku interesuje nas możliwość generowania tablic z dużego włókna według jakiegoś z góry założonego rozkładu - np. rozkładu jednostajnego. Jest to szczególny przypadek zastosowania algorytmu Diaconisa–Sturmfelsa [DS98].

Rozważmy problem generowania losowej tablicy $X = (x_{ij})$ z przestrzeni $\mathcal{F}(r, c)$. Niech

$$H = \frac{\prod_i \binom{c_i}{x_{1i} \cdots x_{mi}}}{\binom{N}{r_1 \cdots r_m}}, \quad \text{gdzie } N = \sum_{i=1}^m r_i = \sum_{i=1}^n c_i, \quad (1.1)$$

będzie rozkładem hipergeometrycznym na $\mathcal{F}(r, c)$. Jest to rozkład warunkowy danych pod warunkiem statystyk dostatecznych r i c dla klasycznego modelu niezależności².

Metoda Monte Carlo generowania tablic z \mathbf{H} przebiega w następujący sposób. Niech $X \in \mathbb{N}^{m \times n}$ będzie tablicą o zadanych rozkładach brzegowych. Wybierz losowo parę wierszy i parę kolumn. Przecinając się one w czterech miejscach tablicy, a X jest modyfikowane o

$$\begin{matrix} +1 & -1 \\ -1 & +1 \end{matrix} \text{ albo } \begin{matrix} -1 & +1 \\ +1 & -1 \end{matrix} \text{ z prawdopodobieństwem } \frac{1}{2} \text{ każde.}$$

²Rzeczywiście:

$$P(X|r, c) = \frac{P(X, r|c)}{P(r)} = \frac{P(X|c)}{P(r)},$$

bo jeżeli znamy c to znajomość X determinuje już r .

Ta modyfikacja nie zmienia rozkładów brzegowych. Jeśli zmiana powoduje wystąpienie ujemnego elementu w którymś miejscu to musimy zaniechać ruchu i znów wylosować parę kolumn i wierszy. To definiuje nieprzywiedlny i nieokresowy łańcuch Markowa na $\mathcal{F}(r, c)$. Stosując standardową procedurę Metropolisa-Hastingsa, którą omówimy dla tego przypadku później, możemy łańcuch zmodyfikować do nieprzywiedlnego, nieokresowego, odwracalnego łańcucha Markowa o rozkładzie stacjonarnym H .

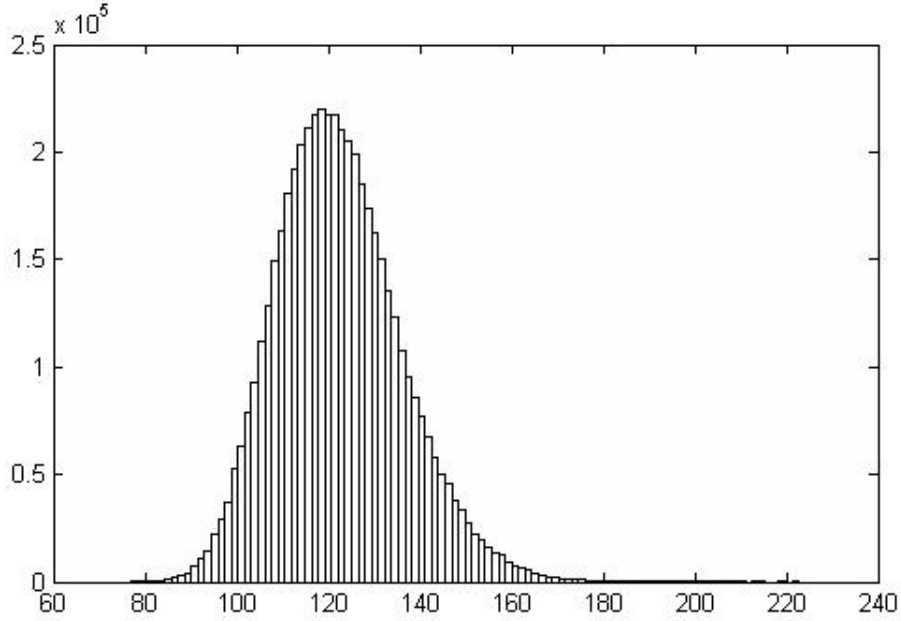
Przykład 1.3 Tabela 2 zawiera dane zebrane w celu weryfikacji zależności pomiędzy datą narodzin a datą śmierci (patrz [AH85]). Dane te dotyczą miesięcy narodzin i śmierci 82 przodków królowej Wiktorii. Jak twierdzą Diaconis i Sturmfels [DS98] powszechnie wysuwa się twierdzenie, że zmienne te wykazują zależność. Istotnie, przy wartości statystyki $\chi^2 = 115,6$ oraz $\nu = 121$ stopniach swobody, zastosowanie testu Pearsona sugeruje odrzucenie hipotezy o niezależności. Zauważmy jednak, że żadna z komórek tej tabeli nie przekracza wartości 5, a zatem reguły stosowalności testu Pearsona są mocno złamane. Z drugiej strony przeprowadzenie testu dokładnego jest niemożliwe z powodu dużego rozmiaru tablicy 2 (jest zbyt wiele tablic o tych samych rozkładach brzegowych).

Tabela 2: Zależności pomiędzy miesiącem narodzin a miesiącem śmierci.

Narodziny	Miesiąc śmierci											suma	
	sty	lut	mar	kwi	maj	cze	lip	sie	wrz	paź	lis		gru
sty	1	0	0	0	1	2	0	0	1	0	1	0	6
lut	1	0	0	1	0	0	0	0	0	1	0	2	5
mar	1	0	0	0	2	1	0	0	0	0	0	1	5
kwi	3	0	2	0	0	0	1	0	1	3	1	1	12
maj	2	1	1	1	1	1	1	1	1	1	1	0	12
cze	2	0	0	0	1	0	0	0	0	0	0	0	3
lip	2	0	2	1	0	0	0	0	1	1	1	2	10
sie	0	0	0	3	0	0	1	0	0	1	0	2	7
wrz	0	0	0	1	1	0	0	0	0	0	1	0	3
paź	1	1	0	2	0	0	1	0	0	1	1	0	7
lis	0	1	1	1	2	0	0	2	0	1	1	0	9
gru	0	1	1	0	0	0	1	0	0	0	0	0	3
suma	13	4	7	10	8	4	5	3	4	9	7	8	82

Przeprowadzamy zatem opisany powyżej eksperyment Monte Carlo³. Na rysunku 1 został przedstawiony histogram statystyk χ^2 dla $5 \cdot 10^6$ tablic, które pojawiały się w kolejnych krokach. Tablicą startową była analizowana tablica 2. Okazuje się, że test asymptotyczny przeszacowuje prawdopodobieństwo, że $\chi^2 \leq 115,6$. W teście asymptotycznym wynosi ono 0,3775 (ok. 37,75%). W przeprowadzonej symulacji ok. 32,95% tablic charakteryzowało się wartością statystyki χ^2 mniejszą niż 115,6.

³Nie pokazaliśmy jeszcze, że tak zdefiniowany algorytm rzeczywiście ma dobre własności - w szczególności nie pokazaliśmy jak szybko łańcuch Markowa zbiega do swojego rozkładu stacjonarnego.



Rysunek 1: Histogram z eksperymentu Monte Carlo dla przykładu 1.3.

2 Algorytm wykorzystujący bazy Markowa

Niech \mathcal{X} będzie zbiorem skończonym. Interesować nas będą te klasy modeli, które opisywane są rodzinami wykładniczymi postaci:

$$P_\theta(x) = Z(\theta) \cdot e^{\langle \theta, T(x) \rangle}, \quad \theta \in \mathbb{R}^d, \quad (2.1)$$

gdzie $x \in \mathcal{X}$, $Z(\theta)$ jest stałą normującą, $\langle \cdot, \cdot \rangle$ oznacza iloczyn wewnętrzny a $T : \mathcal{X} \rightarrow \mathbb{N}^d \setminus \{0\}$ stanowi statystyki dostateczne (często utożsamia się je z macierzą wymiaru $d \times |\mathcal{X}|$). Jeśli X_1, X_2, \dots, X_N są niezależne, o tych samych rozkładach, wówczas statystyka $t = T(X_1) + \dots + T(X_N)$ jest statystyką dostateczną dla θ . Niech

$$\mathcal{Y}_t = \{(x_1, \dots, x_N) \in \mathcal{X}^N : T(x_1) + \dots + T(x_N) = t\}. \quad (2.2)$$

Pod warunkiem t rozkład prawdopodobieństwa warunkowego (X_1, \dots, X_N) jest jedno-stajny na \mathcal{Y}_t . Ogólnie jednak ciężko jest wyliczyć elementy przestrzeni \mathcal{Y}_t lub próbkować z rozkładu jednostajnego na \mathcal{Y}_t^4 .

Zwykle formułuje się ten problem w terminach rozkładu hipergeometrycznego w następujący sposób. Możemy napisać

$$t = \sum_{i=1}^N T(X_i) = \sum_{x \in \mathcal{X}} G(x) T(x), \quad \text{gdzie } G(x) = \#\{i : X_i = x\}.$$

⁴Uwaga: to jest inna przestrzeń niż $\mathcal{F}(r, c)$ w poprzednim paragrafie. Tutaj interesuje nas jakie **kolejne** realizacje mogą prowadzić np. do tablicy kontyngencji o ustalonych rozkładach brzegowych. W analizie danych *iid* nie interesuje nas kolejność realizacji, co motywuje przeformułowanie modelu (patrz dalej).

Kluczowe jest, że liczebności $G(x)$ tworzą statystykę dostateczną dla dowolnych danych *iid*. Zdefiniujemy zbiór wszystkich możliwych danych o ustalonej statystyce dostatecznej jako:

$$\mathcal{F}_t = \{f : \mathcal{X} \rightarrow \mathbb{N} : \sum_{x \in \mathcal{X}} f(x)T(x) = t\}. \quad (2.3)$$

Na przykład w przypadku tablic kontyngencji element f odpowiada tablicy o rozkładach brzegowych $t = (r, c)$.

Skoro $T(x)$ jest niezerowe dla wszystkich x , a zawsze startujemy z jakąś tablicą o statystyce dostatecznej t , to w szczególności \mathcal{F}_t jest niepusty i skończony (bo \mathcal{X} jest skończony, a jądro T przecięte z \mathbb{N}^d daje zbiór pusty). Obraz rozkładu jednostajnego na \mathcal{Y}_t względem przekształcenia z \mathcal{Y}_t do \mathcal{F}_t nazywamy rozkładem hipergeometrycznym

$$H_t(f) = \frac{N!}{|\mathcal{Y}_t|} \cdot \frac{1}{\prod_{x \in \mathcal{X}} f(x)!} \quad (2.4)$$

Problem losowania jednostajnie z \mathcal{Y}_t został zatem zredukowany do losowania z H_t na \mathcal{F}_t przy zadanym t . Warto dodać, że jak się później okaże, problem znalezienia $|\mathcal{Y}_t|$ nie jest istotny.

Dla dwuwymiarowych tablic kontyngencji mamy $\mathcal{X} = \{(i, j), 1 \leq i \leq m, 1 \leq j \leq n\}$. Dla klasycznego modelu niezależności $T((i, j)) \in \mathbb{N}^{m+n}$ wektor długości $m + n$, którego dwa elementy są jedynekami a reszta zerami. Jedyнки w $T((i, j))$ znajdują się na pozycji i oraz $m + j$. Statystyka dostateczna zawiera rozkłady brzegowe tablicy kontyngencji dla N obserwacji, tzn. $t = (r, c)$. Zbiór \mathcal{F}_t jest zbiorem wszystkich tablic wymiaru $m \times n$ o ustalonych rozkładach brzegowych. Rozkład hipergeometryczny 2.4 odpowiada klasycznej wersji z równania 1.1.

2.1 Bazy Markowa, algorytm próbkowania.

Algorytm przedstawiony w pracy Diaconisa i Sturmfelsa [DS98] polega na znalezieniu uogólnienia ruchów typu $\begin{matrix} + & - \\ - & + \end{matrix}$ dla ogólnych dyskretnych rodzin wykładniczych.

Definicja 2.1 *Skończony zbiór $f_1, \dots, f_L : \mathcal{X} \rightarrow \mathbb{Z}$ (tutaj $\mathbb{Z} = 0, \pm 1, \pm 2, \dots$) nazywamy bazą Markowa dla T jeśli*

1.

$$\sum_{x \in \mathcal{X}} f_i(x)T(x) = 0, \quad 1 \leq i \leq L, \quad (2.5)$$

2. dla każdego t i każdej pary $f, f' \in \mathcal{F}_t$ istnieją f_{i_1}, \dots, f_{i_K} takie, że

$$f' = f + \sum_{j=1}^K \epsilon_j f_{i_j} \quad \text{oraz} \quad f + \sum_{j=1}^k \epsilon_j f_{i_j} \geq 0 \quad \text{dla} \quad 1 \leq k \leq K, \quad (2.6)$$

gdzie $\epsilon_i = \pm 1$.

Baza Markowa pozwala skonstruować łańcuch Markowa na \mathcal{F}_t . Będąc w $f \in \mathcal{F}_t$ wybierz i jednostajnie ze zbioru $\{1, 2, \dots, L\}$ oraz $\epsilon = \pm 1$. Jeśli $f + \epsilon f_i$ jest nieujemne (tzn. $(f + \epsilon f_i)(x) \geq 0$ dla każdego $x \in \mathcal{X}$), łańcuch zmienia stan na $f + \epsilon f_i$. W pozostałych przypadkach łańcuch pozostaje w f .

Z definicji wynika kilka podstawowych własności tego łańcucha. Warunek 2.5 sprawia, że w każdym kroku jesteśmy w \mathcal{F}_t . Istotnie, bo jeśli $\sum f(x)T(x) = t$ to

$$\sum (f + \epsilon f_i)(x)T(x) = \sum f(x)T(x) + \epsilon \sum f_i(x)T(x) \stackrel{\text{war. 2.5}}{=} \sum f(x)T(x) = t.$$

Warunek 2.6 zapewnia, że każde dwa elementy \mathcal{F}_t wzajemnie się komunikują, tzn. łańcuch jest nieprzywiedlny. Istnieje bowiem ścieżka pomiędzy nimi taka, że w każdym kroku pozostajemy w przestrzeni \mathcal{F}_t . W tej chwili w najmniejszy sposób nie jest oczywiste, że baza Markowa w ogóle istnieje. W dalszych rozważaniach nie tylko dowiedzimy jej istnienia, ale również pokażemy jak w efektywny sposób jej szukać.

Poniższy lemat mówi, jak poprawić tak skonstruowany łańcuch Markowa by generowane próby rzeczywiście pochodziły z rozkładu H_t (lub jakiegoś innego).

Lemat 2.2 *Niech $\sigma(g)$ będzie dodatnią funkcją na \mathcal{F}_t . Niech f_1, \dots, f_L będzie bazą Markowa. Skonstruujmy łańcuch Markowa w następujący sposób: wybierzmy i jednostajnie ze zbioru $\{1, 2, \dots, L\}$ oraz $\epsilon = \pm 1$ z prawdopodobieństwem 0,5 niezależnie od i . Jeśli łańcuch jest w $g \in \mathcal{F}_t$, rusza się do $g + \epsilon f_i$ (o ile to jest nieujemne) z prawdopodobieństwem*

$$\min\left\{\frac{\sigma(g + \epsilon f_i)}{\sigma(g)}, 1\right\}.$$

W pozostałych przypadkach łańcuch pozostaje w g . Tak skonstruowany łańcuch Markowa na \mathcal{F}_t jest nieprzywiedlny, odwracalny i nieokresowy. Jego rozkład stacjonarny jest proporcjonalny do $\sigma(g)$.

Dowód: Niech Q będzie wyjściowym łańcuchem Markowa bez kroku Metropolisa. Przez P będziemy oznaczać łańcuch skonstruowany w lemacie. Oczywiście $Q(f, f') = Q(f', f)$, bo jeśli $f' = f + \epsilon f_i$ (czyli jeśli przejście z f do f' w jednym kroku ma niezerowe prawdopodobieństwo) to $f = f' - \epsilon f_i$. Łatwo sprawdzić, że

$$\sigma(g)P(g, g') = \sigma(g')Q(g', g) \min\left\{\frac{\sigma(g')}{\sigma(g)}, 1\right\} = \sigma(g')P(g', g).$$

Z warunku 2.6 mamy nieprzywiedlność P . Łańcuch jest nieokresowy, bo dla pewnych g istnieje dodatnie prawdopodobieństwo pozostania (wtedy kiedy $g + \epsilon f_i$ nie jest nieujemne). \square

Uwaga 2.3 *Diaconis, Eisenbud i Holmes [DEH97] używali bazy Markowa do generowania ruchów bezpośrednio na przestrzeni \mathcal{Y}_t . Wydaje się to użyteczne podejście dla problemu rzadkich tablic kontyngencji.*

2.2 Szybkość zbieżności

Prowadzone są intensywne prace mające na celu wyznaczenie ograniczeń na szybkość zbieżność opisanych wcześniej łańcuchów Markowa do swoich rozkładów stacjonarnych. Najwięcej wyników uzyskano dla przypadku dwuwymiarowych tablic kontyngencji oraz bazy Markowa typu $\begin{smallmatrix} + & - \\ - & + \end{smallmatrix}$. Poniżej pokażemy jeden z typowych wyników. Przez γ będziemy oznaczać średnicę grafu, którego wierzchołkami są elementy \mathcal{F}_t , a krawędź istnieje jeśli w bazie Markowa istnieje ruch łączący dwa elementy.

Twierdzenie 2.4 *Niech $m, n \in \mathbb{N} \setminus \{0\}$ oraz $r = (r_1, \dots, r_n)$ i $c = (c_1, \dots, c_m)$ takie, że $\sum c_i = \sum r_i = N$. Niech U będzie rozkładem jednostajnym na $\mathcal{F}(r, c)$, gdzie $\mathcal{F}(r, c)$ zdefiniowane jak wcześniej. Niech $P(f, f')$ będzie ruchem typu $\begin{smallmatrix} + & - \\ - & + \end{smallmatrix}$. Wówczas:*

$$\|P_f^k - U\|_{tv} \leq A_1 e^{-A_2 c} \quad \text{dla } k = c\gamma^2, c > 0,$$

gdzie f jest dowolnym stanem początkowym, A_1 i A_2 są stałymi, które zależą od r, c tylko poprzez m i n , oraz γ jest średnicą grafu – mamy ograniczenie $\gamma \leq \frac{N}{2}$.

W drugą stronę, istnieje $f \in \mathcal{F}(r, c)$ i stałe A_3, A_4 j.w. takie że:

$$\|P_f^k - U\|_{tv} \geq A_3 e^{-A_4 c} \quad \text{dla } k = c\gamma^2, c > 0,$$

Powyższe twierdzenie pokazuje zatem, że rząd γ^2 kroków są warunkiem koniecznym i dostatecznym do osiągnięcia stacjonarności. Stałe A_i rosną wykładniczo wraz z $d = m \cdot n$ i wynoszą około $(\frac{d}{4})^{\frac{d}{4}}$.

3 Algebraiczne metody szukania baz Markowa

W tej sekcji pokażemy jak w efektywny sposób szukać bazy Markowa dla danego modelu. Algorytm będzie bazował na stosunkowo prostej algebrze. Z góry we wszystkich niejasnych miejscach odwołujemy się do pierwszych trzech rozdziałów książki Cox, Little, O’Shea [CLO96].

3.1 Preliminaria algebraiczne

Podstawowym obiektem naszych badań będą ideały w pierścieniu wielomianów $k[x_1, \dots, x_n]$.

Definicja 3.1 *Podzbiór $I \subset k[x_1, \dots, x_n]$ nazywamy ideałem jeśli:*

- $0 \in I$.
- Jeżeli $f, g \in I$, to $f + g \in I$.
- Jeżeli $f \in I$ oraz $h \in k[x_1, \dots, x_n]$, wówczas $hf \in I$.

Kanonicznym przykładem ideału jest ideał generowany przez skończoną liczbę wielomianów:

Definicja 3.2 Niech f_1, \dots, f_s wielomiany w $k[x_1, \dots, x_n]$. Definiujemy:

$$\langle f_1, \dots, f_s \rangle = \left\{ \sum_{i=1}^s h_i f_i : h_1, \dots, h_s \in k[x_1, \dots, x_n] \right\}.$$

Kluczowe jest spostrzeżenie, że $\langle f_1, \dots, f_s \rangle$ istotnie jest ideałem. Ideał ten możemy interpretować jako zbiór wszystkich wielomianów spełniających układ równań $f_1 = 0, \dots, f_s = 0$.

Możliwość uporządkowania wszystkich jednomianów jest kluczową kwestią większości algorytmów algebry obliczeniowej, np. algorytmu dzielenia wielomianów wielu zmiennych. W przypadku pierścienia $k[x_1, \dots, x_n]$ zależnie od kontekstu stosuje się alternatywnie kilka różnych porządków. Poniższe rozumowanie wskazuje, jak takie porządki znajdować.

Zauważmy przede wszystkim, że możemy odtworzyć jednomian $x^\alpha = x_1^{\alpha_1} \cdots x_n^{\alpha_n}$ z wektora $\alpha = (\alpha_1, \dots, \alpha_n) \in \mathbb{Z}_{\geq 0}^n$. Ta obserwacja wprowadza 1 – 1 odpowiedniość pomiędzy jednomianami w $k[x_1, \dots, x_n]$ a elementami $\mathbb{Z}_{\geq 0}^n$. Zatem jeśli zdefiniujemy jakiś porządek na $\mathbb{Z}_{\geq 0}^n$ to zada on w sposób jednoznaczny porządek na jednomianach z $k[x_1, \dots, x_n]$: jeśli $\alpha > \beta$ to $x^\alpha > x^\beta$.

Definicja 3.3 Porządkiem na jednomianach z pierścienia $k[x_1, \dots, x_n]$ nazywamy każdą relację ' $>$ ' na $\mathbb{Z}_{\geq 0}^n$ (równoważnie każdą relację na zbiorze jednomianów x^α , $\alpha \in \mathbb{Z}_{\geq 0}^n$) spełniającą:

- (i) ' $>$ ' jest porządkiem liniowym na $\mathbb{Z}_{\geq 0}^n$.
- (ii) Jeśli $\alpha > \beta$ i $\gamma \in \mathbb{Z}_{\geq 0}^n$, to $\alpha + \gamma > \beta + \gamma$.
- (iii) Spełniona jest zasada minimum, tzn. każdy niepusty podzbiór $\mathbb{Z}_{\geq 0}^n$ ma najmniejszy element względem ' $>$ '.

Zaprezentujemy teraz najważniejsze przykłady.

Definicja 3.4 [Porządek leksykograficzny] Niech $\alpha = (\alpha_1, \dots, \alpha_n)$ i $\beta = (\beta_1, \dots, \beta_n) \in \mathbb{Z}_{\geq 0}^n$. Piszemy $\alpha >_{\text{lex}} \beta$, jeśli w wektorze $\alpha - \beta \in \mathbb{Z}^n$ pierwsza niezerowa współrzędna jest dodatnia. Będziemy pisać $x^\alpha >_{\text{lex}} x^\beta$ jeśli $\alpha >_{\text{lex}} \beta$.

I tak na przykład $(3, 2, 4) >_{\text{lex}} (3, 2, 1)$, bo $(3, 2, 4) - (3, 2, 1) = (0, 0, 3)$.

Należy jeszcze tylko dodać, że zmienne w $k[x_1, \dots, x_n]$ uporządkujemy w standardowy sposób, tzn.

$$(1, 0, \dots, 0) >_{\text{lex}} (0, 1, 0, \dots, 0) >_{\text{lex}} \dots >_{\text{lex}} (0, \dots, 0, 1),$$

a zatem $x_1 >_{\text{lex}} x_2 >_{\text{lex}} \dots >_{\text{lex}} x_n$. W przypadku trzech zmiennych mamy na przykład $x >_{\text{lex}} y^5 z^3$.

Porządek leksykograficzny jest analogiczny do uporządkowania słów w słownikach, tyle że zamiast liter i słów, mamy do czynienia z liczbami całkowitymi i wektorami. Litery porządkujemy tak, że $a >_{\text{lex}} b >_{\text{lex}} \dots >_{\text{lex}} \acute{z}$. Możemy zatem na przykład napisać $\text{pies} >_{\text{lex}} \text{piwo}$.

Twierdzenie 3.5 *Porządek leksykograficzny w $\mathbb{Z}_{\geq 0}^n$ jest porządkiem na jednomianach w $k[x_1, \dots, x_n]$.*

Podamy teraz serię definicji innych porządków na jednomianach w $k[x_1, \dots, x_n]$. Choć są one mniej intuicyjne, to okazuje się jednak, że w niektórych algorytmach ich wykorzystanie powoduje, że algorytmy są efektywniejsze niż przy zastosowaniu porządku leksykograficznego.

Definicja 3.6 *[Porządek leksykograficzny z gradacją] Niech $\alpha = (\alpha_1, \dots, \alpha_n)$ i $\beta = (\beta_1, \dots, \beta_n) \in \mathbb{Z}_{\geq 0}^n$. Piszemy $\alpha >_{\text{grlex}} \beta$, jeśli*

$$|\alpha| = \sum_{i=1}^n \alpha_i > |\beta| = \sum_{i=1}^n \beta_i, \quad \text{albo} \quad |\alpha| = |\beta| \quad \text{i} \quad \alpha >_{\text{lex}} \beta.$$

Definicja 3.7 *[Odwrotny porządek leksykograficzny z gradacją] Niech $\alpha = (\alpha_1, \dots, \alpha_n)$ i $\beta = (\beta_1, \dots, \beta_n) \in \mathbb{Z}_{\geq 0}^n$. Piszemy $\alpha >_{\text{grevlex}} \beta$, jeśli*

$$|\alpha| = \sum_{i=1}^n \alpha_i > |\beta| = \sum_{i=1}^n \beta_i, \quad \text{albo} \quad |\alpha| = |\beta|$$

oraz pierwszy **od prawej** niezerowy element jest ujemny.

Różnica między dwoma powyższymi porządkami jest subtelna. Mamy na przykład

$$x^5yz >_{\text{grlex}} x^4yz^2$$

gdyż oba jednomiany mają stopień równy 7 oraz $x^5yz >_{\text{lex}} x^4yz^2$. W tym konkretnym przypadku mamy również

$$x^5yz >_{\text{grevlex}} x^4yz^2$$

ale argumentacja jest inna: x^5yz jest większe gdyż mniejsza zmienna z występuje w tym jednomianie z mniejszą potęgą.

W dalszej części tego tekstu będziemy korzystać z następującej terminologii.

Definicja 3.8 *Niech $f = \sum_{\alpha} a_{\alpha} x^{\alpha}$ będzie niezerowym wielomianem w $k[x_1, \dots, x_n]$ i niech $>$ będzie porządkiem na jednomianach.*

(i) *Stopniem wielomianu f będziemy nazywać*

$$\deg(f) = \max(\alpha \in \mathbb{Z}_{\geq 0}^n : a_{\alpha} \neq 0),$$

gdzie maksimum jest brane względem $>$.

(ii) *Czynnikiem wiodącym wielomianu f jest ten spośród składników sumy definiującej wielomian f , w którym jest jednomian największy ze względu na zadany porządek.*

Niech na przykład $f = 4xy^2z + 4z^2 - 5x^3 + 7x^2z^2$ z porządkiem leksykograficznym. Wówczas

$$\deg(f) = (3, 0, 0)$$

$$\text{LT}(f) = -5x^3.$$

Lemat 3.9 Niech $f, g \in k[x_1, \dots, x_n]$ będą niezerowymi wielomianami, wówczas:

(i) $\deg(fg) = \deg(f) + \deg(g)$.

(ii) Jeżeli $f + g \neq 0$, wówczas $\deg(f + g) \leq \max(\deg(f), \deg(g))$.

Jeżeli dodatkowo $\deg(f) \neq \deg(g)$, to w (ii) mamy równość.

3.2 Bazy Markowa i ideały

Jak poprzednio \mathcal{X} niech będzie skończonym zbiorem i niech będzie dane $T : \mathcal{X} \rightarrow \mathbb{N}^d \setminus \{0\}$. Dla każdego $x \in \mathcal{X}$ wprowadźmy zmienną, którą też oznaczymy przez x (np. w tablicach kontyngencji każdej komórce tablicy odpowiada inna zmienna). Rozważmy pierścień wielomianów od tych zmiennych, ozn. $k[\mathbf{x}]$, gdzie k jest ciałem (np. liczb rzeczywistych). Funkcja $g : \mathcal{X} \rightarrow \mathbb{N}$ będzie reprezentowana jako jednomian $\prod_{x \in \mathcal{X}} x^{g(x)}$. Ten jednomian będziemy oznaczać \mathbf{x}^g . Funkcja $T : \mathcal{X} \rightarrow \mathbb{N}^d$ taka, że $x \mapsto (T(x)_1, \dots, T(x)_d)$ reprezentowana jest przez homomorfizm⁵

$$\phi_T : k[\mathbf{x}] \rightarrow k[t_1, \dots, t_d] \tag{3.1}$$

$$x \mapsto t_1^{T(x)_1} t_2^{T(x)_2} \dots t_d^{T(x)_d}, \tag{3.2}$$

gdzie $T(x)_i$ oznacza i -tą współrzędną $T(x) \in \mathbb{N}^d$. Zauważmy, że ϕ_T działa na jednomianach tak, że

$$\phi_T(\mathbf{x}^f) = \prod_{x \in \mathcal{X}} t_1^{f(x) \cdot T(x)_1} \dots t_d^{f(x) \cdot T(x)_d} = \prod_{i=1}^d t_i^{\sum_{x \in \mathcal{X}} f(x) \cdot T(x)_i}.$$

Podstawowym obiektem naszej analizy będzie jądro homomorfizmu ϕ_T , czyli ideał $\mathcal{I}_T = \{p \in k[\mathbf{x}] : \phi_T(p) = 0\}$.

Każdą funkcję $f : \mathcal{X} \rightarrow \mathbb{Z}$ można zapisać jako różnicę między swoją częścią dodatnią f^+ i ujemną f^- : $f^+(x) = \max(f(x), 0)$, $f^-(x) = \max(-f(x), 0)$.

Lemat 3.10 Zachodzi $\sum f(x)T(x) = 0$ wtedy i tylko wtedy, gdy dwumian $\mathbf{x}^{f^+} - \mathbf{x}^{f^-}$ należy do \mathcal{I}_T .

Dowód: Zachodzi $\sum f(x)T(x) = 0$ wtedy i tylko wtedy gdy $\sum f^+(x)T(x) = \sum f^-(x)T(x)$, czyli dla każdego $i = 1, \dots, d$ zachodzi $\sum f^+(x)T(x)_i = \sum f^-(x)T(x)_i$. Dalej:

$$\phi_T(\mathbf{x}^{f^+} - \mathbf{x}^{f^-}) = \prod_{i=1}^d t_i^{\sum_{x \in \mathcal{X}} f^+(x) \cdot T(x)_i} - \prod_{i=1}^d t_i^{\sum_{x \in \mathcal{X}} f^-(x) \cdot T(x)_i}.$$

A zatem mamy tezę. \square

Twierdzenie 3.11 Zbiór funkcji f_1, \dots, f_L jest bazą Markowa wtedy i tylko wtedy, gdy zbiór

$$\{\mathbf{x}^{f_i^+} - \mathbf{x}^{f_i^-} \mid 1 \leq i \leq L\}$$

generuje ideał \mathcal{I}_T .

⁵Uwaga, bo raz $x \in \mathcal{X}$ a innym razem $x \in k[\mathbf{x}]$.

Dowód: Dowód przebiega w dwóch krokach.

Krok 1: Niech \mathcal{I}' będzie ideałem generowanym przez dwumiany postaci

$$\mathbf{x}^{f^+} - \mathbf{x}^{f^-} \text{ dla } f \text{ takich, że } \sum f(x)T(x) = 0. \quad (3.3)$$

Pokażemy wprawdzie, że $\mathcal{I}' = \mathcal{I}_T$. Oczywiście $\mathcal{I}' \subseteq \mathcal{I}_T$, bo każdy generator \mathcal{I}' jest w \mathcal{I}_T na podstawie lematu 3.10. Aby udowodnić zawieranie w drugą stronę, ustalmy porządek na jednomianach jako porządek leksykograficzny z gradacją. Przeprowadźmy rozumowanie nie wprost. Przypuśćmy, że $\mathcal{I}' \subsetneq \mathcal{I}_T$ (tzn. inkluzja jest ścisła). Niech p będzie minimalny w sensie zadanego porządku wśród wielomianów z $\mathcal{I}_T \setminus \mathcal{I}'$ (czyli jego jednomian wiodący jest najmniejszy). Niech \mathbf{x}^α będzie największym jednomianem wielomianu p . Skoro $\phi_T(p) = 0$, to musi istnieć w p jakiś inny jednomian \mathbf{x}^β taki, że $\phi_T(\mathbf{x}^\alpha) = \phi_T(\mathbf{x}^\beta)$. Istotnie, $\phi_T(\mathbf{x}^\alpha)$ jest jednomianem, a zatem gdyby nie istniało odpowiednie \mathbf{x}^β to nie znalazł by się czynnik kasujący ten jednomian. Wydzielmy z obu jednomianów wspólny czynnik pisząc $\mathbf{x}^\alpha - \mathbf{x}^\beta = \mathbf{x}^\gamma(\mathbf{x}^{\alpha'} - \mathbf{x}^{\beta'})$ tak, że α' oraz β' mają rozłączne nośniki (tzn. mają niezerowe elementy w różnych miejscach). Skoro $\phi_T(\mathbf{x}^\gamma) \neq 0$ (bo dla każdego $x \in \mathcal{X}$ $T(x) \neq 0$) to jasne jest, że musi być spełnione $\phi_T(\mathbf{x}^{\alpha'}) = \phi_T(\mathbf{x}^{\beta'})$. Ustalmy $h = \alpha' - \beta'$, z lematu 3.10 mamy wówczas $\sum h(x)T(x) = 0$, a zatem $\mathbf{x}^\alpha - \mathbf{x}^\beta = \mathbf{x}^\gamma(\mathbf{x}^{h^+} - \mathbf{x}^{h^-}) \in \mathcal{I}'$. Odejmując wielokrotność dwumianu $\mathbf{x}^\alpha - \mathbf{x}^\beta$ od p , dostajemy wielomian o mniejszym jednomianie wiodącym. Ale na mocy założenia ten już siedzi w \mathcal{I}' . A zatem $\mathcal{I}' = \mathcal{I}_T$.

Krok 2: Następnie niech $\mathcal{B} = \{\mathbf{x}^{f_i^+} - \mathbf{x}^{f_i^-} \mid 1 \leq i \leq L\}$. Własność 2.5 jest równoważna z tym, że $\mathcal{B} \subset \mathcal{I}_T$. Musimy zatem pokazać, że 2.6 jest spełnione wtedy i tylko wtedy, gdy \mathcal{B} generuje \mathcal{I}_T .

\implies Przypuśćmy, że spełnione jest 2.6. Na mocy lematu oraz pierwszego kroku dowodu wystarczy pokazać, że każda $f : \mathcal{X} \rightarrow \mathbb{Z}$ t.je $\sum f(x)T(x) = 0$, dwumian $\mathbf{x}^{f^+} - \mathbf{x}^{f^-}$ leży w ideale generowanym przez \mathcal{B} (tzn. da się przedstawić jako wielomianowa kombinacja dwumianów z \mathcal{B}). Z 2.6 mamy ciąg (f_{i_j}) elementów z \mathcal{B} taki, że $f^+ + \sum_{j=1}^K \epsilon_j f_{i_j} = f^-$ dla $\epsilon_j = \pm 1$ taki, że po drodze dla każdego $1 \leq k \leq K$ zachodzi $f^+ + \sum_{j=1}^k \epsilon_j f_{i_j} \geq 0$. Jeśli $K = 1$ oraz $\epsilon_1 = 1$ wówczas $f^- = f^+ + f_{i_1}$ albo inaczej $f^- - f^+ = f_{i_1}^+ - f_{i_1}^-$. Części ujemne i dodatnie obu wektorów mają rozłączne nośniki, a zatem $f^- = f_{i_1}^+$ oraz $f^+ = f_{i_1}^-$, a stąd

$$\mathbf{x}^{f^+} - \mathbf{x}^{f^-} = -(\mathbf{x}^{f_{i_1}^+} - \mathbf{x}^{f_{i_1}^-}) \in \langle \mathcal{B} \rangle$$

Podobną argumentację możemy przeprowadzić dla $\epsilon_1 = -1$.

Dowód dla $K > 1$ przeprowadzimy indukcyjnie. Przypuśćmy, że implikacja w prawą stronę jest prawdziwa dla $K - 1$. Z drugiego warunku definicji 2.1 wiemy, że $f^+ + \sum_{j=1}^k \epsilon_j f_{i_j} \geq 0$ oraz $f^- - \sum_{j=k}^{K-k+1} \epsilon_j f_{i_j} \geq 0$ dla każdego $1 \leq k \leq K$. W szczególności mamy $f^- - \epsilon_K f_{i_K} \geq 0$ oraz $f^+ + \epsilon_1 f_{i_1} \geq 0$. Korzystając z części dla $K = 1$ wiemy, że dwumiany $\mathbf{x}^{f^+} - \mathbf{x}^{f^+ + \epsilon_1 f_{i_1}}$ oraz $\mathbf{x}^{f^- - \epsilon_K f_{i_K}} - \mathbf{x}^{f^-}$ należą do $\langle \mathcal{B} \rangle$. Skoro tak, to w ideale siedzi również suma tych dwumianów (z definicji ideału):

$$(\mathbf{x}^{f^+} - \mathbf{x}^{f^+ + \epsilon_1 f_{i_1}}) + (\mathbf{x}^{f^- - \epsilon_K f_{i_K}} - \mathbf{x}^{f^-}) \in \langle \mathcal{B} \rangle,$$

gdzie drugi nawias siedzi w $\langle \mathcal{B} \rangle$ na mocy indukcji, a zatem $\mathbf{x}^{f^+} - \mathbf{x}^{f^-} \in \langle \mathcal{B} \rangle$. Zatem zbiór generatorów ideału \mathcal{I}_T siedzi w ideale generowanym przez \mathcal{B} , a z tego $\mathcal{I}_T \subseteq \langle \mathcal{B} \rangle$. Co dowodzi, że zachodzi równość między tymi ideałami.

\Leftarrow Przypuśćmy, że \mathcal{B} generuje I_T . Dla g, g' takich, że $\sum(g(x) - g'(x))T(x) = 0$ (wtedy $\mathbf{x}^g - \mathbf{x}^{g'} \in \mathcal{I}_T$) istnieje przedstawienie

$$\mathbf{x}^g - \mathbf{x}^{g'} = \sum_{j=1}^m \epsilon_j \mathbf{x}^{h_j} (\mathbf{x}^{f_{i_j}^+} - \mathbf{x}^{f_{i_j}^-}). \quad (3.4)$$

Możemy założyć, że $\epsilon_j = \pm 1$, gdyż w innym przypadku wystarczy kilka razy dodać ten sam element. Jeśli $m = 1$ oraz $\epsilon_1 = 1$, reprezentacja 3.4 pokazuje, że spełnione są oba warunki z definicji bazy Markowa, gdyż $g - f_{i_1}^+ = h_1 = g' - f_{i_1}^-$, a zatem $g' + f_{i_1} = g$ i g komunikuje się z g' . Podobna argumentacja dla $\epsilon_1 = -1$.

Niech $m > 1$ i załóżmy, że dla $m-1$ reprezentacja 3.4 implikuje spełnienie warunków bycia bazą Markowa. Z równania 3.4 widać, że $\mathbf{x}^g = \mathbf{x}^{h_r} \mathbf{x}^{f_{i_r}^\pm}$ dla pewnego r – załóżmy, że $\mathbf{x}^g = \mathbf{x}^{h_r} \mathbf{x}^{f_{i_r}^-}$. Wówczas $h_r = g - f_{i_r}^-$ jest nieujemny (bo h_r jest wykładnikiem potęgi), a zatem $g + f_{i_r}$ również jest nieujemny. Dodając $\mathbf{x}^{h_r} (\mathbf{x}^{f_{i_r}^+} - \mathbf{x}^{f_{i_r}^-})$ po obu stronach równania 3.4 oraz wykorzystując, że $h_r + f_{i_r}^+ = g + f_{i_r}$ dostajemy, że

$$\mathbf{x}^{g+f_{i_r}} - \mathbf{x}^{g'} = \sum_{j \neq r} \epsilon_j \mathbf{x}^{h_j} (\mathbf{x}^{f_{i_j}^+} - \mathbf{x}^{f_{i_j}^-})$$

ale powyższa suma ma $m-1$ elementów i możemy skorzystać z założenia indukcyjnego. Zatem $g + f_{i_r}$ może być połączony z g' ciągiem dozwolonych ruchów, a zatem \mathcal{B} jest bazą Markowa. \square

Powyższe twierdzenie jest fundamentalne dla tej teorii, bo łączy w elegancki sposób obiekt algebraiczny z obiektem statystycznym. Redukuje ono problem szukania bazy Markowa do znalezienia zbioru generatorów dla ideału $\mathcal{I}_T \in k[\mathbf{x}]$. Jego waga podparta jest poniższą uwagą.

Uwaga 3.12 *Klasyczne w geometrii algebraicznej twierdzenie Hilberta o bazie mówi, że każdy ideał w pierścieniu wielomianów jest skończenie generowany. Zatem dla każdego ideału \mathcal{I}_T znajdziemy skończony zbiór generujących go dwumianów.*

Uwaga 3.13 *Założyliśmy, że T ma wartości w \mathbb{N}^d . Twierdzenie jest również prawdziwe dla $T : \mathcal{X} \rightarrow \mathbb{Z}^d$. Wówczas ϕ_T przekształca do pierścienia $k[t_1, \dots, t_d, t_1^{-1}, \dots, t_d^{-1}]$.*

4 Wykorzystanie komputera do znajdowania generatorów ideału \mathcal{I}_T

Siła algorytmu Diaconisa-Sturmfelsa tkwi w tym, że obliczenia wielomianowe są niezwykle szybkie. Można je przeprowadzać praktycznie na dowolnym systemie do obliczeń symbolicznych, jak **Maple** czy darmowa **Maxima**. Warto jednak podkreślić, że istnieje kilka aplikacji dedykowanych do obliczeń wielomianowych, które działają znacznie szybciej. Poniżej postaramy się zaprezentować wygodny sposób na przeprowadzenie obliczeń dla algorytmu opisanego w tym dokumencie.

4.1 Dwa sposoby opisu ideału \mathcal{I}_T

W pierwszej kolejności pokażemy jak opisać ideał \mathcal{I}_T tak aby dane wejściowe mogły zostać odczytane przez takie programy jak Maple, Mathematica, CoCoA, 4ti2 lub Macaulay2.

Na początku musimy się zdecydować na jakiś porządek wielomianowy – niech to będzie porządek leksykograficzny z gradacją. Dzięki temu dla każdego wielomianu p w sposób jednoznaczny potrafimy wyznaczyć jego czynnik wiodący $\text{LT}(p)$.

Niech \mathcal{I} będzie ideałem w pierścieniu $k[\mathbf{x}]$ z ustalonym porządkiem. Bazą Gröbnera dla \mathcal{I} jest zbiór wielomianów $\{p_1, \dots, p_L\} \subset \mathcal{I}$ taki, że ideał generowany przez

$$\{\text{LT}(p_1), \dots, \text{LT}(p_L)\}$$

równy jest ideałowi generowanemu przez

$$\{\text{LT}(p) \mid p \in \mathcal{I}\},$$

który oznacza się zwyczajowo przez $\text{LT}(\mathcal{I})$. Okazuje się, że baza Gröbnera generuje \mathcal{I} (patrz Cox i in. [CLO96]) i istnieje efektywny algorytm na jej znajdowanie zaimplementowany w większości systemów do obliczeń symbolicznych. Baza Gröbnera jest *minimalna* jeśli żaden z jej wielomianów nie może zostać skasowany, a *zredukowana* jeśli dla każdej pary i, j żaden z czynników wielomianu p_i nie jest podzielny przez $\text{LT}(p_j)$. Przy ustalonym porządku wielomianowym mamy jednoznacznie wyznaczoną zredukowaną bazę Gröbnera. Poniższe twierdzenie daje łatwy sposób na znajdowanie zredukowanej bazy Gröbnera dla ideału \mathcal{I}_T ⁶.

Twierdzenie 4.1 *Niech \mathcal{X} będzie zbiorem skończonym. Niech dana będzie $T : \mathcal{X} \rightarrow \mathbb{N}^d$. Niech $\mathbf{t} = (t_1, \dots, t_d)$. Przy ustalonym porządku na zmiennych \mathbf{x} rozszerzmy go do porządku na $\mathbf{x} \cup \mathbf{t}$ biorąc $t_1 > t_2 > \dots > t_d > x$ dla wszystkich $x \in \mathbf{x}$. Niech $\mathcal{J}_T = \{x - \mathbf{t}^{T(x)} \mid x \in \mathbf{x}\}$. Wówczas $\mathcal{I}_T = \mathcal{J}_T \cap k[\mathbf{x}]$ a zredukowaną bazę Gröbnera dla \mathcal{I}_T można znaleźć biorąc te tylko elementy bazy dla \mathcal{J}_T , które zawierają jedynie elementy pierścienia $k[\mathbf{x}]$.*

Dowód: Jest bezpośrednią aplikacją twierdzenia o eliminacji (patrz Cox i in. [CLO96], s. 114, 128).

Nowsze prace z algebry torycznej modeli graficznych (patrz np. [GMS02]) korzystają z reprezentacji przekształcenia $T : \mathcal{X} \rightarrow \mathbb{N}^d$ w postaci macierzy. T jest zadane na bazie $x \in \mathcal{X}$ zatem jej kolumny odpowiadają obrazom kolejnych elementów zbioru \mathcal{X} . Często elementy tego zbioru utożsamiamy z wektorami jednostkowymi przestrzeni $\mathbb{R}^{|\mathcal{X}|}$. Zapis modelu w postaci macierzy wydaje się prostszy niż pracowite wypisywanie wszystkich dwumianów, jak w twierdzeniu 4.1. Szczęśliwie kilka programów obliczeniowych umożliwia skorzystanie z reprezentacji macierzowej, co zostanie opisane poniżej. Wpierw jednak zapiszemy powyższe rozważania dla przykładu tablic kontyngencji.

⁶W ogólności interesuje nas dowolny zbiór generatorów tego ideału, jednak bazy Gröbnera mają mnóstwo użytecznych właściwości, a obliczenia z ich użyciem są szczególnie efektywne.

Przykład 4.2 (Model niezależności dla tablic kontyngencji) W przypadku tablic kontyngencji będziemy zmienne pierścienia $k[\mathbf{x}]$ indeksować nie liczbami od 1 do $|\mathcal{X}|$ lecz w sposób naturalny komórkami tablic. Dla dwóch wymiarów (tablica wymiaru $m \times n$) dostajemy zatem $m \cdot n$ zmiennych postaci x_{ij} . Statystyka dostateczna wylicza rozkłady brzegowe, czyli $f \mapsto (f_{1+}, \dots, f_{m+}, f_{+1}, \dots, f_{+n})$. Zatem na bazie $T : \mathbb{N}^{m \cdot n} = \mathbb{N}^m \otimes \mathbb{N}^n \rightarrow \mathbb{N}^{m+n} = \mathbb{N}^m \oplus \mathbb{N}^n$ działa tak, że $e'_{ij} = e_i \otimes e_j \rightarrow e_i \oplus e_j$, czyli wektor jednostkowy długości mn przechodzi na wektor długości $m+n$, w którym są same zera oraz dwie jedynki na miejscu i -tym i $(m+j)$ -tym. Indukowany homomorfizm pierścieni to:

$$\begin{aligned} \phi_T : \quad k[x_{11}, \dots, x_{mn}] &\longrightarrow k[s_1, \dots, s_m, t_1, \dots, t_n] \\ x_{ij} &\mapsto s_i \cdot t_j. \end{aligned}$$

Przypadek tablic dwuwymiarowych jest, jak się okazuje, bardzo prosty. Mamy twierdzenie, które mówi, że niezależnie od założonego porządku wielomianowego baza Gröbnera dla tego modelu to:

$$G_T = \{x_{il}x_{jk} - x_{ik}x_{jl} \mid 1 \leq i < l \leq m, 1 \leq k < j \leq n\},$$

co odpowiada wszystkim minorom 2×2 macierzy (x_{ij}) .

4.2 Obliczenia w CoCoA

CoCoA (<http://cocoa.dima.unige.it/>) jest darmowym i bardzo popularnym programem do obliczeń w algebrze przemiennej. Znajdowanie generatorów dowolnego ideału w pierścieniu wielomianów przeprowadzane jest w bardzo efektywny sposób. Co ważniejsze - aplikacja ta posiada również specjalną funkcję `Toric`, która jest dedykowana do obliczania generatorów ideału torycznego⁷ (również z postaci macierzowej).

Rozpoczynamy od zdefiniowania pierścienia, w którym będą przeprowadzane obliczenia. Domyślnym porządkiem w programie CoCoA jest porządek odwrotny leksyko-graficzny z gradacją. W tym przykładzie zajmiemy się tablicami kontyngencji wymiaru 3×3 , gdzie T jest wektorem rozkładów brzegowych, dlatego mamy 9 zmiennych indeksowanych elementami macierzy, czyli x_{ij} dla $1 \leq i, j \leq 3$ oraz 6 zmiennych które zadają przeciwdziedzinę homomorfizmu ϕ_T , które będziemy oznaczać $s_1, s_2, s_3, t_1, t_2, t_3$. Tak jak już zostało to opisane wcześniej, ideał toryczny możemy obliczać albo korzystając z jednomianów będących obrazami zmiennych x_{ij} przy homomorfizmie ϕ_T (wymaga to rozpatrywania pierścienia o większej liczbie zmiennych) lub poprzez podanie macierzy przekształcenia T .

Przypadek zadany przez jednomiany W pierwszym przypadku wpierw definiujemy pierścień z rozszerzoną liczbą zmiennych (nad ciałem liczb wymiernych)

```
Use R := Q[x[1..3], s[1..3], t[1..3]];
```

⁷Ideał \mathcal{I}_T jest przykładem ideału torycznego. Nazwa wzięła się z geometrii algebraicznej, gdzie ideały toryczne definiują rozmaitości toryczne (patrz Sturmfels [Stu96]).

Homomorfizm ϕ_T przekształca $x_{ij} \mapsto s_i t_j$. Okazuje się (patrz twierdzenie 4.1), że generatory ideału torycznego uzyskujemy licząc bazę Gröbnera ideału w rozszerzonym pierścieniu generowanego przez dwumianu postaci $x_{ij} - s_i t_j$. W programie wpisujemy:

```
Set Indentation;
I:=Ideal(x[1,1]-s[1]*t[1],x[1,2]-s[1]*t[2],x[1,3]-s[1]*t[3],
        x[2,1]-s[2]*t[1],x[2,2]-s[2]*t[2],x[2,3]-s[2]*t[3],
        x[3,1]-s[3]*t[1],x[3,2]-s[3]*t[2],x[3,3]-s[3]*t[3]);
G:=GBasis(I);
K:=Elim(s,Ideal(G));
Elim(t,Ideal(K));
```

Pierwsza linijka służy formatowaniu wyniku. Druga definiuje interesujący nas ideał poprzez podanie jego generatorów. Następnie polecenie `GBasis` oblicza bazę Gröbnera \mathcal{G} tego ideału. Twierdzenie 4.1 mówi, że szukane generatory ideału torycznego \mathcal{I}_T da się odzyskać licząc przecięcie $\mathcal{G} \cap k[\mathbf{x}]$, co realizują dwie ostatnie linijki kodu. Dostajemy wynik:

```
Ideal(
  -x[2,3]x[3,2] + x[2,2]x[3,3],
  -x[1,3]x[3,2] + x[1,2]x[3,3],
  -x[1,3]x[2,2] + x[1,2]x[2,3],
  -x[2,3]x[3,1] + x[2,1]x[3,3],
  -x[2,2]x[3,1] + x[2,1]x[3,2],
  -x[1,3]x[3,1] + x[1,1]x[3,3],
  -x[1,2]x[3,1] + x[1,1]x[3,2],
  -x[1,3]x[2,1] + x[1,1]x[2,3],
  -x[1,2]x[2,1] + x[1,1]x[2,2])
```

Z powyższych generatorów odzyskujemy bazę Markowa rozpatrując macierze 3×3 , gdzie kładziemy 1 w miejscu (i, k) i (j, l) oraz -1 w miejscu (j, k) oraz (i, l) , jeśli odpowiadającym generatorem jest $x_{ik}x_{jl} - x_{il}x_{jk}$ (istotnie wszystkie są tej postaci).

Generatory te odpowiadają dokładnie dziewięciu prostym ruchom postaci $\begin{matrix} + & - \\ - & + \end{matrix}$.

Przypadek zadany przez macierz przekształcenia T Sprawa jest dużo prostsza jeśli dysponujemy macierzą przekształcenia T . Ogólnie pisaliśmy, że zmienne x utożsamiamy z elementami $x \in \mathcal{X}$. W przypadku macierzy kontyngencji wymiaru $m \times n$ możemy rozpatrywać zmienne x_{ij} . I tak na przykład element zbioru \mathcal{X} , któremu odpowiada zmienna x_{11} daje w obrazie przekształcenia T wektor $[1, 0, \dots, 0; 1, 0, \dots, 0]$. Ogólnie element x_{ij} daje wektor zer z dwoma jedynkami: na i -tym i na $(j + m)$ -tym miejscu (T na bazie oblicza rozkłady brzegowe macierzy typu E_{ij}). Macierz przekształcenia T jest zbudowana z kolumn będących obrazami bazy x_{ij} . W programie `CoCoA` da się bardzo łatwo policzyć generatory takiego modelu za pomocą polecenia `Toric`:

```

Use R:=Q[x[1..3,1..3]];
Set Indentation;
T:=Mat([[1,0,0,1,0,0],[1,0,0,0,1,0],[1,0,0,0,0,1],[0,1,0,1,0,0],
        [0,1,0,0,1,0],[0,1,0,0,0,1],[0,0,1,1,0,0],[0,0,1,0,1,0],[0,0,1,0,0,1]]);
T:=Transposed(T);
Toric(T);

```

Jako wynik dostajemy te same dziewięć generatorów, co poprzednio. Jest to zatem wygodniejsza i szybsza metoda przeprowadzenia tych samych obliczeń.

4.3 Obliczenia w 4ti2

W programie 4ti2 (<http://www.4ti2.de/>) w sposób niezwykle wygodny możemy uzyskać macierz statystyki dostatecznej T (wcześniej musieliśmy ją wpisywać ręcznie). W istocie nasz model może być dowolnym modelem hierarchicznym dla n zmiennych dyskretnych. Model taki zadany jest przez kompleks symplecjalny⁸ na zbiorze $\{1, \dots, n\}$ lub równoważnie przez generatory tego kompleksu, czyli sympleksy maksymalnego wymiaru. Szczególnym przypadkiem modeli hierarchicznych są modele graficzne, w których generatorami kompleksu są maksymalne kliki grafu. Zwyczajowe odnośniki do literatury to artykuł Geiger, Meek i Sturmfels [GMS02] (kontekst algebry torycznej) oraz monografia Lauritzen [Lau96] (kontekst statystyczny).

W przypadku modeli pełnej niezależności (wszystkie zmienne są wzajemnie niezależne) odpowiadający graf składa się z izolowanych wierzchołków, a zatem generatorami kompleksu są zbiory jednoelementowe. Na przykład dla modelu dwóch zmiennych (każda z trzema poziomami, jak w przykładzie dla programu CoCoA) generujemy plik `model3ind.mod` wpisując:

```

2
3 3
2
1 2
1 1

```

Pierwsza linijka to liczba zmiennych. Druga linijka to kolejno ich poziomy. Trzecia linijka to liczba generatorów sympleksu. Dalej kolejno zostały wymienione te generatory, przy czym pierwszy spis oznacza liczbę wierzchołków w generatorze a dalsze to lista tych generatorów. Wpisując w terminalu

```
> genModel model3ind
```

jako wynik zapisany do pliku `model3ind` dostajemy macierz:

```

6 9
1 1 1 0 0 0 0 0 0

```

⁸Kompleks symplecjalny K spełnia dwa warunki: (i) Każdy podsympleks sympleksu leżącego w kompleksie K leży w K . (ii) Przecięcie dolnych dwóch sympleksów w K leży w K .

```

0 0 0 1 1 1 0 0 0
0 0 0 0 0 0 1 1 1
1 0 0 1 0 0 1 0 0
0 1 0 0 1 0 0 1 0
0 0 1 0 0 1 0 0 1

```

Aby znaleźć bazę Markowa dla tego modelu, używamy polecenia `markov`. Wpisując

```
> markov model3ind
```

w pliku `model3ind.mar` dostajemy jako wynik zestawienie generatorów tego modelu, które u nas tak jak poprzednio odpowiadają dziewięciu podstawowym ruchom:

```

9 9
-1 0 1 0 0 0 1 0 -1
-1 0 1 1 0 -1 0 0 0
-1 1 0 0 0 0 1 -1 0
-1 1 0 1 -1 0 0 0 0
0 -1 1 0 0 0 0 1 -1
0 -1 1 0 1 -1 0 0 0
0 0 0 -1 0 1 1 0 -1
0 0 0 -1 1 0 1 -1 0
0 0 0 0 -1 1 0 1 -1

```

Co jednak istotne, duża elastyczność specyfikacji modelu pozwala nam badać dużo bardziej złożone hipotezy. Na przykład jeśli badamy trzy zmienne (wszystkie o trzech poziomach) i badamy tylko czy istnieje łączna zależność wszystkich zmiennych, to modelowi alternatywnemu (w którym maksymalne zależności są dwuelementowe) odpowiada pełny graf o trzech wierzchołkach.

Odpowiedni plik wygląda jak poniżej:

```

3
3 3 3
3
2 1 2
2 2 3
2 3 1

```

Zbiór generatorów ideału \mathcal{I}_T dla tego przypadku jest 81-elementowy. Mamy 27 generatorów typu:

$$\begin{array}{ccc}
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0
\end{array}
\quad
\begin{array}{ccc}
0 & 0 & 0 \\
0 & + & - \\
0 & - & +
\end{array}
\quad
\begin{array}{ccc}
0 & 0 & 0 \\
0 & - & + \\
0 & + & -
\end{array}
\tag{4.1}$$

czyli jak w przypadku dwuwymiarowym. Te ruchy jednak nie dają łańcucha nieprzywiedlnego. Mamy dodatkowo 54 ruchy stopnia 6 postaci:

$$\begin{array}{ccc}
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0
\end{array}
\quad
\begin{array}{ccc}
0 & - & + \\
+ & 0 & - \\
- & + & 0
\end{array}
\quad
\begin{array}{ccc}
0 & + & - \\
- & 0 & + \\
+ & - & 0
\end{array}
\tag{4.2}$$

Obliczenia te wykorzystamy w sekcji poświęconej przykładom wykorzystania opisowego algorytmu.

5 Przykłady

5.1 Trójwymiarowe tablice kontyngencji

Rozważania tej sekcji zaczerpnięte są z wykładów Steffena L. Lauritzena [Lau02], które są dostępne w Sieci.

5.1.1 Modele tablic kontyngencji

W przypadku dwuwymiarowych tablic kontyngencji przez p_{ij} oznaczaliśmy prawdopodobieństwo, że obiekt charakteryzuje się i -tą cechą pierwszej zmiennej oraz j -tą cechą drugiej zmiennej. W takim przypadku całkowite prawdopodobieństwo tablicy możemy zapisać jako:

$$P(f = (f_{ij})) = \frac{N!}{\prod_{i,j} f_{ij}!} \prod_{i,j} p_{ij}^{f_{ij}}.$$

Modele dla dwuwymiarowych tablic kontyngencji zadawane są przez poniższe restrykcyjne na p_{ij} (domyślnie zachodzi $p_{ij} \geq 0$ i $\sum_{i,j} p_{ij} = 1$):

M_0 : p_{ij} bez dodatkowych restrykcyjne

M_1 : niezależność: $p_{ij} = p_{i+}p_{+j}$

M_2 : jednostajność względem pierwszej zmiennej $p_{ij} = \frac{1}{r}p_{+j}$

M_2^* : jednostajność względem drugiej zmiennej $p_{ij} = p_{i+} \frac{1}{s}$

M_3 : całkowita jednostajność $p_{ij} = \frac{1}{r} \frac{1}{s}$

Modele te oczywiście związane są poprzez następujące relacje zawierania:

$$M_0 \supset M_1 \supset \begin{matrix} M_2 \\ M_2^* \end{matrix} \supset M_3.$$

Oszacowania największej wiarygodności prawdopodobieństw p_{ij} dla poszczególnych modeli wynoszą:

$$M_0: \hat{p}_{ij} = \frac{f_{ij}}{N}$$

$$M_1: \hat{p}_{ij} = \frac{f_{i+}}{N} \cdot \frac{f_{+j}}{N}$$

$$M_2: \hat{p}_{ij} = \frac{1}{r} \cdot \frac{f_{+j}}{N}$$

$$M_2^*: \hat{p}_{ij} = \frac{f_{i+}}{N} \cdot \frac{1}{s}$$

$$M_3: \hat{p}_{ij} = \frac{1}{r \cdot s}$$

gdzie f_{i+} oraz f_{+j} oznaczają odpowiedni element wektora rozkładów brzegowych. Modele mogą być testowane za pomocą statystyki ilorazu wiarygodności danej przez:

$$-2 \log Q = 2 \sum_{i,j} f_{ij} (\log \hat{p}_{ij} - \log \hat{\hat{p}}_{ij}),$$

gdzie \hat{p}_{ij} oznacza oszacowania p_{ij} za pomocą założonego modelu, a $\hat{\hat{p}}_{ij}$ odpowiada testowanemu modelowi, będącemu podmodelem wyjściowego.

Statystykę ilorazu wiarygodności można przybliżać statystyką χ^2 Pearsona:

$$-2 \log Q \approx \chi^2 = \sum_{i,j} \frac{(N \cdot \hat{p}_{ij} - N \hat{\hat{p}}_{ij})^2}{N \cdot \hat{\hat{p}}_{ij}}.$$

Dokładny rozkład obu powyższych statystyk w ogólnym przypadku jest nieznanym, dlatego wnioskowanie opiera się w praktyce na rozkładzie asymptotycznym, którym jest rozkład χ^2 z odpowiednią liczbą stopni swobody.

Większość rozważań dwuwymiarowych rozszerza się na przypadek tablic trójwymiarowych. Jeśli przez f_{ijk} oznaczymy obserwacje poszczególnych komórek tablicy a p_{ijk} ich prawdopodobieństwa, możemy jak poprzednio rozważać modele typu:

- niezależność pomiędzy (X, Y) a Z : $p_{ijk} = p_{ij+} \cdot p_{++k}$
- jednostajność ze względu na Z : $p_{ijk} = p_{ij+} \cdot \frac{1}{t}$
- całkowita niezależność: $p_{ijk} = p_{i++} \cdot p_{+j+} \cdot p_{++k}$

Ale przypadek trójwymiarowy przynosi także nowe możliwości formułowania modeli. Naturalną hipotezą dotyczącą trzech zmiennych może być na przykład, że X i Z są warunkowo niezależne względem zmiennej Y :

$$p_{ik|j} = p_{i|j} \cdot p_{k|j}$$

lub równoważnie

$$p_{ijk} = \frac{p_{ij+} \cdot p_{+jk}}{p_{+j+}}.$$

Jeżeli popatrzymy na trójwymiarową tablicę kontyngencji jak na kostkę to warunek $Y = j$ wycina plasterki z tej kostki, który jest dwuwymiarową tablicą dla X i Z . Powyższa hipoteza warunkowej niezależności oznacza w istocie, że każdy z s takich plasterków odpowiada niezależności zmiennych X i Z . Oszacowanie największej wiarygodności dla tego modelu wynosi:

$$\hat{p}_{ijk} = \frac{f_{ij+} \cdot f_{+jk}}{f_{+j+} \cdot N}$$

Jak w przypadku dwuwymiarowym, wnioskowanie może być dokonywane na podstawie:

$$-2 \log Q = 2 \sum_{i,j,k} f_{ijk} (\log \hat{p}_{ijk} - \log \hat{\hat{p}}_{ijk}) \approx \sum_{i,j,k} \frac{(N \cdot \hat{p}_{ijk} - N \cdot \hat{\hat{p}}_{ijk})^2}{N \hat{\hat{p}}_{ijk}}.$$

Standardowo teorię modelowania trójwymiarowych tablic kontyngencji opiera się na równaniu typu:

$$\log p_{ijk} = \alpha + \beta_i^1 + \beta_j^2 + \beta_k^3 + \gamma_{ij}^{12} + \gamma_{jk}^{23} + \gamma_{ik}^{13} + \delta_{ijk}^{123}$$

i nakładaniu restrykcji zerowych na pewne z powyższych parametrów. Na przykład modelowi warunkowej niezależności odpowiada założeniu $\delta_{ijk}^{123} \equiv 0$ oraz $\gamma_{ik}^{13} \equiv 0$. Modele tego typu nazywa się modelami logliniowymi, a czynniki pojawiające się w modelu nazywa się interakcjami. Jak widać modele logliniowe łatwo uogólniają się do przypadku n -wymiarowego. Słabością tych modeli jest jednak trudność w interpretacji wyników.

W praktyce wygodniej jest ograniczać się do modeli hierarchicznych. Modele tego typu zadane są przez zbiór \mathcal{C} podzbiorów zbioru $\Gamma = \{1, \dots, n\}$ (zbioru indeksów zmiennych w analizowanym systemie) takich, że żaden nie jest podzbiorem innego. Model hierarchiczny zatem to rodzina rozkładów prawdopodobieństw taka, że:

$$p \in \mathcal{P}_{\mathcal{C}} \iff \log p(i) = \sum_{a \subseteq \Gamma} \phi_a(i),$$

gdzie $\phi_a \equiv 0$ o ile $a \not\subseteq c$ dla jakiegoś $c \in \mathcal{C}$. Zatem \mathcal{C} jest zbiorem maksymalnych możliwych interakcji.

Dla przypadku n zmiennych niech $I = [m_1] \times \dots \times [m_n]$ oznacza zbiór indeksów n -wymiarowej tablicy kontyngencji. Dla $i = (i_1, \dots, i_n)$ przez $f(i)$ będziemy oznaczać i -ty element tablicy f . Statystyką dostateczną dla modelu hierarchicznego jest $\{f(i_c)\}$ takie, że $c \in \mathcal{C}$, czyli suma po wszystkich $\{f(i)\}$ takich, że na współrzędnych odpowiadających c indeksy zgadzają się i są równe i_c . Ta sytuacja podlega również pod ogólny kontekst zarysowany w tym opracowaniu.

5.1.2 Model bez interakcji trójstronnych.

W przypadku tablic trójwymiarowych mamy $\Gamma = \{1, 2, 3\}$ i możemy rozważać model bez interakcji trójstronnych, którego zbiorem generatorów jest $\mathcal{C} = \{\{1, 2\}, \{2, 3\}, \{1, 3\}\}$. Równoważnie model ten jest zadany przez zbiór ilorazów postaci:

$$\frac{p_{111} \cdot p_{ij1}}{p_{i11} \cdot p_{1j1}} = \frac{p_{11k} \cdot p_{ijk}}{p_{i1k} \cdot p_{1jk}} \quad 2 \leq i \leq r, 2 \leq j \leq s, 2 \leq k \leq t. \quad (5.1)$$

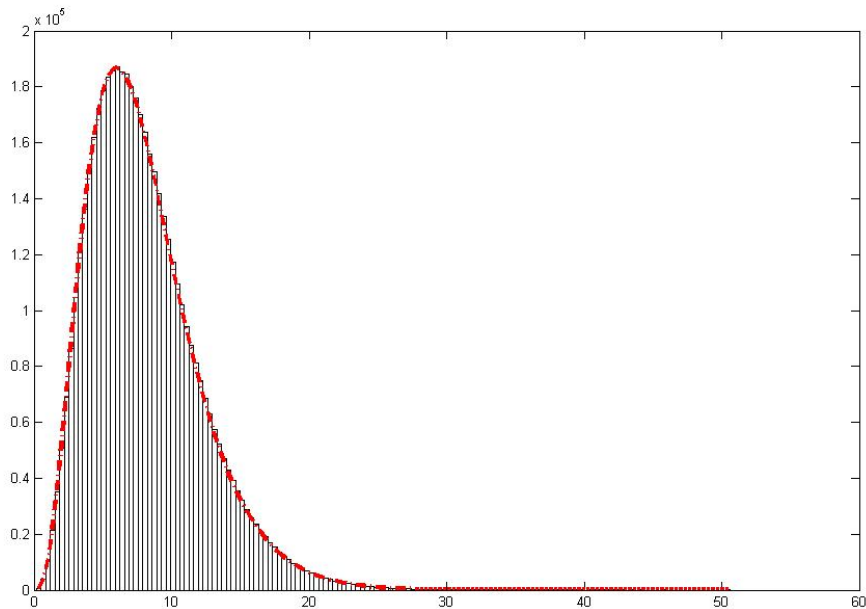
Statystykami dostatecznymi są w tym wypadku wszystkie dwuwymiarowe rozkłady brzegowe (liniowe sumy), czyli f_{ij+} , f_{i+k} oraz f_{+jk} , gdzie na przykład $f_{ij+} = \sum_k f_{ijk}$.

Jako przykład (te same dane analizują Sturmfels i Diaconis) rozważmy dane z badania opinii publicznej z 1972 roku dotyczące podejścia do aborcji wśród białych chrześcijan. Część analizowanych danych możemy zapisać w tablicy kontyngencji wymiaru $3 \times 3 \times 3$ (patrz tabela 5.1.2). Dane dotyczą łącznych obserwacji trzech zmiennych. Pierwsza zmienna to kościół, do którego przynależy dany chrześcijanin. Druga zmienna oznacza rodzaj wykształcenia, a trzecia to stosunek do aborcji (pozytywny, mieszany, negatywny).

Przeprowadziliśmy nasz algorytm dla tego przypadku. Ruchy dla łańcucha Markowa uzyskano za pomocą programu 4ti2. Jest to 81 ruchów opisanych w sekcji 4.3.

	Północni Protestanci			Południowi Protestanci			Katolicy		
	P	M	N						
Podstawowe	9	16	41	8	8	46	11	14	38
Średnie	85	52	105	35	29	54	47	35	115
Wyższe	77	30	38	37	15	22	25	21	42

Przeprowadzono $5 \cdot 10^6$ iteracji łańcucha. Okazuje się, że asymptotyczna statystyka dla tego modelu, czyli χ^2 o ośmiu stopniach swobody bardzo dobrze przybliża rozkład rzeczywistej statystyki. Wykres 2 przedstawia histogram wyników wraz z rozkładem asymptotycznym.



Rysunek 2: Histogram z eksperymentu Monte Carlo wraz z wykresem statystyki asymptotycznej $\chi^2_{\nu=8}$.

Widać zatem, że algorytm działa dobrze i daje się łatwo przeprowadzić. Uważne kontrolowanie porządku na indeksach dla tablic wyższych wymiarów wydaje się jedyną trudnością, na szczęście nie jest to wielki problem.

5.2 Regresja logistyczna

Na koniec podamy za Sturmfelsem i Diaconisem przykład inny niż tablice kontyngencji — regresję logistyczną. Jest to standardowa procedura budowania regresji dla danych dyskretnych.

Dla każdej z N obserwacji niech będzie dany binarny indyktor Y oraz wektor regresorów z . Zakładamy, że zmienne objaśniające z sąbrane ze skończonego podzbioru

i	0	1	2	3	4	5	6	7	8	9	10	11	12
$n_1(i)$	4	2	4	6	5	13	25	27	75	29	32	36	115
$n(i)$	6	2	4	9	10	20	34	42	124	58	77	95	360
$p(i)$	0,66	1	1	0,66	0,5	0,65	0,74	0,64	0,6	0,5	0,42	0,38	0,32

Tabela 3: Stosunek mężczyzn do stwierdzenia „Kobiety powinny zająć się prowadzeniem domu, a mężczyznom zostawić rządzenie państwem” (1974)

$\mathcal{A} \subset \mathbb{Z}^d$. Model logistyczny zadany jest przez relację postaci:

$$P(Y = 1|z) = \frac{e^{z \cdot \beta}}{1 + e^{z \cdot \beta}}, \quad P(Y = 0|z) = \frac{1}{1 + e^{z \cdot \beta}},$$

gdzie wektor parametrów $\beta \in \mathbb{R}^d$ ma być estymowany. Funkcja wiarygodności wynosi:

$$\prod_{i=1}^N e^{Y_i(z_i \cdot \beta)} / (1 + e^{z_i \cdot \beta}).$$

Niech $n(z)$ będzie liczbą indeksów $i \in \{1, 2, \dots, N\}$ takich, że $z_i = z$, a $n_1(z)$ niech będzie liczbą indeksów $i \in \{1, 2, \dots, N\}$ takich, że $z_i = z$ oraz $Y_i = 1$. Zestawienie $\{n(z)\}_{z \in \mathcal{A}} \in \mathbb{N}^{|\mathcal{A}|}$ oraz suma wektorowa $\sum_{z \in \mathcal{A}} n_1(z)z \in \mathbb{Z}^d$ tworzą wspólnie statystykę dostateczną (determinują funkcję wiarygodności).

Zapiszmy problem w notacji z poprzednich sekcji. Niech $\mathcal{X} = \{(0, z), (1, z), z \in \mathcal{A}\}$ oraz niech $T : \mathcal{X} \rightarrow \mathbb{Z}^{d+|\mathcal{A}|}$ będzie zdefiniowana jako

$$T(0, z) = (0; 0, \dots, 0, 1, 0, 0, \dots, 0) \quad (5.2)$$

$$T(1, z) = (z; 0, \dots, 0, 1, 0, 0, \dots, 0) \quad (5.3)$$

gdzie jedynka w ostatnich $|\mathcal{A}|$ współrzędnych pojawia się na z -ej pozycji. Wówczas dla konkretnych danych $f : \mathcal{X} \rightarrow \mathbb{N}$, suma $t = \sum_{x \in \mathcal{X}} f(x)T(x)$ ustala statystykę dostateczną.

Algorytm prezentowany w tym opracowaniu zastosujemy do przykładu modelu dla danych z badania opinii publicznej z 1974 roku, w którym mężczyźni zapytano o ich stosunek do stwierdzenia „Kobiety powinny zająć się prowadzeniem domu, a mężczyznom zostawić rządzenie państwem”. Niech $Y = 1$ jeżeli respondent się zgadza, a $Y = 0$ jeżeli nie. Dla każdego respondenta mamy podaną liczbę i lat jego edukacji $0 \leq i \leq 12$. Dane zawiera tablica 3.

Tutaj $n_1(z)$ to liczba próbujących stwierdzenie, a $n(i)$ to całkowita liczba jednostek, których edukacja przebiegała i lat. Podane jest również $p(i) = \frac{n_1(i)}{n(i)}$, stosunek próbujących. Te stosunki wydają się maleć wraz ze wzrostem i . Naturalne jest by próbować dopasować do tych danych regresję postaci:

$$P(Y = 1|i) = \frac{e^{\alpha+i \cdot \beta}}{1 + e^{\alpha+i \cdot \beta}}.$$

Przykład ten odpowiada zatem naszemu przypadkowi dla $d = 2$ (pojawia się wektor jedynek, który odpowiada wyrazowi wolnemu regresji), gdzie $\mathcal{A} = \{(1, 0), (1, 1), \dots, (1, 12)\}$.

Statystyki dostateczne w tym przypadku to:

$$\{n(i)\}_{i=0}^{12}, \quad \sum_{i=0}^{12} n_1(i), \quad \sum_{i=1}^{12} n_1(i)i.$$

Dla danych z tabeli 3 oszacowania największej wiarygodności dla parametrów α i β wyniosły odpowiednio $\hat{\alpha} = 2,0545$ oraz $\hat{\beta} = -0,2305$. Statystyka χ^2 dla jakości dopasowania wynosi $\sum_{i=0}^{12} (n(i)\hat{p}(i) - n_1(i))^2 / n(i)\hat{p}(i) = 11,951$. Nieregularne rozmieszczenie obserwacji oraz to, że kilka komórek charakteryzuje się niską liczebnością sprawia, że istnieją obawy, czy spełnione są założenia stosowalności testów asymptotycznych.

W tym wypadku algorytm wystartujemy z wektora danych długości $2 \cdot 13$ (bo $|\mathcal{X}| = 2 \cdot 13$), którego współrzędne mówią na przemian ile zaobserwowano osobników o danym i nie zgadzających się ze stwierdzeniem a ile zgadzających się:

f=[2 4 0 2 0 4 3 3 5 5 7 13 9 25 15 27 49 75 29 29 45 32 59 36 245 115]

Macierz statystyki dostatecznej T jest wymiaru 15×26 :

```

0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
0 0 0 1 0 2 0 3 0 4 0 5 0 6 0 7 0 8 0 9 0 10 0 11 0 12
1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1

```

Za pomocą programu 4ti2 dostajemy dla tej macierzy bazę Markowa składającą się z 16968 ruchów, które dają się łatwo wykorzystać w przeprowadzeniu eksperymentu.

6 Funkcja napisana w programie Matlab

Fabio Rapallo podał w swoich notatkach (patrz: <http://www.dima.unige.it/~rapallo/papers/tutorial.pdf>) funkcję napisaną w Matlabie, która umożliwia generowanie wartości p dla testów dla tablic kontyngencji. Program ten na wejściu bierze między innymi tablicę startową, jej oszacowanie największej wiarygodności oraz macierz, w której kolejne wiersze odpowiadają ruchom bazy Markowa. Po drobnych korektach program dodatkowo generuje histogram wyników (statystyk χ^2).

```

function [p,c]=simul(tab,MLE,m,N,bis,step);
%tab = the observed table;
%mle = maximum likelihood estimate;
%m = the matrix representation of the moves;
%N = number of MCMC replicates;
%bis = number of burn-in steps;
%step = reduction of correlation step;
%-----
% od PWZ: funkcja wyrzuca wartosc-p oraz wektor wartosci statystyk w kolejnych
% krokach, macierze tab i MLE podajemy w postaci wektorów leżących - wymaga to
% założenia implicite jakiegoś porządku leksykograficznego na współrzędnych

nc=length(tab);
p=0;
c=zeros(N,1);
chisqref=sum(((tab - MLE).^2)./MLE);
mm=-m;
m=[m; mm]; %PWZ: zamiast rzucac moneta, rozszerzamy po prostu ruchy
[nmoves,ncm]=size(m);
numit=N*step+bis; %N to liczba wysymulowanych ruchów, efektywnie trzeba więcej
for i=1:numit
    tabp=zeros(1,nc);
    r=ceil(nmoves*rand(1)); %losuje numer ruchu
    tabp=tab+m(r,:);
    if tabp>=zeros(1,nc) % sprawdza, czy mozna sie ruszyc w ta strone
        mhr=1; %bedziemy generowac krok metropolisa - losujemy wzgl. H_t
        for j=1:nc
            if m(r,j)~=0
                mhr=mhr*prod(1:tab(j))/prod(1:tabp(j)); %rozklad hipergeometryczny
            end;
        end;
        alpha=rand(1);
        if mhr>=alpha % regula akceptacji ruchu
            tab=tabp; % rusza sie
        end;
    end;
    if (rem(i-bis,step)==0) & (i>bis) %opuszcza w sympulacja bis ruchow i notuje
%co step ruchow aby uniknac autokorelacji
        c((i-bis)/step)=sum(sum(((tab - MLE).^2)./MLE));
        if c((i-bis)/step)>=chisqref %patrzy czy wartosc statystyki wyzsza
            p=p+1;
        end;
    end;
end;
end;

```

```
p=p/N;
% program wyrzuca wartosc p oraz c - wektor uzyskanych statystyk
```

Dodatkowymi parametrami programu jest liczba iteracji, liczba kroków symulacji do pominięcia oraz ewentualnie liczba okresów, co który ma być notowany wynik (aby uniknąć autokorelacji wyników). Z tym programem procedura przeprowadzania eksperymentu wydaje się już prosta. W pierwszej kolejności przy użyciu programu `4ti2` generujemy bazę Markowa. Program zapisze wynik w wygodnym formacie, bo każda linijka odpowiada innemu ruchowi. W przypadku użycia Matlaba, będzie trzeba z pliku `model.mar` wyciąć pierwszą linijkę, która mówi o wymiarach zapisanej w pliku macierzy. Wówczas na przykład dla tablicy 4×4 , którą omawialiśmy krótko w tym dokumencie wystarczy w Matlabie napisać:

```
tab=[68 119 26 7; 20 84 17 94; 15 54 14 10; 5 29 14 16;];
moves=load('4x4.mar');
```

aby załadować dane. Następnie przerabiamy naszą tablicę na wektor i wyliczamy jej oszacowanie największej wiarygodności (w przykładzie mamy tablicę 4×4 i model niezależności):

```
[nr,nc]=size(tab);
N=sum(sum(tab));
csum=sum(tab); % sumy po kolumnach
rsum=sum(tab'); %sumy po wierszach
tabv=reshape(tab',1,nc*nr);
MLE=rsum'*csum/N;
MLEv=reshape(MLE',1,nc*nr);
```

Następnie deklarujemy liczbę iteracji oraz resztę parametrów. Ostatecznie uruchamiamy funkcję `simul` napisaną przez Fabio Rapallo:

```
iter=1000000; %liczba iteracji
cor=5; %kroki redukujace korelacje
bis=10000; % liczba opuszczonych krokow
[p,stat]=simul(tabv,MLEv,moves,iter,bis,cor);
```

Przy tych parametrach, algorytm musi wykonać ponad 5 milionów ruchów. Jednak już po kilku minutach dostaniemy wyniki. Wpisując `p` dostaniemy wartość `p` dla naszego testu, czyli procent tablic o niższej wartości statystyki χ^2 niż nasza wyjściowa. Wpisując `hist(stat,150)`; dostajemy histogram wyników.

Jeżeli chodzi o modele innego typu to czasem potrzebne są modyfikacje. Na przykład w przypadku regresji logistycznej, w każdym kroku algorytmu konieczne jest estymowanie parametrów regresji, co należy uwzględnić w programie.

Literatura

[Agr96] Alan Agresti. *An Introduction to Categorical Data Analysis*. John Wiley & Sons, New York, 1996.

- [Agr02] Alan Agresti. *Categorical Data Analysis*. John Wiley & Sons, New York, wydanie drugie, 2002.
- [AH85] DF Andrews, AM Herzberg. *Data*. Springer-Verlag New York, 1985.
- [CLO96] David A. Cox, John B. Little, Don O’Shea. *Ideals, Varieties, and Algorithms*. Springer-Verlag, NY, wydanie 2nd, 1996. 536 pages.
- [DEH97] Persi Diaconis, David Eisenbud, Susan Holmes. Speeding up algebraic random walks. *Dept. Mathematics, Brandeis Univ. Preprint.*, 1997.
- [DS98] Persi Diaconis, Bernd Sturmfels. Algebraic algorithms for sampling from conditional distributions. *Ann. Statist.*, 26(1):363–397, 1998.
- [GMS02] Dan Geiger, Christopher Meek, Bernd Sturmfels. On the toric algebra of graphical models, 2002.
- [Lau96] Steffen L. Lauritzen. *Graphical models*. Number 17 serii Oxford Statistical Science Series. Clarendon Press, Oxford, 1996.
- [Lau02] Steffen L. Lauritzen. Lectures on Contingency Tables. electronic edition, 2002.
- [PRW01] Giovanni Pistone, Eva Riccomagno, Henry P. Wynn. *Algebraic Statistics: Computational Commutative Algebra in Statistics*. Chapman&Hall, 2001.
- [Stu96] Bernd Sturmfels. *Gröbner bases and convex polytopes*, wolumen 8 serii *University Lecture Series*. American Mathematical Society, Providence, RI, 1996.