



Diplomstudiengang Mathematik mit Schwerpunkt Biowissenschaften

## DIPLOMARBEIT

Ein Monte-Carlo Verfahren zur Erzeugung von  
Binärmatrizen und dessen Anwendung in der  
Ökologie

vorgelegt von: Arne Boll

Betreuende Gutachterin: Prof. Dr. Christine Müller

Zweiter Gutachter: Dr. Peter Harmand

eingereicht am: 16.07.2009

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Grundlagen</b>	<b>4</b>
2.1	Wahrscheinlichkeitstheorie . . . . .	4
2.2	Stichproben . . . . .	11
2.3	Schätzfunktionen . . . . .	13
2.4	Statistische Tests . . . . .	15
2.5	R-Funktionen . . . . .	18
2.5.1	Newton's Identitäten . . . . .	20
2.5.2	Rekursionsformeln für R-Funktionen . . . . .	22
<b>3</b>	<b>Eigenschaften von Binärmatrizen</b>	<b>25</b>
3.1	2x2-swaps . . . . .	27
3.2	Der Satz von Gale-Ryser und seine Anwendung . . . . .	29
3.2.1	Definitionen und Folgerungen . . . . .	29
3.2.2	Der Satz von Gale-Ryser . . . . .	34
3.2.3	Anwendung des Satzes von Gale-Ryser auf die Generierung von Binärmatrizen mit festen Zeilen- und Spaltensummen . . . . .	38
<b>4</b>	<b>Importance Sampling</b>	<b>46</b>
4.1	Funktionsweise des Importance Sampling . . . . .	46
4.2	Bestimmung der Güte beim Importance Sampling . . . . .	50
4.3	Sequentielles Importance Sampling . . . . .	54
<b>5</b>	<b>Bedingte Bernoulli-Modelle</b>	<b>56</b>
5.1	Bedingte Bernoulli-Modelle . . . . .	56
5.2	Legitimation des bedingten Bernoulli-Modells bei der Erzeugung von Binärmatrizen mit festen Zeilen- und Spaltensummen . . . . .	66
5.3	Stichprobensystem zur bedingten Bernoulli-Verteilung . . . . .	68
<b>6</b>	<b>Ein sequentielles Importance Sampling Verfahren zur Erzeugung von Binärmatrizen mit festen Zeilen- und Spaltensummen</b>	<b>70</b>

6.1	Der Algorithmus . . . . .	70
6.2	Verbesserung der Probeverteilung . . . . .	74
6.2.1	Anpassung der Gewichte . . . . .	74
6.2.2	Anpassung der Datenmatrix . . . . .	74
<b>7</b>	<b>Anwendung des SIS-Verfahrens auf ökologische Daten</b>	<b>76</b>
7.1	Diamond's Versammlungs-Regeln . . . . .	77
7.2	Teststatistiken zum Testen auf Konkurrenz . . . . .	78
7.2.1	Eigenschaften der Statistiken . . . . .	82
7.3	Auswertung ökologischer Daten . . . . .	87
7.3.1	Darwins Finken . . . . .	88
7.3.2	Patterson Atmar Datensatz . . . . .	91
7.4	Fazit . . . . .	98
<b>8</b>	<b>Anhang</b>	<b>100</b>
8.1	Der SIS-Algorithmus . . . . .	100

# 1 Einleitung

Binärmatrizen sind Matrizen über der Menge  $\{0, 1\}$ . Solche Binärmatrizen werden häufig im Bereich der Ökologie aufgestellt. Sie beschreiben dort meist die An- beziehungsweise Abwesenheit mehrerer Arten in verschiedenen Lebensräumen und werden daher auch Presence-Absence-Matrizen genannt.

Eine Zeile einer Binärmatrix entspricht einer der beobachteten Arten und jeweils eine Spalte einen der Lebensräume. Eine Eins in Zeile  $i$  und Spalte  $j$  gibt die Anwesenheit der  $i$ -ten Art im  $j$ -ten Lebensraum und eine Null entsprechend die Abwesenheit an.

In der Regel ist man daran interessiert, ob interspezifische Konkurrenz die beobachteten Arten bei ihrer Lebensraumwahl beeinflusst hat oder nicht. Um eine solche Hypothese bei einer konkreten Presence-Absence-Matrix statistisch zu überprüfen, werden zum einen geeignete Teststatistiken und zum anderen Vergleichswerte benötigt.

Verfahren, die zu einer gegebenen Problemstellung unabhängige Stichproben mittels eines geeigneten Computerprogramms generieren, nennt man Monte-Carlo Verfahren.

In dieser Arbeit wird es darum gehen ein Monte-Carlo Verfahren zu erarbeiten, um auf der Menge aller Binärmatrizen mit denselben Zeilen- und Spaltensummen wie eine auf beobachteten Daten basierende Binärmatrix gleichverteilte, beziehungsweise annähernd gleichverteilte Binärmatrizen zu generieren. Dieses Verfahren wird dann anhand verschiedener Presence-Absence-Matrizen auf seine Effizienz überprüft.

In Kapitel 2 werden zunächst einige Grundlagen aus der Wahrscheinlichkeitstheorie, der Statistik sowie der Kombinatorik angegeben. Diese Grundlagen sind für das Verständnis der weiteren Arbeit notwendig.

Das dritte Kapitel befasst sich näher mit Binärmatrizen. Es werden dort für Monte-Carlo Verfahren wichtige Eigenschaften von Binärmatrizen vorgestellt. In dieser Arbeit werden ausschliesslich Binärmatrizen mit vorher festgelegten Zeilen- und Spaltensummen betrachtet. Dadurch wird auf natürliche Unterschiede der betrachteten Arten beziehungsweise Lebensräume eingegangen. Größere Lebensräume können beispielsweise mehr unterschiedlichen Arten Platz bieten, als kleine Lebensräume. In Abschnitt 3.1 wird gezeigt, dass sämtliche Binärmatrizen mit denselben Zeilen- und

Spaltensummen durch sogenannte 2-swaps ineinander umwandelbar sind. Abschnitt 3.2 befasst sich mit dem Satz von Gale-Ryser nach [28]. Dieser gibt an, unter welchen Umständen eine Binärmatrix zu gegebenen Zeilen- und Spaltensummen existiert.

Kapitel 4 behandelt ein spezielles Monte-Carlo Verfahren, das Importance Sampling. Soll eine Stichprobe von einer bestimmten unbekanntem Verteilung generiert werden, so kann Importance Sampling angewendet werden. Dabei wird die Stichprobe anstatt von der geforderten Verteilung von einer, dieser möglichst ähnlichen Verteilung generiert.

Spezielle Maße, wie der Chi-Quadrat-Abstand, geben dabei wieder, wie ähnlich sich die beiden Verteilungen sind. Dadurch wird die Güte der zur Erzeugung der Stichprobe verwendeten Verteilung angegeben. Es wird außerdem gezeigt, warum es sinnvoll ist Binärmatrizen spaltenweise und somit sequentiell zu generieren und welchen Vorteil dies für das Finden einer geeigneten Verteilung für das Importance Sampling bietet.

In Kapitel 5 wird eine für die Erzeugung von gleichverteilten Binärmatrizen geeignete Verteilung ermittelt. Da in dieser Arbeit Binärmatrizen spaltenweise generiert werden, muss bei der Betrachtung einer bestimmten Spalte eine Stichprobe von Nullen und Einsen abhängig von der Spaltensumme erzeugt werden. Dies wird durch so genannte bedingte Bernoulli-Modelle beschrieben.

Abschnitt 5.1 behandelt die allgemeine Theorie um bedingte Bernoulli-Modelle und wie man eine Dichte eines bedingten Bernoulli-Modells entwickeln kann. Der Abschnitt richtet sich dabei nach [5]. Dabei spielt auch die Eigenschaft der Entropie einer Verteilung eine wichtige Rolle.

Abschnitt 5.2 erläutert aus welchem Grund bedingte Bernoulli-Modelle eine geeignete Verteilung für das sequentielle Importance Sampling von Binärmatrizen mit festen Zeilen- und Spaltensummen liefert. Abschnitt 5.3 gibt schließlich ein Verfahren zur Ziehung einer Stichprobe gemäß der bedingten Bernoulli-Modelle an.

Das sechste Kapitel liefert dann das bei der Auswertung von Presence-Absence-Matrizen verwendete Importance Sampling Verfahren nach [7] sowie Anpassungsmöglichkeiten an unterschiedliche Situationen.

Kapitel 7 befasst sich dann mit der Auswertung konkreter ökologischer Daten. Dazu wird zunächst erläutert, wie sich Tests bei Binärmatrizen durchführen lassen. Abschnitt 7.1 enthält Diamonds Versammlungsregeln (nach [10]), die Aspekte der interspezifischen Konkurrenz wiedergeben. In Abschnitt 7.2 werden dann basierend auf Diamonds Regeln Teststatistiken angegeben, mit denen überprüft werden kann, ob zwischen den verschiedenen in einer Binärmatrix wiedergegebenen Arten ein Konkurrenzzusammenhang besteht oder nicht.

In Abschnitt 7.3 wird an einem größeren Datensatz ökologischer Binärmatrizen das

verwendete Importance Sampling Verfahren auf seine Tauglichkeit bei der statistischen Auswertung dieser Matrizen überprüft. Gleichzeitig wird festgestellt, ob bei den vorliegenden Matrizen ein Konkurrenzzusammenhang wahrscheinlich ist. Im Anhang ist der Quelltext des Importance Sampling Algorithmus angegeben.

## 2 Grundlagen

In diesem Kapitel werden einige im Laufe dieser Arbeit benötigte, grundlegende Definitionen und Sätze eingeführt.

### 2.1 Wahrscheinlichkeitstheorie

In dieser Arbeit kommen ausschließlich diskrete Mengen zur Anwendung. Es sei also im Folgenden immer  $(\Omega, P)$  ein diskreter Wahrscheinlichkeitsraum, wobei die Menge  $\Omega = \{\omega_1, \dots, \omega_N\}$  diskret ist und als Population bezeichnet wird.

$\mathcal{P}(\Omega) = \{A \mid A \subset \Omega\}$  bezeichne die Potenzmenge von  $\Omega$  und jedes  $A \in \mathcal{P}$  wird als Ereignis bezeichnet. Die für die Definition eines Maßes auf der Menge  $\Omega$  notwendige  $\sigma$ -Algebra sei im Folgenden stets durch  $\mathcal{P}(\Omega)$  gegeben.  $P$  sei ein Wahrscheinlichkeitsmaß auf  $\mathcal{P}(\Omega)$  im Sinne von Definition 2.1.

**Definition 2.1 (Wahrscheinlichkeitsmaß)** (nach [18] S. 31)

Sei  $\mathcal{P}(\Omega) = \{A \mid A \subset \Omega\}$  die Potenzmenge einer abzählbaren Menge  $\Omega$ . Eine Abbildung  $P : \mathcal{P}(\Omega) \rightarrow [0, 1]$  mit  $A \mapsto P(A)$  heißt Wahrscheinlichkeitsmaß (auf  $\mathcal{P}(\Omega)$ ), wenn

$$P(\Omega) = 1 \tag{2.1}$$

und für eine paarweise disjunkte Folge  $(A_n)_{n \in \mathbb{N}} \in \mathcal{P}(\Omega)$

$$P\left(\bigcup_{n \in \mathbb{N}} A_n\right) = \sum_{n \in \mathbb{N}} P(A_n) \tag{2.2}$$

gilt. Dabei wird die Eigenschaft (2.1) Normiertheit und die Eigenschaft (2.2)  $\sigma$ -Additivität genannt.  $(\Omega, P)$  wird dann als diskreter Wahrscheinlichkeitsraum bezeichnet.

**Definition 2.2 (bedingte Wahrscheinlichkeit)** (nach [18] S. 21)

Es sei  $(\Omega, P)$  ein diskreter Wahrscheinlichkeitsraum und  $A$  und  $B$  seien zwei beliebige Ereignisse aus  $\mathcal{P}(\Omega)$  mit  $P(B) > 0$ . Dann ist

$$P(A \mid B) := \frac{P(A \cap B)}{P(B)} \tag{2.3}$$

die bedingte Wahrscheinlichkeit von  $A$  unter der Bedingung, dass  $B$  bereits eingetreten ist.

**Satz 2.3** (nach [18] S. 23)

Sei  $(\Omega, P)$  ein diskreter Wahrscheinlichkeitsraum und  $P(B) > 0$  für ein Ereignis  $B \subset \Omega$ . Durch die Abbildung  $P_B : \mathcal{P}(\Omega) \rightarrow [0, 1]$  mit  $P_B(A) := P(A | B)$  ist ein Wahrscheinlichkeitsmaß auf  $\mathcal{P}(\Omega)$  definiert.

BEWEIS (nach [18] S. 23)

Es genügt die Normiertheit und die  $\sigma$ -Additivität von  $P_B$  nachzuweisen. Es gilt

$$P_B(\Omega) = \frac{P(\Omega \cap B)}{P(B)} = \frac{P(B)}{P(B)} = 1$$

und mit einer disjunkten Folge  $(A_n)_{n \in \mathbb{N}}$  mit  $A_n \subset \Omega$  für alle  $n \in \mathbb{N}$  ist auch die Folge  $(A_n \cap B)_{n \in \mathbb{N}}$  mit  $(A_n \cap B) \subset \Omega$  für alle  $n \in \mathbb{N}$  disjunkt und somit

$$\begin{aligned} P\left(\bigcup_{n \in \mathbb{N}} A_n \middle| B\right) &= \frac{P((\bigcup_{n \in \mathbb{N}} A_n) \cap B)}{P(B)} = \frac{P(\bigcup_{n \in \mathbb{N}} (A_n \cap B))}{P(B)} \\ &= \frac{\sum_{n \in \mathbb{N}} P(A_n \cap B)}{P(B)} = \sum_{n \in \mathbb{N}} P(A_n | B). \quad \blacksquare \end{aligned}$$

**Satz 2.4** (nach [18] S. 22)

Seien  $A_1, \dots, A_n \subset \Omega$  mit  $P(A_1 \cap \dots \cap A_n) > 0$ . Dann lässt sich  $P(A_1 \cap \dots \cap A_n)$  als Produkt bedingter Wahrscheinlichkeiten darstellen. Es ist

$$P(A_1 \cap \dots \cap A_n) = P(A_1)P(A_2 | A_1)P(A_3 | A_1 \cap A_2) \cdots P(A_n | A_1 \cap \dots \cap A_{n-1}). \quad (2.4)$$

BEWEIS (nach [18] S. 23)

Diese Begebenheit lässt sich einfach mittels vollständiger Induktion nach  $n$  beweisen. Für  $n = 2$  erhält man durch Umformen von (2.3) für  $A_1, A_2 \subset \Omega$  und  $P(A_1 \cap A_2) > 0$  (und somit auch  $P(A_2) > 0$ )

$$P(A_1 \cap A_2) = P(A_1)P(A_2 | A_1).$$

Für  $n \Rightarrow n + 1$  lässt sich zunächst für beliebige Ereignisse  $A_1, \dots, A_{n+1} \subset \Omega$  mit  $P(A_1 \cap \dots \cap A_{n+1}) > 0$  die Gleichung (2.3) anwenden und man erhält

$$P(A_1 \cap \dots \cap A_n \cap A_{n+1}) \stackrel{(2.3)}{=} P(A_{n+1} | A_1 \cap \dots \cap A_n)P(A_1 \cap \dots \cap A_n).$$

Es lässt sich nun die Induktionsvoraussetzung (2.4) anwenden, sodass weiter

$$\begin{aligned}
 P(A_1 \cap \dots \cap A_n \cap A_{n+1}) &\stackrel{(2.4)}{=} P(A_{n+1} \mid A_1 \cap \dots \cap A_n) \\
 &P(A_1)P(A_2 \mid A_1)P(A_3 \mid A_1 \cap A_2) \\
 &\dots P(A_n \mid A_1 \cap \dots \cap A_{n-1}) \\
 &= P(A_1)P(A_2 \mid A_1)P(A_3 \mid A_1 \cap A_2) \\
 &\dots P(A_{n+1} \mid A_1 \cap \dots \cap A_n)
 \end{aligned}$$

gilt und der Satz somit bewiesen ist. ■

**Definition 2.5 (Unabhängigkeit)** (nach [18] S. 26)

Sei  $(\Omega, P)$  ein diskreter Wahrscheinlichkeitsraum und  $\{A_i \mid i \in I\} \subset \mathcal{P}(\Omega)$  bezeichne eine Familie von Ereignissen, wobei  $I \subset \mathbb{N}$  sei. Diese Familie heißt unabhängig, wenn für jede endliche Teilfamilie  $\{A_i \mid i \in J \subset I\}$

$$P\left(\bigcap_{i \in J} A_i\right) = \prod_{i \in J} P(A_i)$$

gilt.

**Satz 2.6 (diskrete Dichte/Zähldichte)** (nach [14] S. 18)

Sei  $(\Omega, \mathcal{P}(\Omega))$  ein Ereignisraum und  $p = (p(\omega))_{\omega \in \Omega}$  eine Folge in  $[0, 1]$ , für die  $\sum_{\omega \in \Omega} p(\omega) = 1$  gilt. Dann wird durch  $p$  genau ein Wahrscheinlichkeitsmaß  $P$  auf  $\mathcal{P}(\Omega)$  durch

$$P(A) := \sum_{\omega \in A} p(\omega) \quad \text{für } A \in \mathcal{P}(\Omega)$$

definiert. Jedes  $p$  dieser Art heißt diskrete Dichte oder Zähldichte zu  $P$ .

BEWEIS

(2.1) und (2.2) sind offensichtlich erfüllt und somit ist  $P$  ein Wahrscheinlichkeitsmaß. ■

**Definition 2.7 (Produkttraum)** (nach [18] S. 27)

Es seien  $(\Omega_1, P_1), \dots, (\Omega_n, P_n)$  beliebige diskrete Wahrscheinlichkeitsräume. Durch das kartesische Produkt

$$\Omega := \prod_{i=1}^n \Omega_i := \{\omega = (\omega_1, \dots, \omega_n) \mid \omega_i \in \Omega_i \text{ für } i \in \{1, \dots, n\}\}$$

und

$$P := \prod_{i=1}^n P_i,$$

wobei  $P(\omega) := \prod_{i=1}^n P_i(\omega_i)$  gelte, sei der diskrete Wahrscheinlichkeitsraum  $(\Omega, P)$  als Produktraum von  $(\Omega_1, P_1), \dots, (\Omega_n, P_n)$  definiert.

Durch  $P$  ist dabei trivialerweise ein Wahrscheinlichkeitsmaß auf  $\mathcal{P}(\Omega)$  definiert und  $(\Omega, P)$  bilden einen Wahrscheinlichkeitsraum.

**Definition 2.8 (Zufallsvariable)** (nach [18] S. 42)

Sei  $(\Omega, P)$  ein diskreter Wahrscheinlichkeitsraum und  $\mathcal{X}$  eine beliebige Menge. Eine beliebige Abbildung  $X : \Omega \rightarrow \mathcal{X}$  wird diskrete Zufallsvariable (auf  $\Omega$ ) genannt, wenn der Wertebereich von  $X$  auch diskret ist.

**Definition 2.9 (Verteilung)** (nach [18] S. 42)

Sei  $(\Omega, P)$  ein diskreter Wahrscheinlichkeitsraum und  $X : \Omega \rightarrow \mathcal{X}$  eine diskrete Zufallsvariable mit Werten in einer beliebigen Menge  $\mathcal{X}$ .  $P(\{\omega \in \Omega \mid X(\omega) = x\})$  beschreibt die Wahrscheinlichkeit für  $X$  einen bestimmten Wert  $x \in \mathcal{X}$  anzunehmen. Man schreibt meist kurz  $P(X = x)$ .  $P$  wird dabei Verteilung von  $X$  genannt.

**Definition 2.10 (gemeinsame Verteilung)** (nach [18] S. 43)

Sei  $(\Omega, P)$  ein diskreter Wahrscheinlichkeitsraum,  $n \in \mathbb{N}$  und seien  $X_i : \Omega \rightarrow \mathcal{X}_i$  für  $i \in \{1, \dots, n\}$  diskrete Zufallsvariablen, dann kann man diese zu einer diskreten Zufallsvariablen  $X$  zusammenfassen. Der Wertebereich von  $X$  ist dann durch die Menge  $\mathcal{X} := \mathcal{X}_1 \times \dots \times \mathcal{X}_n$  gegeben und es ist  $X(\omega) := (X_1(\omega), \dots, X_n(\omega))$ . Die Verteilung von  $X$  sei durch

$$P(X = x) := P(X_1 = x_1, \dots, X_n = x_n)$$

mit  $x = (x_1, \dots, x_n) \in \mathcal{X}$  bestimmt und heißt gemeinsame (diskrete) Verteilung von  $X_1, \dots, X_n$ .

**Definition 2.11 (Randverteilung)** (nach [18] S. 44)

Sei  $(X_1, \dots, X_n)$  für  $n \in \mathbb{N}$  ein  $n$ -dimensionaler Zufallsvektor auf  $(\Omega, P)$  mit Werten in  $\mathcal{X} := \mathcal{X}_1 \times \dots \times \mathcal{X}_n$  und  $\{i_1, \dots, i_k\} \subset \{1, \dots, n\}$ . Man nennt dann die Verteilung von  $(X_{i_1}, \dots, X_{i_k})$  ( $k$ -dimensionale) Randverteilung von  $(X_1, \dots, X_n)$ .

Als gemeinsamer Wahrscheinlichkeitsraum wird oftmals der Produktraum angenommen.

**Definition 2.12 (Unabhängigkeit von Zufallsvariablen)** (nach [18] S. 45)

Sei  $(\Omega, P)$  ein diskreter Wahrscheinlichkeitsraum und  $\{X_i : \Omega \rightarrow \mathcal{X}_i \mid i \in I \subset \mathbb{N}\}$  eine Familie diskreter Zufallsvariablen. Diese Familie heißt unabhängig, wenn für alle  $A_i \in \mathbb{X}_i$  die Ereignisse  $\{X_i \in A_i\}$  für  $i \in I$  unabhängig sind.

**Satz 2.13** (nach [18] S. 45)

Sei  $(\Omega, P)$  ein diskreter Wahrscheinlichkeitsraum,  $n \in \mathbb{N}$  und seien  $X_i : \Omega \rightarrow \mathcal{X}_i$  für  $i \in \{1, \dots, n\}$  diskrete Zufallsvariablen.  $X_1, \dots, X_n$  sind genau dann unabhängig, wenn für alle  $(x_1, \dots, x_n) \in \mathcal{X}_1 \times \dots \times \mathcal{X}_n$

$$P(X_1 = x_1, \dots, X_n = x_n) = \prod_{i=1}^n P(X_i = x_i)$$

gilt.

BEWEIS

Der Beweis dieses Satzes ergibt sich durch Anwenden des Unabhängigkeitsbegriffs für Ereignisse und ist beispielsweise in [18] auf Seite 45 nachzulesen. ■

Die bedingte Wahrscheinlichkeit kann anstatt für Ereignisse auch für diskrete Zufallsvariablen definiert werden. Sind  $X, Y$  diskrete Zufallsvariablen auf  $\Omega$  mit Werten in  $\mathcal{X}$  und Verteilung  $P$  und gilt  $P(Y = y) > 0$  für ein  $y \in \mathcal{X}$ , dann sei für alle  $x \in \mathcal{X}$

$$P(X = x \mid Y = y) := \frac{P(X = x, Y = y)}{P(Y = y)}.$$

Analog zu Satz 2.4 gilt für diskrete Zufallsvariablen  $X_1, \dots, X_n$  auf  $\Omega$  mit Werten in  $\mathcal{X}$  und Verteilung  $P$

$$\begin{aligned} P(X_1 = x_1, \dots, X_n = x_n) &= P(X_1 = x_1)P(X_2 = x_2 \mid X_1 = x_1) \\ &\quad \cdots P(X_n = x_n \mid X_1 = x_1, \dots, X_{n-1} = x_{n-1}). \end{aligned} \tag{2.5}$$

Ebenso ist der Begriff der Dichte auch für Zufallsvariablen sinnvoll. Sei  $X$  diskrete Zufallsvariable auf  $(\Omega, P)$  mit Werten in  $\mathcal{X}$ . Dann bezeichne  $p_X(x) := P(X = x)$  für alle  $x \in \mathcal{X}$  die diskrete Dichte der Verteilung von  $X$ .

Eine wichtige Grundlage in den später folgenden Kapiteln ist der Begriff der bedingten Dichte.

**Definition 2.14 (diskrete bedingte Dichte)** (nach [32] S. 84)

Seien  $X : \Omega \rightarrow \mathcal{X}_1$  und  $Y : \Omega \rightarrow \mathcal{X}_2$  diskrete Zufallsvariablen auf  $\Omega$  mit Verteilung  $P$ . Es gelte für die Randverteilung  $P(Y = y) > 0$  für alle  $y \in \mathcal{X}_2$ . Dann bezeichne

$$p_{X|Y=y}(x) := \frac{p_{(X,Y)}(x, y)}{p_Y(y)} \tag{2.6}$$

die diskrete bedingte Dichte von  $X$ , wobei  $Y = y$  gegeben ist und  $p_{(X,Y)}(x, y)$  die gemeinsame diskrete Dichte von  $X$  und  $Y$  bezeichnet.

**Definition 2.15 (Erwartungswert)** (nach [18] S. 46)

Sei  $X : \Omega \rightarrow \mathcal{X} \subset \mathbb{R}$  eine diskrete Zufallsvariable auf dem Wahrscheinlichkeitsraum  $(\Omega, P)$ . Gilt

$$\sum_{\omega \in \Omega} |X(\omega)|P(\omega) < \infty,$$

so besitzt  $X$  einen Erwartungswert, der dann durch

$$E_P(X) := \sum_{\omega \in \Omega} X(\omega)P(\omega) \quad (2.7)$$

gegeben ist. Besteht keine Verwechslungsgefahr bezüglich  $P$ , so schreibt man auch nur  $E(X)$ .

**Definition 2.16 (Varianz/Standardabweichung)** (nach [18] S. 52)

Sei  $X : \Omega \rightarrow \mathcal{X} \subset \mathbb{R}$  eine diskrete Zufallsvariable auf  $(\Omega, P)$ . Existiert  $E(X^2)$ , dann heißt

$$\text{Var}(X) = \text{Var}_P(X) := E((X - E(X))^2) = E(X^2) - E(X)^2 \quad (2.8)$$

Varianz von  $X$  bezüglich  $P$ . Der Ausdruck  $\sqrt{\text{Var}(X)}$  wird als Standardabweichung von  $X$  bezeichnet.

In dieser Arbeit wird die Bernoulli-Verteilung eine große Rolle spielen. Die Bernoulli-Verteilung beschreibt ein Experiment mit genau zwei möglichen Ereignissen. Ein solches Experiment nennt man auch Bernoulli-Experiment.

Ein bekanntes Beispiel ist der Münzwurf, ein anderes das Ziehen von Kugeln aus einem Beutel mit Kugeln in zwei verschiedenen Farben. Beim Münzwurf wird in der Regel angenommen, dass das Ereignis „Kopf“ die gleiche Eintrittswahrscheinlichkeit besitzt wie das Ereignis „Zahl“. Beim Ziehen der Kugeln hängt die Eintrittswahrscheinlichkeit der jeweiligen Farbe von der Anzahl der verschiedenfarbigen Kugeln im Beutel ab.

Bezeichnet man die beiden möglichen Ereignisse mit 0 und 1 so ist  $\Omega^n = \{0, 1\}^n$  der Stichprobenraum für  $n$ -maliges Werfen beziehungsweise Ziehen. Zunächst benötigt man für die Definition der Bernoulli-Verteilung eine geeignete Dichte für den Ereignisraum  $(\Omega^n, \mathcal{P}(\Omega^n))$ .

**Definition 2.17 (Produktdichte)** (nach [14] S. 32)

Sei  $p$  eine diskrete Dichte auf  $\Omega$ , dann heißt die diskrete Dichte

$$p^n(\omega) := \prod_{i=1}^n p(\omega_i) \quad (2.9)$$

$n$ -fache Produktdichte von  $p$  auf  $\Omega^n$ .

**Definition 2.18 (Bernoulli-Verteilung für  $n$  Alternativ-Versuche)** (nach [14] S. 32)

Sei  $p$  eine Dichte auf  $\Omega$ . Es bezeichne  $p^n(\omega) = \prod_{i=1}^n p(\omega_i)$  die  $n$ -fache Produktdichte von  $p$  auf  $\Omega^n$  mit  $\omega = (\omega_1, \dots, \omega_n)$  und  $\omega_i \in \Omega$  für  $i \in \{1, \dots, n\}$ . Sei nun  $\Omega = \{0, 1\}$  und  $p(1) = q \in [0, 1]$ , dann erhält man die Produktdichte

$$p^n(\omega) = q^{\sum_{i=1}^n \omega_i} (1 - q)^{n - \sum_{i=1}^n \omega_i}. \quad (2.10)$$

Das zu  $p^n$  gehörige Wahrscheinlichkeitsmaß  $P$  heißt dann Bernoulli-Verteilung für  $n$  Alternativ-Versuche mit der Erfolgswahrscheinlichkeit  $q$ .

Stark verwandt mit der Bernoulli-Verteilung ist die Binomialverteilung. Sie beschreibt die Wahrscheinlichkeit, dass bei  $n$ -maligem Durchführen eines Bernoulli-Experiments genau  $k$  Erfolge auftreten.

**Definition 2.19 (Binomial-Verteilung)** (nach [14] S. 34)

Sei  $\mathcal{X} = \{0, 1, \dots, n\}$  für  $n \in \mathbb{N}$ . Es beschreibe

$$\mathcal{B}_{n,p}(k) = \binom{n}{k} p^k (1 - p)^{n-k} \quad (2.11)$$

für  $k \in \mathcal{X}$  und  $p \in [0, 1]$  eine Dichte auf  $\mathcal{X}$ . Das zur Dichte  $\mathcal{B}_{n,p}(k)$  gehörige Wahrscheinlichkeitsmaß wird ebenfalls mit  $\mathcal{B}_{n,p}$  bezeichnet und Binomialverteilung genannt.

**Folgerung 2.20** (nach [14] S. 34)

Seien  $\Omega = \{0, 1\}$  und  $X_1, \dots, X_n$  unabhängige Alternativ-Versuche eines Bernoulli-Experiments mit Werten in  $\Omega$  mit der Erfolgswahrscheinlichkeit  $p \in [0, 1]$ . Sei weiter  $\mathcal{X} = \{0, 1, \dots, n\}$  mit  $n \in \mathbb{N}$ . Die Zufallsvariable  $X = \sum_{i=1}^n X_i$  mit

$$X : \Omega^n \rightarrow \mathcal{X} \quad \text{und} \quad \omega \rightarrow \sum_{i=1}^n \omega_i$$

ist dann  $\mathcal{B}_{n,p}$ -verteilt. □

BEWEIS

Der Beweis folgt direkt aus den Definitionen 2.19 und 2.18. Es beschreibt  $\binom{n}{k}$  die Anzahl aller möglichen  $n$ -Tupel  $\omega = (\omega_1, \dots, \omega_n)$  mit  $\sum_{i=1}^n \omega_i = k$ . Die Wahrscheinlichkeit für jedes Element  $\omega \in \Omega^n$  ist durch (2.10) gegeben.

Sei  $P$  die Verteilung von  $X$ , dann folgt sofort

$$P(\{\omega \in \Omega^n \mid X(\omega) = k\}) = P(\{\omega \in \Omega^n \mid \sum_{i=1}^n \omega_i = k\}) \stackrel{(2.10)}{=} \binom{n}{k} p^k (1-p)^{n-k}$$

und somit  $P = \mathcal{B}_{n,p}$ . ■

Besitzen die  $n$  Bernoulli-Experimente nicht die gleiche Erfolgswahrscheinlichkeit, so benötigt man eine weitere Verteilung, die in Kapitel 5 vorgestellt wird.

## 2.2 Stichproben

Dieser Abschnitt richtet sich weitestgehend nach [2] und stellt Grundlagen für die später angewandten Methoden zur Verfügung.

Sei  $\Omega = \{\omega_1, \dots, \omega_N\}$  und  $\mathcal{P}(\Omega)$  die Potenzmenge von  $\Omega$ . Für eine Teilmenge  $A \subset \Omega$  bezeichne  $|A|$  die Anzahl der Elemente in  $A$  und  $A^c = \Omega \setminus A$  das Komplement von  $A$ . Eine Stichprobe bezeichnet Realisierungen eines Zufallsexperiments auf  $(\Omega, P)$ . Eine nicht notwendigerweise sortierte Menge  $s = \{s_1, \dots, s_n\}$  von  $n \in \{1, \dots, N\}$  verschiedenen Elementen aus  $\Omega$  bildet eine Stichprobe aus  $\Omega$  ohne zurücklegen. Da  $\mathcal{P}(\Omega)$  die Menge aller Teilmengen von  $\Omega$  ist, beinhaltet  $\mathcal{P}(\Omega)$  alle möglichen Stichproben ohne zurücklegen von der Population  $\Omega$  und wird Stichprobenraum genannt. Lässt man die Größe  $n \in \{1, \dots, N\}$  der auszuwählenden Stichprobe fest, bezeichne  $\Omega_n = \{A \subset \Omega \mid |A| = n\}$  die Menge aller Stichproben der gewünschten Größe. Im Folgenden sei  $n \in \{1, \dots, N\}$  stets fest und es werden Stichproben ohne zurücklegen gebildet.

**Definition 2.21 (Stichprobenverfahren)** *(nach [2] S. 1897)*

*Ein Wahrscheinlichkeitsmaß  $P$ , das auf  $\Omega_n$  definiert ist, heißt Stichprobenverfahren, wenn  $P(s) \geq 0$  und  $\sum_{s \in \Omega_n} P(s) = 1$  für alle  $s \in \Omega_n$  gilt.*

**Definition 2.22 (Stichprobensystem)** *(nach [2] S. 1897)*

*Ein Stichprobensystem ist ein Vorgang zur Ziehung von Stichproben nach einem Stichprobenverfahren.*

In Abschnitt 5.3 wird ein Stichprobensystem aufgestellt, mit dem es möglich sein wird, Stichproben von Binärmatrizen mit gewissen Anforderungen zu generieren. Eine Möglichkeit eines Stichprobensystems ist im folgenden Beispiel 2.23 angegeben.

**Beispiel 2.23** (nach [2] S. 1897)

Man reihe alle möglichen Stichproben  $s_{(1)}, \dots, s_{(|\Omega_n|)}$  aus  $\Omega_n$  in einer beliebigen Reihenfolge auf und gebe jeder Stichprobe  $s_{(i)}$  für  $i \in \{1, \dots, |\Omega_n|\}$  die gleiche Wahrscheinlichkeit ausgewählt zu werden.  $\square$

Der Nachteil des in Beispiel 2.23 vorgestellten Stichprobensystems ist, dass zunächst sämtliche Elemente aus  $\Omega_n$  bekannt sein müssen. Dies kann bei einer größeren Population  $\Omega$  recht aufwändig werden, da es  $N!/(n!(N-n)!)$  mögliche Stichproben gibt.

Weniger aufwändig ist es, eine Stichprobe Schritt für Schritt auszuwählen. Dabei nimmt man bei jedem Schritt ein Element von den noch nicht ausgewählten Elementen aus  $\Omega$  mit einer geeigneten Wahrscheinlichkeit in die Stichprobe auf. Ein solches Stichprobensystem ist im folgenden Beispiel 2.24 für eine ungeordnete Stichprobe ohne zurücklegen angegeben.

**Beispiel 2.24 (Schritt für Schritt)** (nach [2] S. 1897)

In diesem Verfahren bezeichne  $s^{(k)}$  die Menge der Elemente, die nach  $k$  Schritten in die Stichprobe aufgenommen wurden.

Man beginne mit  $s^{(0)} = \emptyset$  und nehme in Schritt  $k > 0$  ein Element  $j$  aus  $(s^{(k-1)})^c$  mit der Wahrscheinlichkeit

$$P_1(j \mid s^{(k-1)}) := \frac{\sum_{s \in \Omega_n, s^{(k-1)} \subset s, j \in s} P(s)}{(n-k-1) \sum_{s \in \Omega_n, s^{(k-1)} \subset s} P(s)} \quad (2.12)$$

in die Stichprobe auf, wobei  $P$  ein Stichprobenverfahren auf  $\Omega_n$  sei. Es ist dann  $s^{(k)} = s^{(k-1)} \cup \{j\}$ , wenn  $j$  das ausgewählte Element ist. Gibt man jedem Element die gleiche Wahrscheinlichkeit in der Stichprobe enthalten zu sein, dann ist für die Stichprobe  $s = (s_1, \dots, s_n)$

$$P(s) = \frac{1}{N(N-1)(N-2) \cdots (N-n+1)}. \quad (2.13)$$

Nach  $n$  Schritten hat man eine Stichprobe der gewünschten Größe.  $\square$

Man kann den verschiedenen Elementen aus  $\Omega$  auch unterschiedliche Wahrscheinlichkeiten geben, gezogen zu werden. Möchte man beispielsweise Binärmatrizen mit festen Zeilen- und Spaltensummen spaltenweise generieren, dann kann man einen Eintrag eher mit einer 1 belegen, wenn die entsprechende Zeilensumme groß ist. Dieser Fall wird in Kapitel 5 beschrieben.

## 2.3 Schätzfunktionen

Häufig ist in der Statistik die Verteilung eines Zufallsexperiments nicht genau bekannt. So kann zum Beispiel die Erfolgswahrscheinlichkeit eines Bernoulli-Experiments unbekannt sein. Dann kann man dieses Experiment einige Male durchführen und aus den Ergebnissen den unbekanntem Parameter, wie zum Beispiel die Erfolgswahrscheinlichkeit  $p$ , mit Hilfe von Schätzfunktionen anhand dieser Stichprobe annähernd ermitteln.

Sei der Stichprobenraum  $\Omega$  eine nichtleere, höchstens abzählbare Menge und bezeichne  $\mathfrak{P} = \{P_\theta : \theta \in \Theta\}$  eine Familie von Wahrscheinlichkeitsmaßen auf  $\Omega$ , wobei  $\Theta$  die Menge der möglichen Werte der unbekanntem Parameter der Wahrscheinlichkeitsmaße ist. Da die Verteilung eines Zufallsexperiments von mehreren Faktoren abhängen kann, sei  $\Theta \subset \mathbb{R}^k$ . Sei weiter  $g : \Theta \rightarrow \tilde{\Theta}$  eine zu schätzende Funktion. Im Fall des Bernoulli-Experiments könnte dies zum Beispiel die Erfolgswahrscheinlichkeit  $p$  sein, sodass  $g(\theta) = \theta = p$  sei. Unter Umständen ist man aber nicht direkt am Parameter der Verteilung interessiert. So könnte man beispielsweise die Varianz der Bernoulli-Verteilung schätzen wollen, dann wäre  $g(\theta) = p(1 - p)$ .

**Definition 2.25 (Schätzfunktion)** (nach [18] S. 61)

Sei  $g : \Theta \rightarrow \tilde{\Theta}$  eine Funktion, dann heißt jede Funktion  $\hat{g} : \Omega \rightarrow \tilde{\Theta}$  Schätzfunktion für  $g$ .

Da man Stichproben auch als Realisierungen einer Zufallsvariablen  $X$  mit  $X(\omega) = \omega$  für  $\omega \in \Omega$  auffassen kann, kann man zur Verdeutlichung auch  $\hat{g}(X)$  schreiben.

Da Definition 2.25 auch Schätzungen zulässt, die die zu schätzenden Argumente nicht gut wiedergeben, müssen weitere Kriterien für Schätzfunktionen gegeben werden. Eine erstrebenswerte Eigenschaft einer Schätzung ist beispielsweise die Erwartungstreue gemäß Definition 2.26.

**Definition 2.26 (Erwartungstreue)** (nach [18] S. 64)

Sei  $\hat{g} : \Omega \rightarrow \tilde{\Theta}$  eine Schätzfunktion für  $g(\theta)$ . Dann heißt  $\hat{g}$  erwartungstreu, falls  $E_\theta(\hat{g}(X)) = g(\theta)$  für eine Zufallsvariable  $X$  mit Werten in  $\Omega$  und für alle  $\theta \in \Theta$  gilt.

**Beispiel 2.27**

Es sei die Stichprobe  $x = (x_1, \dots, x_n)$  eine Realisierung einer diskreten Zufallsvariablen  $X$ , deren Verteilung vom Parameter  $\theta \in \Theta$  abhängt. Weiter sei  $y = (y_1, \dots, y_n)$  eine Realisierung einer diskreten Zufallsvariablen  $Y$ , deren Verteilung ebenfalls vom Parameter  $\theta \in \Theta$  abhängt. Dann ist

1. das arithmetische Mittel  $\bar{x} := \frac{1}{n} \sum_{i=1}^n x_i$  (nach [11] S. 16) eine erwartungstreue Schätzung für  $E_\theta(X)$ ,

2. die (korrigierte) Stichprobenvarianz  $S^2(x) := \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2$  (nach [11] S. 57) eine erwartungstreue Schätzung für  $Var_\theta(X)$  und
3. die Stichprobenstandardabweichung  $S(x) := \sqrt{S^2(x)} = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2}$  (nach [11] S. 22) **keine** erwartungstreue Schätzung für die Standardabweichung  $\sqrt{Var(X)}$ . □

Eine nicht erwartungstreue Schätzung verzerrt die Ergebnisse. Ein theoretischer Wert, der den Umfang der Verzerrung angibt, ist der in Definition 2.28 angegebene Bias.

**Definition 2.28 (Bias)** (nach [18] S. 64)

Als *Bias (Verzerrung)* einer Schätzfunktion  $\hat{g} : \Omega \rightarrow \tilde{\Theta}$  für  $g(\theta)$  mit  $\theta \in \Theta$  bezeichnet man

$$B(\hat{g}) := E_\theta(\hat{g}(X)) - g(\theta), \tag{2.14}$$

wobei  $X$  eine diskrete Zufallsvariable mit Werten in  $\Omega$  sei.

In einigen Fällen kann es sein, dass es nicht oder nur mit sehr viel Aufwand möglich ist, eine erwartungstreue Schätzung zu liefern. In anderen Fällen kann es auch sein, dass eine verzerrte Schätzung aufschlussreicher ist, als eine erwartungstreue. So ist zum Beispiel der Median als verzerrte Schätzung in Beobachtungen mit vielen Ausreißern manchmal treffender als das erwartungstreue arithmetische Mittel.

Ein Wert, der die "Güte" einer Schätzung angibt, ist der mittlere quadratische Fehler (engl. mean squared error) kurz *MSE*, der in Definition 2.29 angegeben wird.

**Definition 2.29 (MSE)** (nach [18] S. 65)

Sei  $\hat{g} : \Omega \rightarrow \tilde{\Theta}$  eine Schätzfunktion für  $g(\theta)$  mit  $\theta \in \Theta$  und sei  $X$  eine diskrete Zufallsvariable mit Werten in  $\Omega$ . Der *mittlere quadratische Fehler (MSE)* von  $\hat{g}(X)$  für  $g(\theta)$  ergibt sich durch

$$MSE(\hat{g}) := E_\theta((\hat{g}(X) - g(\theta))^2). \tag{2.15}$$

Da für die Varianz einer Schätzfunktion  $\hat{g}(X)$

$$Var_\theta(\hat{g}(X)) \stackrel{(2.8)}{=} E_\theta((\hat{g}(X) - E_\theta(\hat{g}(X)))^2) = E_\theta(\hat{g}(X)^2) - (E_\theta(\hat{g}(X)))^2 \tag{2.16}$$

gilt, gilt für *MSE*( $\hat{g}$ )

$$MSE(\hat{g}) = Var_\theta(\hat{g}(X)) + B(\hat{g})^2.$$

Es ist für  $\hat{g}(X)$  mit (2.14)

$$\begin{aligned}
 MSE(\hat{g}) &\stackrel{(2.15)}{=} E_{\theta}((\hat{g}(X) - g(\theta))^2) \\
 &= E_{\theta}(\hat{g}(X)^2 - 2\hat{g}(X)g(\theta) + g(\theta)^2) \\
 &= E_{\theta}(\hat{g}(X)^2) - 2g(\theta)E_{\theta}(\hat{g}(X)) + g(\theta)^2 \\
 &= E_{\theta}(\hat{g}(X)^2) - (E_{\theta}(\hat{g}(X)))^2 + (E_{\theta}(\hat{g}(X)))^2 - 2g(\theta)E_{\theta}(\hat{g}(X)) + g(\theta)^2 \\
 &\stackrel{(2.16)}{=} Var_{\theta}(\hat{g}(X)) + (E_{\theta}(\hat{g}(X)) - g(\theta))^2 \\
 &\stackrel{(2.14)}{=} Var_{\theta}(\hat{g}(X)) + B(\hat{g})^2.
 \end{aligned}$$

Ist eine Schätzfunktion  $\hat{g}(X)$  erwartungstreu, so ist natürlich  $MSE(\hat{g}) = Var(\hat{g}(X))$ . Es ist aber durchaus möglich, dass eine verzerrte Schätzfunktion  $\hat{g}(X)$  für  $g(\theta)$  einen kleineren  $MSE$  hat, als eine unverzerrte Schätzfunktion  $\tilde{g}(X)$  für  $g(\theta)$ . Dies ist dann der Fall, wenn die Varianz von  $\hat{g}(X)$  erheblich kleiner ist, als die von  $\tilde{g}(X)$ . In diesem Fall ist diejenige Schätzung mit dem kleineren  $MSE$  vorzuziehen.

## 2.4 Statistische Tests

In diesem Abschnitt werden die Grundlagen zur statistischen Testtheorie angegeben. Statistische Tests sind Entscheidungsregeln für Zufallsvariablen. Sie stellen ein Werkzeug für die Auswertung von erhobenen Daten dar. Das folgende Beispiel 2.30 gibt eine Situation wieder, auf die ein statistischer Test angewendet werden kann.

### Beispiel 2.30

Gegeben sei eine Tierpopulation, von der man erfahren möchte, ob ein bestimmter Umwelteinfluss negative Auswirkungen auf deren Fruchtbarkeit hat. Es ist meist nicht möglich die gesamte Population als Versuchsobjekt zu verwenden. Stattdessen wählt man eine gewisse Anzahl an Individuen zufällig aus und setzt diese separiert dem Umwelteinfluss aus. Danach überprüft man deren Fruchtbarkeit. Es sei  $p_0$  die vorher bekannte Fruchtbarkeitsrate der gesamten Population, die dem Umwelteinfluss nicht ausgesetzt wurde. Es bezeichne dann  $p$  die Fruchtbarkeitsrate der dem Umwelteinfluss ausgesetzten Versuchsindividuen. Hat man  $n$  Individuen beobachtet, so ergibt sich als Beobachtung  $x = (x_1, \dots, x_n) \in \{0, 1\}^n$ . Dabei bedeutet  $x_i = 0$  für  $i \in \{1, \dots, n\}$ , dass das Individuum  $i$  unfruchtbar ist. Hat der Umwelteinfluss einen negativen Einfluss auf die Fruchtbarkeit der Population, so würde man vermuten, dass  $p < p_0$  gilt. Die Beobachtung  $x$  ist dann eine Realisierung des Zufallsvektors  $X = (X_1, \dots, X_n)$ , wobei  $X_1, \dots, X_n$  unabhängige Wiederholungen eines Bernoulli-Experiments sind. □

**Definition 2.31 (Test)** (nach [14] S. 255 und [18] S. 94)

Sei  $X : \Omega \rightarrow \mathcal{X}$  eine diskrete Zufallsvariable auf  $\Omega$  mit  $X(\omega) = x$  und es bezeichne  $\mathcal{P} = \{P_\theta : \theta \in \Theta\}$  die Menge aller möglichen Verteilungen von  $X$ . Sei  $\Theta = \Theta_0 \cup \Theta_1$  mit  $\Theta_0 \neq \emptyset$  und  $\Theta_1 \neq \Theta$ , man nennt dann die These „ $\theta \in \Theta_0$ “ Hypothese und „ $\theta \in \Theta_1$ “ Alternative. Man bezeichnet die Hypothese mit  $H_0$  und die Alternative mit  $H_1$ . Ein Test von  $\Theta_0$  gegen  $\Theta_1$  ist nun eine Entscheidungsregel  $\varphi : \mathcal{X} \rightarrow \{0, 1\}$ , die für alle  $x \in \mathcal{X}$  festlegt, ob etwas gegen die Hypothese spricht oder nicht.

Ist  $\varphi(x) = 0$ , so spricht nichts gegen die Hypothese, bei  $\varphi(x) = 1$  wird die Hypothese abgelehnt. Die Menge  $R = \{x \in \mathcal{X} \mid \varphi(x) = 1\}$  wird daher Ablehnungsbereich oder auch kritischer Bereich genannt. Man kann  $\varphi$  immer als  $\varphi(x) = 1_R(x)$  schreiben, wobei  $1_R(x) = 1$  für  $x \in R$  und  $1_R(x) = 0$  für  $x \notin R$  ist.

Nun kann ein Test natürlich auch zu falschen Entscheidungen führen. Es gibt dabei zwei Arten von Fehlern, die auftreten können. Ist  $\varphi(x) = 1$ , aber  $\theta \in \Theta_0$ , so spricht man von dem Fehler erster Art beziehungsweise  $\alpha$ -Fehler. Ist hingegen  $\varphi(x) = 0$ , aber  $\theta \in \Theta_1$ , so ist von dem Fehler zweiter Art beziehungsweise  $\beta$ -Fehler die Rede. Die Hypothese wird auch Nullhypothese genannt, da diejenigen Verteilungen als Hypothese gewählt werden, die der Aussage entsprechen, dass der untersuchte Faktor keinen Einfluss auf die Versuchsindividuen hat. Mit dieser Wahl der Nullhypothese kann dann die Fehlerwahrscheinlichkeit erster Art klein gehalten werden, sodass ein Test seltener fälschlicherweise die Hypothese ablehnt und somit einen Einfluss aufzeigt, der nicht vorhanden ist.

**Definition 2.32 (Gütefunktion)** (nach [14] S. 255)

Ist  $\varphi : \mathcal{X} \rightarrow \{0, 1\}$  ein Test für die Hypothese  $\theta \in \Theta_0$ , dann heißt die Funktion

$$G_\varphi : \Theta \rightarrow [0, 1], \quad \text{mit } G_\varphi(\theta) = E_\theta(\varphi) = P_\theta(\varphi(X) = 1) \quad (2.17)$$

Gütefunktion des Tests  $\varphi$ .

**Definition 2.33 (Test zum Niveau  $\alpha$ )** (nach [14] S. 255)

Sei  $\varphi : \mathcal{X} \rightarrow \{0, 1\}$  ein Test für die Hypothese  $\theta \in \Theta_0$ . Gilt

$$G_\varphi(\theta) \leq \alpha \quad \text{für alle } \theta \in \Theta_0 \quad (2.18)$$

für ein  $\alpha \in [0, 1]$ , so nennt man  $\varphi$  einen Test zum Niveau  $\alpha$ . Gilt außerdem

$$G_\varphi(\theta) \geq \alpha \quad \text{für alle } \theta \in \Theta_1 \quad (2.19)$$

so heißt  $\varphi$  unverfälscht.

Bei einem Test zum Niveau  $\alpha$  ist der Fehler erster Art also höchstens  $\alpha$ . Übliche Werte für  $\alpha$  sind  $\alpha = 0,05$  oder  $\alpha = 0,01$ . Das bedeutet, in weniger als 5% beziehungsweise 1% der Fälle würde man sich fälschlicherweise gegen eine Hypothese entscheiden.

**Definition 2.34 (Teststatistik/Prüfgröße)** (nach [18] S. 94)

Sei eine Funktion  $T : \mathcal{X} \rightarrow \mathbb{R}$  so gewählt, dass kleine (oder große) Werte von  $T$  gegen die Hypothese  $\theta \in \Theta_0$  sprechen. Man nennt  $T$  dann Teststatistik und der Ablehnungsbereich  $R$  lässt sich als  $R = \{x \in \mathcal{X} \mid T(x) \leq t\}$  (beziehungsweise  $R = \{x \in \mathcal{X} \mid T(x) \geq t\}$ ) schreiben, wobei  $t$  als kritischer Wert bezeichnet und die Hypothese für  $T(x) \leq t$  (oder  $T(x) \geq t$ ) verworfen wird. Das Ergebnis einer Teststatistik, angewandt auf eine gegebene Stichprobe, nennt man Prüfgröße.

**Beispiel 2.35** (Fortsetzung von Beispiel 2.30)

Angenommen für die schon bekannte Fruchtbarkeitsrate der gesamten Tierpopulation gilt  $p_0 = 0.9$  und es sei  $n = 20$  die Anzahl der Versuchsindividuen, die einem Umwelteinfluss ausgesetzt wurden. Es sei  $x = (x_1, \dots, x_{20})$  mit  $x_i \in \{0, 1\}$  für  $i \in \{1, \dots, 20\}$ . Offensichtlich hat man es hier mit einem Binomialmodell mit  $\mathcal{X} = \{0, 1, \dots, 20\}$  und  $P_\theta = \mathcal{B}_{20, \theta}$  (vergleiche Definition 2.19) zu tun. Die Nullhypothese ist dann  $H_0 : p \geq p_0$ , das heißt, der Umwelteinfluss hat keinen schädlichen Einfluss auf die Fruchtbarkeit der Tiere, und die Alternative ist folglich  $H_1 : p < p_0$ , das heißt, der Umwelteinfluss hat einen schädlichen Einfluss auf die Fruchtbarkeit. Daraus ergibt sich  $\Theta = [0, 1]$ ,  $\Theta_0 = [0.9, 1]$  und  $\Theta_1 = [0, 0.9)$ . Als Teststatistik  $T$  ergibt sich  $T(x) = \sum_{i=1}^{20} x_i$ . Nach Folgerung 2.20 besitzt  $\sum_{i=1}^{20} X_i$  eine  $\mathcal{B}_{20, p}$ -Verteilung. Nimmt man als Niveau  $\alpha = 0.05$  an, sei also der Fehler erster Art maximal 5%, so ist der kritische Wert  $t$ , für den

$$G_\varphi(0.9) = E_{0.9}(\varphi) = P_{0.9}(\varphi(X) = 1) = P_{0.9}\left(\sum_{i=1}^{20} X_i < t\right) \leq 0.05$$

gilt, gesucht. Mit Hilfe der Binomialverteilung ergibt sich nun 15 als maximaler Wert für  $t$ . Es lässt sich nämlich

$$P_{0.9}\left(\sum_{i=1}^{20} X_i < 15\right) \approx 0.04317$$

ermitteln.

Die Hypothese würde also abgelehnt und somit dem Umwelteinfluss ein negativer Einfluss auf die Fruchtbarkeit zugesprochen werden, wenn weniger als 15 Tiere fruchtbar wären. □

In der Praxis wird bei statistischen Tests meistens der p-Wert (engl. p-value) angegeben.

**Definition 2.36 (p-Wert)** (nach [9] S. 253)

Sei eine Stichprobe  $x = (x_1, \dots, x_n)$  eine Realisierung des diskreten,  $\mathcal{X}$ -wertigen Zufallsvektors  $X = (X_1, \dots, X_n)$  auf  $\Omega$  und sei  $T : \mathcal{X} \rightarrow \mathbb{R}$  eine Teststatistik für das Testproblem  $H_0 : \theta \in \Theta_0$  gegen  $H_1 : \theta \in \Theta_1$  mit  $\Theta = \Theta_0 \cup \Theta_1$ . Sei weiter  $\varphi(x) = 1_{\{T(x) \leq c\}}(x)$  ein Test zum Niveau  $\alpha$  für  $H_0$  gegen  $H_1$ , wobei  $c$  der kleinste Wert sei, für den  $P_\theta(T(X) \leq c) \leq \alpha$  mit  $\theta \in \Theta_0$  gilt, dann ist

$$p^*(x) = \sup_{\theta \in \Theta_0} P_\theta(T(X) \leq T(x)) \quad (2.20)$$

der p-Wert der Stichprobe  $x$  zum Test  $\varphi$ .

Sprechen im Sinne von Definition 2.36 große Werte einer Prüfgröße  $T$  gegen die Nullhypothese, so gibt der p-Wert an, mit welcher Wahrscheinlichkeit eine ebenso große und größere Prüfgröße, wie die der interessierenden Stichprobe, bei richtiger Nullhypothese erwartet werden kann. Die Nutzung des p-Wertes ist nur sinnvoll, wenn vor der Durchführung eines Tests ein geeignetes Signifikanzniveau  $\alpha$  festgelegt wurde. Ist der p-Wert  $p^*$  größer als das vorher festgelegte Signifikanzniveau  $\alpha$ , so wird  $H_0$  nicht abgelehnt. Ist hingegen der p-Wert kleiner oder gleich  $\alpha$ , so wird  $H_0$  abgelehnt, da die Stichprobe dann signifikant gegen  $H_0$  spricht.

**Beispiel 2.37** (Fortsetzung von Beispiel 2.35)

Angenommen, es wurde beobachtet, dass nur 12 Tiere fruchtbar sind, dann ist

$$p^*(x) = \sup_{\theta \in \Theta_0} P_\theta(T(X) < t) = P_{0.9}(T(X) < 12) \approx 0.00042 \quad (2.21)$$

Es ist  $0.00042 < 0.05$  und somit kann die Nullhypothese  $H_0 : p \geq p_0$  mittels dieses Test abgelehnt werden. Es kann also geschlossen werden, dass bei der vorliegenden Stichprobe ein negativer Effekt auf die Fruchtbarkeit vorliegt.  $\square$

## 2.5 R-Funktionen

R-Funktionen sind kombinatorische Funktionen. Ein Spezialfall der R-Funktionen ist der Binomialkoeffizient. Mit Hilfe der R-Funktionen wird es in Kapitel 5 möglich sein, ein Stichprobensystem zu entwickeln, mit dem Stichproben von Binärmatrizen spaltenweise generiert werden können. Sie bilden somit eine sehr wichtige Grundlage dieser Arbeit.

Im Folgenden sei stets  $\Omega = \{1, \dots, N\}$  und es bezeichne  $\mathbb{N}_0$  die Menge der natürlichen Zahlen einschließlich der Null und  $\mathbb{N} := \mathbb{N}_0 \setminus \{0\}$ .

**Definition 2.38 (R-Funktionen)** (nach [5] S. 459)

Sei  $A \subset \Omega$  eine nichtleere Teilmenge und  $k \in \mathbb{N}_0$ . Für  $x = (x_{i_1}, \dots, x_{i_{|A|}}) \in \mathbb{R}^{|A|}$  ist die R-Funktion für  $k \in \{1, \dots, |A|\}$  gegeben durch

$$R(k, A) = \sum_{B \subset A, |B|=k} \left( \prod_{i \in B} x_i \right). \quad (2.22)$$

Für  $k = 0$  sei  $R(0, A) = 1$  und für  $k > |A|$  sei  $R(k, A) = 0$ .

Gilt in Definition 2.38, dass  $x_i = 1$  für alle  $i \in \{1, \dots, |A|\}$  ist, so erhält man den Spezialfall  $R(k, A) = \binom{|A|}{k}$ , die Anzahl aller  $k$ -elementigen Teilmengen von  $A$ , den Binomialkoeffizienten.

**Eigenschaften 2.39** (nach [5] S. 459)

Für eine beliebige nichtleere Menge  $A \subset \Omega$  und eine natürliche Zahl  $k \in \{1, \dots, |A|\}$  gilt für beliebige  $x \in \mathbb{R}^{|A|}$

$$\sum_{i \in A} (x_i R(k-1, A \setminus \{i\})) = k R(k, A). \quad (2.23)$$

□

BEWEIS

In [5] ist kein Beweis angegeben. Die Tatsache folgt aber direkt durch einige Umformungen. Es gilt

$$\begin{aligned} \sum_{i \in A} (x_i R(k-1, A \setminus \{i\})) &= \sum_{i \in A} x_i \sum_{\substack{B \subset A \setminus \{i\} \\ |B|=k-1}} \left( \prod_{j \in B} x_j \right) \\ &= \sum_{i \in A} \sum_{\substack{B \subset A \setminus \{i\} \\ |B|=k-1}} \left( x_i \prod_{j \in B} x_j \right). \end{aligned}$$

Sei nun  $B' \subset A$  mit  $|B'| = k$ , dann ist

$$|\{B \subset B' \mid |B| = k-1\}| = k. \quad (2.24)$$

Somit gilt

$$\begin{aligned} \sum_{i \in A} \sum_{\substack{B \subset A \setminus \{i\} \\ |B|=k-1}} \left( x_i \prod_{j \in B} x_j \right) &= k \sum_{\substack{B' \subset A \\ |B'|=k}} \left( \prod_{i \in B'} x_i \right) \\ &= k R(k, A). \end{aligned} \quad \blacksquare$$

### 2.5.1 Newton's Identitäten

Die naheliegendste Möglichkeit  $R(k, A)$  für  $(x_{i_1}, \dots, x_{i_{|A|}}) \in \mathbb{R}^{|A|}$  zu ermitteln, besteht in der komponentenweisen Auswertung. Manchmal benötigt man allerdings sämtliche R-Funktionen  $R(k, A)$  für alle  $k \in \{1, \dots, |A|\}$ . Diese R-Funktionen können dann effektiver rekursiv erschlossen werden. Eine Möglichkeit zur rekursiven Berechnung bieten Newton's Identitäten. Diese gehen auf Isaac Newton zurück und können einen Zusammenhang zwischen den R-Funktionen und den Potenzsummen im Sinne von Definition 2.40 von beliebigen  $x_1, \dots, x_{|A|} \in \mathbb{R}$  herstellen.

**Definition 2.40 (Potenzsumme)** (ähnlich wie in [5] S. 460)

Seien  $A = \{1, \dots, N\} \subset \mathbb{N}$  und  $x_1, \dots, x_N \in \mathbb{R}$ . Dann bezeichne

$$T(k, A) = \sum_{i \in A} x_i^k \quad (2.25)$$

die  $k$ -te Potenzsumme von  $x = (x_1, \dots, x_N)$  für  $k \in \mathbb{N}$ .

Mit  $A$  und  $x$  wie in Definition 2.40 und  $N \geq k \in \mathbb{N}$  ist

$$R(k, A) = \sum_{1 \leq i_1 < \dots < i_k \leq N} \prod_{j=1}^k x_{i_j} \text{ für } i_1, \dots, i_k \in A. \quad (2.26)$$

**Satz 2.41** (nach [23] S. 749)

Sei  $A = \{1, \dots, N\}$ , dann gilt für  $x_1, \dots, x_N \in \mathbb{R}$  und für alle  $N \geq k \in \mathbb{N}$

$$0 = kR(k, A) + \sum_{i=1}^k (-1)^i T(i, A) R(k-i, A). \quad (2.27)$$

Diese Gleichungen nennt man Newton's Identitäten.

BEWEIS (nach [23] S. 750)

Man setze zunächst für  $x_1, \dots, x_N \in \mathbb{R}$ ,  $k \in \{2, \dots, N\}$ ,  $i \in \{1, \dots, k-1\}$  und  $h \in \{1, \dots, k-1\}$

$$U(k-i, h) = \sum_{\substack{1 \leq j, j_1, \dots, j_{k-i} \leq N \\ j_1 < \dots < j_{k-i} \\ j \neq j_1, \dots, j_{k-i}}} x_{j_1} \cdot \dots \cdot x_{j_{k-i}} \cdot x_j^h \quad (2.28)$$

und für  $k=1$

$$U(0, h) = \sum_{j=1}^N x_j^h \stackrel{(2.25)}{=} T(h, A). \quad (2.29)$$

Da es für  $j \in \{1, \dots, N\}$  genau  $k$  Möglichkeiten gibt sich in die Menge

$$\{j_1, \dots, j_{k-1} \mid j_l \in A \text{ mit } l \in \{1, \dots, k-1\}\}$$

einzuordnen, gilt

$$\begin{aligned} U(k-1, 1) &= \sum_{\substack{1 \leq j_1, \dots, j_{k-1} \leq N \\ j_1 < \dots < j_{k-1} \\ j \neq j_1, \dots, j_{k-1}}} x_{j_1} \cdot \dots \cdot x_{j_{k-1}} \cdot x_j \\ &= k \sum_{1 \leq i_1 < \dots < i_k \leq N} x_{i_1} \cdot \dots \cdot x_{i_k} \\ &\stackrel{(2.26)}{=} kR(k, A). \end{aligned} \tag{2.30}$$

Es gilt weiter

$$\begin{aligned} R(k-i, A)T(i, A) &= \sum_{1 \leq j_1 < \dots < j_{k-i} \leq N} x_{j_1} \cdot \dots \cdot x_{j_{k-i}} \sum_{j=1}^N x_j^i \\ &= \sum_{\substack{1 \leq j_1, \dots, j_{k-i} \leq N \\ j_1 < \dots < j_{k-i}}} x_{j_1} \cdot \dots \cdot x_{j_{k-i}} \cdot x_j^i. \end{aligned} \tag{2.31}$$

Die Summe in (2.31) kann man nun in zwei Teilsommen aufteilen. Die eine für den Fall, dass  $j \neq j_l$  für alle  $l \in \{1, \dots, k-i\}$  ist. Diese Teilsomme ist gleich  $U(k-i, i)$  (siehe (2.28)). Die andere Teilsomme steht für den Fall, dass  $j = j_l$  für ein  $l \in \{1, \dots, k-i\}$  und ist gerade  $U(k-i-1, i+1)$ , sodass nun

$$R(k-i, A)T(i, A) = U(k-i, i) + U(k-i-1, i+1) \quad \text{für } 1 \leq i \leq k \tag{2.32}$$

gilt. Es können nun  $k+1$  Gleichungen aufgestellt werden.

$$\begin{aligned} 0) \quad & kR(k, A) \stackrel{(2.30)}{=} U(k-1, 1) \\ 1) \quad & R(k-1, A)T(1, A) \stackrel{(2.32)}{=} U(k-1, 1) + U(k-2, 2) \\ 2) \quad & R(k-2, A)T(2, A) \stackrel{(2.32)}{=} U(k-2, 2) + U(k-3, 3) \\ & \dots \\ k-2) \quad & R(2, A)T(k-2, A) \stackrel{(2.32)}{=} U(2, k-2) + U(1, k-1) \\ k-1) \quad & R(1, A)T(k-1, A) \stackrel{(2.32)}{=} U(1, k-1) + U(0, k) \\ k) \quad & T(k, A) \stackrel{(2.29)}{=} U(0, k). \end{aligned}$$

Multipliziert man nun für alle  $i \in \{0, \dots, k\}$  die  $i$ -te dieser Gleichungen mit  $(-1)^i$

und addiert sämtliche Gleichungen, so erhält man

$$\begin{aligned}
 kR(k, A) + \sum_{i=1}^k (-1)^i R(k-i, A)T(i, A) &= U(k-1, 1) - U(k-1, 1) \\
 &\quad - U(k-2, 2) + U(k-2, 2) \\
 &\quad + U(k-3, 3) - U(k-3, 3) \\
 &\quad \dots \\
 &\quad + (-1)^{k-2}U(1, k-1) + (-1)^{k-1}U(1, k-1) \\
 &\quad + (-1)^{k-1}U(0, k) + (-1)^kU(0, k) \\
 &= 0,
 \end{aligned}$$

was den Beweis von Satz 2.41 komplettiert. ■

### 2.5.2 Rekursionsformeln für R-Funktionen

Mit Hilfe von Satz 2.41 lässt sich nun die folgende Rekursionsformel für  $R(k, A)$  angeben.

**Satz 2.42** (nach [5] S. 460)

Sei  $A = \{1, \dots, N\} \subset \mathbb{N}$ ,  $B \subset A$  und  $x_1, \dots, x_N \in \mathbb{R}$  beliebig, dann bezeichne ähnlich wie in (2.25)  $T(i, B) = \sum_{j \in B} x_j^i$  für alle  $i \in \mathbb{N}$ . Es gilt dann

$$R(k, B) = \frac{1}{k} \sum_{i=1}^k (-1)^{i+1} T(i, B)R(k-i, B) \quad (2.33)$$

für alle  $k \in \{1, \dots, |B|\}$ .

BEWEIS

Da für die Gültigkeit von Satz 2.41 das genaue Aussehen von  $x_1, \dots, x_N$  unerheblich ist, gilt Satz 2.41 natürlich auch für beliebige nichtleere Teilmengen von  $A$ . Es ist somit nach (2.27)

$$0 = kR(k, B) + \sum_{i=1}^k (-1)^i T(i, B)R(k-i, B)$$

und somit

$$R(k, B) = \frac{1}{k} \sum_{i=1}^k (-1)^{i+1} T(i, B)R(k-i, B)$$

für alle  $k \in \{1, \dots, |B|\}$ . ■

**Beispiel 2.43**

Zur Verdeutlichung der durch die Gleichung (2.33) gegebenen Rekursionsformel sei  $B = \{1, \dots, 4\} \subset \mathbb{N}$ ,  $k = 2$  und  $x = (x_1, x_2, x_3, x_4) = (2, 2, 2, 2)$ . Es ist dann

$$\begin{aligned} R(1, B) &\stackrel{(2.33)}{=} \frac{1}{1} \sum_{i=1}^1 (-1)^{i+1} T(i, B) R(1-i, B) \\ &= T(1, B) R(0, B) = \sum_{i=1}^4 x_i \cdot 1 = 8 \end{aligned}$$

und weiter

$$\begin{aligned} R(2, B) &\stackrel{(2.33)}{=} \frac{1}{2} \sum_{i=1}^2 (-1)^{i+1} T(i, B) R(2-i, B) \\ &= \frac{1}{2} (T(1, B) R(1, B) - T(2, B) R(0, B)) \\ &= \frac{1}{2} \left( \left( \sum_{i=1}^4 x_i \right) 8 - \left( \sum_{i=1}^4 x_i^2 \right) 1 \right) \\ &= \frac{1}{2} (64 - 16) = 24. \end{aligned} \quad \square$$

Es lässt sich auch noch eine weitere Rekursionsformel angeben.

**Satz 2.44** (nach [4] S. 878)

Sei  $A = \{1, \dots, N\} \subset \mathbb{N}$ ,  $B \subset A$  und  $x_1, \dots, x_N \in \mathbb{R}$ . Für  $1 \leq k \leq |B|$  gilt dann

$$R(k, B) = R(k, B \setminus \{k\}) + x_k R(k-1, B \setminus \{k\}). \quad (2.34)$$

BEWEIS

In [4] ist kein Beweis angegeben. (2.34) folgt aber direkt aus der Definition von  $R(k, B)$ . Es ist

$$R(k, B) \stackrel{(2.22)}{=} \sum_{\substack{B' \subset B \\ |B'|=k}} \left( \prod_{i \in B'} x_i \right). \quad (2.35)$$

Diese Summe lässt sich nun in zwei Teilsommen aufteilen. Es ist dann

$$\sum_{\substack{B' \subset B \\ |B'|=k}} \left( \prod_{i \in B'} x_i \right) = \sum_{\substack{B' \subset B \setminus \{k\} \\ |B'|=k}} \left( \prod_{i \in B'} x_i \right) + \sum_{\substack{B' \subset B \setminus \{k\} \\ |B'|=k-1}} \left( x_k \prod_{i \in B'} x_i \right). \quad (2.36)$$

Die erste Teilsomme beschreibt dabei die Summe über alle Teilmengen  $B' \subset B$ , die  $k$  nicht enthalten, und die zweite Teilsomme beschreibt letztendlich die Summe über

diejenigen Teilmengen, die  $k$  enthalten. Insgesamt folgt somit

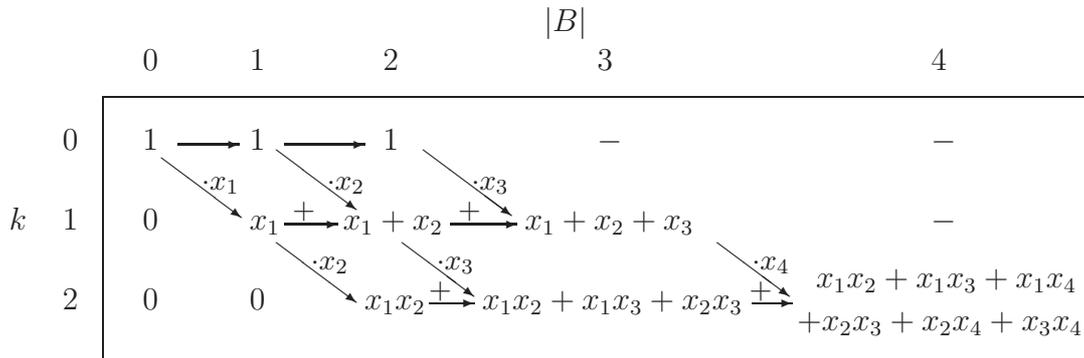
$$\begin{aligned}
 R(k, B) &\stackrel{(2.35)}{=} \sum_{\substack{B' \subset B \\ |B'|=k}} \left( \prod_{i \in B'} x_i \right) \\
 &\stackrel{(2.36)}{=} \sum_{\substack{B' \subset B \setminus \{k\} \\ |B'|=k}} \left( \prod_{i \in B'} x_i \right) + \sum_{\substack{B' \subset B \setminus \{k\} \\ |B'|=k-1}} \left( x_k \prod_{i \in B'} x_i \right) \\
 &= R(k, B \setminus \{k\}) + x_k R(k-1, B \setminus \{k\}). \quad \blacksquare
 \end{aligned}$$

**Beispiel 2.45** (nach [4] S. 879)

Zur Verdeutlichung der durch die Gleichung (2.34) gegebenen Rekursionsformel sei  $B = \{1, \dots, 4\} \subset \mathbb{N}$ ,  $k = 2$  und  $x = (x_1, \dots, x_4) = (2, 2, 2, 2)$ . In Abbildung 2.1 ist der Rekursionsverlauf angegeben. Daraus ergibt sich dann

$$R(k, B) = x_1x_2 + x_1x_3 + x_1x_4 + x_2x_3 + x_2x_4 + x_3x_4 = 24. \quad \square$$

Man vergleiche dieses Ergebnis mit dem Ergebnis aus Beispiel 2.43.



**Abbildung 2.1:** Rekursionsschema (2.34) für  $|B| = 4$  und  $k = 2$ . (Nach [4] S. 879)

Beide Rekursionsmethoden haben Vor- und Nachteile. So müssen für große Werte von  $|B|$  und  $k$  bei (2.33) insgesamt weniger arithmetische Operationen durchgeführt werden um  $R(k, B)$  zu bestimmen. Allerdings ist die Effizienz von (2.33) durch die Genauigkeit des verwendeten Computerprogramms beschränkt. In der Regel werden Werte maximal mit einer Genauigkeit von 16 Stellen dargestellt. Die Potenzsummen  $T(k, B)$  nehmen schnell sehr hohe Werte an. Werden solch hohe Werte addiert beziehungsweise subtrahiert, so wird das Ergebnis verfälscht. Demnach ist für sehr hohe Werte von  $k$  die Rekursionsformel (2.34) vorzuziehen. In der Implementierung des in dieser Arbeit vorgestellten Algorithmus werden deshalb beide Rekursionsformeln jeweils abhängig von dem Wert  $k$  angewendet werden.

## 3 Eigenschaften von Binärmatrizen

Es folgt zunächst eine formale Definition einer Binärmatrix.

### Definition 3.1 (Binärmatrix)

*Es bezeichne  $A \in \{0, 1\}^{m \times n}$  eine  $m \times n$  Matrix mit Einträgen aus der Menge  $\{0, 1\}$ . Man nennt  $A$  dann Binärmatrix.*

Binärmatrizen bilden eine wichtige Struktur für die Betrachtung ökologischer Daten. Oftmals werden in der Ökologie Daten erfasst, die die An- oder Abwesenheit mehrerer unterschiedlicher Arten von Lebewesen in mehreren Lebensräumen beschreiben. Aufgrund dieser Daten sollen dann Rückschlüsse auf Beziehungen zwischen den Arten möglich sein. Solche Daten lassen sich leicht mit Hilfe von Binärmatrizen ausdrücken. Es entspricht dann jeweils eine Zeile einer beobachteten Art und eine Spalte einem beobachteten Lebensraum.

Es entsteht bei  $m$  erfassten Arten und  $n$  erfassten Lebensräumen eine  $m \times n$  Matrix. Wurde Art  $i$  in Lebensraum  $j$  gesichtet, so wird in der Matrix in Zeile  $i$  und Spalte  $j$  eine Eins eingetragen. Andernfalls trägt man eine Null ein. Man nennt solche Binärmatrizen auch Presence-Absence Matrizen.

Um mit Hilfe des in Kapitel 6 folgenden Verfahrens solche ökologischen Daten sinnvoll auswerten zu können, müssen weitere „zufällige“ Binärmatrizen erzeugt werden. Diese werden dann mit der Original-Matrix verglichen. Dabei gibt es mehrere Ansätze solche Matrizen zu generieren. In dieser Arbeit werden ausschließlich Matrizen generiert, deren Zeilen- und Spaltensummen mit denen der Original-Matrix übereinstimmen. Dies hat den ökologischen Hintergrund, dass dadurch natürliche Unterschiede zwischen den Arten und Lebensräumen erhalten bleiben. So bietet ein vergleichsweise großer Lebensraum mehr Arten Platz als ein kleiner und unter den Lebewesen gibt es wiederum Spezialisten, die nur wenige Lebensräume besiedeln, sowie Generalisten, die viele Lebensräume besiedeln können.

Die erste in diesem Kapitel vorgestellte Eigenschaft von Binärmatrizen wird für den Beweis der zweiten Eigenschaft von Binärmatrizen benötigt. Des Weiteren spielt diese erste Eigenschaft eine entscheidende Rolle in einem anderen Ansatz zur Generierung von Binärmatrizen mit festen Zeilen- und Spaltensummen. Die zweite Eigenschaft ist dann entscheidend für das in Kapitel 6 folgende Verfahren.

**Definition 3.2** ( $\Sigma_{rc}$ ) (nach [7] S. 110)

Seien  $m, n \in \mathbb{N}$ ,  $r = (r_1, \dots, r_m) \in \{0, \dots, n\}^m$  und  $c = (c_1, \dots, c_n) \in \{0, \dots, m\}^n$ . Es bezeichnet dann  $\Sigma_{rc}$  die Menge aller  $m \times n$  Binärmatrizen mit Zeilensummenvektor (ZSV)  $r$  und Spaltensummenvektor (SSV)  $c$ . Das heißt für alle Matrizen  $A = (a_{i,j})_{i=1, \dots, m; j=1, \dots, n} \in \Sigma_{rc}$  gilt

$$\sum_{j=1}^n a_{i,j} = r_i \text{ für alle } i \in \{1, \dots, m\}$$

und

$$\sum_{i=1}^m a_{i,j} = c_j \text{ für alle } j \in \{1, \dots, n\}.$$

**Beispiel 3.3**

Eine relativ kleine Binärmatrix basierend auf 11 Schneckenarten und deren Verteilung auf 6 Bahamainseln entspricht den in Tabelle 3.1 aufgefassten Beobachtungen aus [22] und ist durch

$$A = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}^T$$

gegeben. Die Menge aller Binärmatrizen mit den gleichen Zeilen- und Spaltensummen wie  $A$  würde dann mit  $\Sigma_{rc}$  bezeichnet, wobei der SSV  $r = (10, 10, 3, 5, 5, 5)$  und der ZSV  $c = (5, 1, 1, 6, 5, 5, 2, 6, 2, 2, 3)$  ist. □

Insel	Art
Jewfish Cay	$A, B, D, E, F, G, H, I, J, K$
Culmer's Cay	$A, C, D, E, F, G, H, I, J, K$
Duck Cay	$A, D, H$
Rum Cay	$A, D, E, F, H$
Guana Cay	$A, D, E, F, H$
Hawksbill Rock	$D, E, F, H, K$

**Tabelle 3.1:** Verteilung von 11 Schneckenarten auf 6 Bahamainseln. Die Schneckenarten sind wie folgt bezeichnet:  $A \hat{=}$  Cerion,  $B \hat{=}$  Drepanotrema sp. 1,  $C \hat{=}$  Drepanotrema sp. 2,  $D \hat{=}$  Hemitrochus varians,  $E \hat{=}$  Opisthosiphon bahamensis,  $F \hat{=}$  Oleacina solidula,  $G \hat{=}$  Physa cubensis,  $H \hat{=}$  Plagiopycha duclosiana,  $I \hat{=}$  Polygyra plana,  $J \hat{=}$  Succinea barbadensis,  $K \hat{=}$  Urocuptis bahamensis. (Nach [22] S. 60)

### 3.1 2x2-swaps

**Definition 3.4** ( $2 \times 2$ -swap) (nach [29] S. 375)

Sei

$$A_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \text{und} \quad A_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \quad (3.1)$$

Eine Transformation der Einträge einer Matrix  $A \in \{0,1\}^{m \times n}$ , die eine  $2 \times 2$  Untermatrix von  $A$  der Form  $A_1$  in  $A_2$  beziehungsweise eine  $2 \times 2$  Untermatrix von  $A$  der Form  $A_2$  in  $A_1$  umwandelt und dabei alle anderen Einträge von  $A$  unverändert lässt, nennt man einen  $2 \times 2$ -swap. Dabei muss die  $2 \times 2$  Untermatrix nicht aus unmittelbar benachbarten Zeilen und Spalten bestehen, sondern kann sich aus beliebigen Schnittpunkten von je 2 Zeilen und 2 Spalten zusammensetzen. Ein solcher  $2 \times 2$ -swap verändert die Zeilen- und Spaltensummen von  $A$  nicht.

Damit kann nun die erste angekündigte Eigenschaft von Binärmatrizen angegeben werden.

**Satz 3.5** (nach [29] S. 375)

Seien  $A \in \{0,1\}^{m \times n}$  und  $A' \in \{0,1\}^{m \times n}$  aus  $\Sigma_{rc}$ . Dann lässt sich  $A$  durch endlich viele  $2 \times 2$ -swaps in  $A'$  transformieren und ebenso auch  $A'$  in  $A$ .

BEWEIS (nach [29] S. 375)

Der Beweis erfolgt durch Induktion nach der Zeilenanzahl  $m \in \mathbb{N}$ . Für  $A = A'$  ist die minimale Anzahl der benötigten  $2 \times 2$ -swaps natürlich Null. Es sei im Folgenden  $A \neq A'$ .

Für  $m = 2$  gilt die Behauptung offensichtlich, denn die einzige Möglichkeit für  $A$  und  $A'$ , sich bei gleichen Zeilen- und Spaltensummen zu unterscheiden, besteht im Austauschen zweier oder mehrerer  $2 \times 2$  Untermatrizen der Form  $A_1$  oder  $A_2$  aus (3.1).

Sei nun  $m > 2$ . Nach Induktionsvoraussetzung gilt die Behauptung für  $(m - 1) \times n$  Binärmatrizen. Kann man nun die erste Zeile von  $A$  durch  $2 \times 2$ -swaps in die erste Zeile von  $A'$  umwandeln, so kann man die Induktionsvoraussetzung auf die übrigen  $m - 1$  Zeilen von  $A$  anwenden und die Behauptung ist bewiesen.

Angenommen es ist nicht möglich, die erste Zeile von  $A$  durch endlich viele  $2 \times 2$ -swaps in die erste Zeile von  $A'$  umzuwandeln. Es wird im Folgenden gezeigt, dass dieser Zustand nie eintritt.

Es bezeichne  $A^*$  die durch die möglichen Vertauschungen aus  $A$  entstandene Matrix. Man kann nun zur Vereinfachung die Spalten von  $A^*$  und  $A'$  neu anordnen. Dabei werden jeweils die gleichen Spalten in den Matrizen getauscht.

Es habe dann die erste Zeile von  $A^*$  die Gestalt

$$(1_r, 0_s, 1_t, 0_t) \tag{3.2}$$

und die erste Zeile von  $A'$  die Gestalt

$$(1_r, 0_s, 0_t, 1_t), \tag{3.3}$$

wobei  $1_k$  einen Zeilenvektor aus  $k$  Einsen und  $0_k$  einen Zeilenvektor aus  $k$  Nullen für  $k \in \{r, s, t\}$  mit  $r, s, t \in \mathbb{N}$  bezeichnet. Wurden die ersten  $r + s$  Einträge der ersten Zeile von  $A$  erfolgreich umgewandelt worden, so bezeichne  $A_{m-1}^*$  und  $A'_{m-1}$  jeweils die letzten  $m - 1$  Zeilen von  $A^*$  und  $A'$ . Es wird nun gezeigt, dass ein weiterer  $2 \times 2$ -swap möglich ist.

Die Zeilensummen von  $A_{m-1}^*$  und  $A'_{m-1}$  sind nach Voraussetzung gleich und auch die Spaltensummen der jeweils entsprechenden ersten  $r + s$  Spalten von  $A_{m-1}^*$  und  $A'_{m-1}$  sind gleich. Sei nun  $\alpha_i$  die  $(r + s + i)$ -te und  $\beta_i$  die  $(r + s + t + i)$ -te Spalte von  $A_{m-1}^*$ , wobei  $i \in \{1, \dots, t\}$  ist. Analog bezeichne  $\alpha'_i$  und  $\beta'_i$  die jeweiligen Spalten von  $A'_{m-1}$ . Des Weiteren seien  $a_i, b_i, a'_i, b'_i$  die Spaltensummen von  $\alpha_i, \beta_i, \alpha'_i, \beta'_i$ . Es kann unter der Annahme, dass kein weiterer  $2 \times 2$ -swap möglich ist, in der  $j$ -ten Zeile von  $A_{m-1}^*$  nicht gleichzeitig eine Null in der  $\alpha_i$ -ten Spalte und eine Eins in der  $\beta_i$ -ten auftreten, denn sonst könnte man, wie aus (3.2) und (3.3) ersichtlich ist, einen weiteren  $2 \times 2$ -swap durchführen, was der Annahme widerspräche.

Daraus folgt nun aber, dass  $a_i \geq b_i$  sein müßte, und, da die Spaltensummen von  $A'$  und  $A^*$  gleich sind, gälte, wie wiederum aus (3.2) und (3.3) ersichtlich ist,  $a'_i = a_i + 1$  und  $b'_i = b_i - 1$ . Das bedeutete aber auch

$$a'_i - b'_i = a_i - b_i + 2 \geq 2.$$

Daraus würde aber  $a_i < b_i$  folgen, was im Widerspruch zu  $a_i \geq b_i$  steht.

Es folgt, dass in mindestens einer Zeile von  $A'_{m-1}$  in der  $\alpha'_i$ -ten Spalte eine Eins und gleichzeitig in der  $\beta'_i$ -ten Spalte eine Null steht. Vertauscht man nun für alle  $i \in \{1, \dots, t\}$  die Einsen mit den Nullen in dieser Zeile, so erhält man die Matrix  $\tilde{A}_{m-1}$ , die die gleichen Zeilen- und Spaltensummen wie  $A_{m-1}^*$  hat. Nach Induktionsvoraussetzung kann man nun  $A_{m-1}^*$  durch endlich viele  $2 \times 2$ -swaps in  $\tilde{A}_{m-1}$  umwandeln. Angewandt auf  $A^*$  bewirken diese  $2 \times 2$ -swaps aber aufgrund der neu entstandenen Anordnung der Einsen und Nullen wiederum, dass es möglich ist, weitere  $2 \times 2$ -swaps auf  $A^*$  anzuwenden, um die erste Zeile von  $A^*$  an die erste Zeile von  $A'$  anzugleichen. So folgt dann induktiv die Behauptung. ■

## 3.2 Der Satz von Gale-Ryser und seine Anwendung

Bei dem in Kapitel 6 folgenden Verfahren handelt es sich um einen speziellen Fill-Algorithmus, bei dem Einsen entsprechend vorgegebenen Zeilen- und Spaltensummen spaltenweise in eine Null-Matrix eingetragen werden. Dabei kann es vorkommen, dass ungültige Binärmatrizen produziert werden. Dies kann passieren, wenn es nach einigen generierten Spalten keine Möglichkeit mehr gibt, die nächste Spalte den vorgegebenen Zeilen- und Spaltensummen nach zu generieren. Angenommen man hat als Vorgabe  $r = (4, 4, 2, 1)$  als Zeilensummenvektor (ZSV) und  $c = (3, 3, 3, 1, 1)$  als Spaltensummenvektor (SSV) gegeben. Seien nun die ersten beiden Spalten folgendermaßen erzeugt worden:

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}.$$

In diesem Fall gibt es keine Möglichkeit die dritte Spalte zu generieren, da man in die dritte Spalte drei Einsen eintragen müsste, aber aufgrund der Zeilensummenvorgabe nur noch in die ersten beiden Zeilen Einsen eingetragen werden dürfen.

Angenommen man hat eine  $m \times n$  Binärmatrix zu generieren mit  $r \in \{0, \dots, n\}^m$  als ZSV und  $c \in \{0, \dots, m\}^n$  als SSV. Es seien die ersten  $l - 1$  Spalten erfolgreich generiert und die Zeilensummen entsprechend der bereits verteilten Einsen für den nächsten Schritt aktualisiert worden. Damit nun auch die  $l$ -te Spalte generiert werden kann, muss es eine  $m \times (n - (l - 1))$  Binärmatrix mit den aktualisierten Zeilen- und Spaltensummen  $(r_1^{(l)}, \dots, r_m^{(l)})$  und  $(c_1^{(l)}, \dots, c_{n-l+1}^{(l)})$  geben.

Um sicher zu stellen, dass eine solche Matrix existiert, kann man den Satz von Gale-Ryser benutzen. Dieser Satz wurde im selben Jahr von D.Gale und H.J. Ryser unabhängig voneinander aufgestellt, weswegen er nach beiden benannt wurde. Dieser Satz stellt die zweite angekündigte Eigenschaft von Binärmatrizen dar. Für ihn werden jedoch noch einige Definitionen und Folgerungen benötigt.

### 3.2.1 Definitionen und Folgerungen

**Definition 3.6 (Monotonie)** (nach [28] S. 61)

Sei  $a = (a_1, \dots, a_m) \in \mathbb{N}_0^m$ .  $a$  heißt genau dann monoton, wenn

$$a_1 \geq a_2 \geq \dots \geq a_m$$

gilt.

**Definition 3.7 (konjugiert)** (nach [7] S. 113)

Sei  $a = (a_1, \dots, a_m) \in \mathbb{N}_0^m$  und  $a_j^* := |\{a_i \mid a_i \geq j\}|$  für alle  $j \in \mathbb{N}$ , dann nennt man die Folge  $a_1^*, a_2^*, \dots$  konjugiert zu  $a$ . Eine konjugierte Folge ist insbesondere immer monoton.

**Beispiel 3.8**

Sei  $a = (5, 7, 2, 4, 0, 1, 1, 2, 1) \in \mathbb{N}_0^9$ , dann ist die Folge  $a^* = (a_i^*)_{i \in \mathbb{N}}$  mit

$$\begin{aligned} a_1^* &= |\{a_j \mid a_j \geq 1 \text{ für } j \in \{1, \dots, 9\}\}| = 8 \\ a_2^* &= |\{a_i \mid a_i \geq 2 \text{ für } j \in \{1, \dots, 9\}\}| = 5 \\ a_3^* &= |\{a_i \mid a_i \geq 3 \text{ für } j \in \{1, \dots, 9\}\}| = 3 \\ a_4^* &= |\{a_i \mid a_i \geq 4 \text{ für } j \in \{1, \dots, 9\}\}| = 3 \\ a_5^* &= |\{a_i \mid a_i \geq 5 \text{ für } j \in \{1, \dots, 9\}\}| = 2 \\ a_6^* &= |\{a_i \mid a_i \geq 6 \text{ für } j \in \{1, \dots, 9\}\}| = 1 \\ a_7^* &= |\{a_i \mid a_i \geq 7 \text{ für } j \in \{1, \dots, 9\}\}| = 1 \\ a_i^* &= 0 \text{ für } i \geq 8 \end{aligned}$$

konjugiert zu  $a$ . Offensichtlich ist  $a^*$  auch monoton. □

**Definition 3.9 (maximale Matrix)** (nach [28] S. 62)

Seien  $a_1, \dots, a_m \in \{0, 1\}^n$  Zeilenvektoren mit der Eigenschaft

$$\begin{aligned} (a_i)_j = 0 \Rightarrow (a_i)_k = 0 \quad &\text{für alle } i \in \{1, \dots, m\}, \\ &j, k \in \{1, \dots, n\} \text{ und } k \geq j. \end{aligned} \tag{3.4}$$

Dann hat  $a_i$  für  $i \in \{1, \dots, m\}$  die Form

$$a_i = (1, 1, \dots, 1, 0, 0, \dots, 0). \tag{3.5}$$

Eine Matrix  $A \in \{0, 1\}^{m \times n}$  der Form

$$A = \begin{pmatrix} a_1 \\ \vdots \\ a_m \end{pmatrix}$$

heißt maximal.

**Satz 3.10** (vergleiche [28] S. 62)

Sei  $p = (p_1, \dots, p_m) \in \mathbb{N}_0^m$  beliebig und  $\max_{i \in \{1, \dots, m\}}(p_i) = n$ , dann kann eine eindeutig bestimmte maximale  $m \times n$  Binärmatrix  $A^* = (a_{i,j}^*)_{i=1, \dots, m; j=1, \dots, n}$  konstruiert

werden, wobei  $p$  dem ZSV von  $A^*$  entspricht und zusätzlich die Eigenschaft (3.4) erfüllt ist. Eine solche Matrix nennt man dann zu  $p$  gehörende maximale Matrix.

In [28] wird diese Tatsache nicht bewiesen. Zum besseren Verständnis folgt nun trotzdem ein Beweis, der sich in zwei Teile gliedert. Zunächst wird die Existenz einer solchen Matrix  $A^*$  gezeigt. Anschließend wird die Eindeutigkeit dieser Matrix bewiesen.

BEWEIS

Es ist  $p = (p_1, \dots, p_m) \in \mathbb{N}_0^m$  und  $\max_{i \in \{1, \dots, m\}}(p_i) = n$ . Man setze nun

$$a_{i,j}^* = \begin{cases} 1 & , \text{ wenn } j \leq p_i \\ 0 & , \text{ wenn } j > p_i \end{cases}$$

für  $i \in \{1, \dots, m\}$  und  $j \in \{1, \dots, n\}$ . Es werden in  $A^*$  somit jeweils in die ersten  $p_i$  Spalten der  $i$ -ten Zeile Einsen eingetragen und alle anderen Einträge werden gleich Null gesetzt. Diese Matrix genügt der Anforderung (3.4) und  $p$  ist trivialerweise ZSV von  $A^*$ .

Angenommen  $\bar{A} = (\bar{a}_{i,j})_{i=1, \dots, m; j=1, \dots, n}$  ist eine weitere  $m \times n$  Binärmatrix mit ZSV  $p$ , für die ebenfalls die Eigenschaft (3.4) erfüllt ist und  $\bar{A} \neq A^*$  sei. Dann muss  $\bar{a}_{i,j_1} = 1$  in mindestens einer Zeile  $i \in \{1, \dots, m\}$  für ein  $j_1 \in \{1, \dots, n\}$  mit  $j_1 > p_i$  und gleichzeitig, damit die Zeilensumme nicht verändert wird,  $\bar{a}_{i,j_2} = 0$  für ein  $j_2 \in \{1, \dots, n\}$  mit  $j_2 \leq p_i$  gelten. Dann ist aber  $j_1 > j_2$  und (3.4) ist für  $\bar{A}$  nicht erfüllt. Damit ist  $A^*$  eindeutig bestimmt. ■

**Folgerung 3.11** (vergleiche [28] S. 62)

Sei  $p = (p_1, \dots, p_m) \in \mathbb{N}_0^m$ ,  $\max_{i \in \{1, \dots, m\}}(p_i) = n$  und  $A^*$  die zu  $p$  gehörende maximale Matrix. Dann ist der SSV  $p^* = (p_1^*, \dots, p_n^*)$  von  $A^*$  monoton und konjugiert zu  $p$ . Des Weiteren gilt

$$\sum_{i=1}^m p_i = \sum_{j=1}^n p_j^*$$

und  $A^*$  ist die einzige Matrix in  $\Sigma_{pp^*}$ . □

Auch diese Folgerung wird in [28] nicht bewiesen. Der Vollständigkeit halber folgt nun trotzdem ein kleiner Beweis.

BEWEIS

Da die Zeilen von  $A^*$  jeweils die Form (3.5) haben und somit  $p_i^*$  für  $i \in \{1, \dots, n\}$  gerade die Anzahl der Zeilen von  $A^*$  ist, in denen mindestens  $i$  Einsen vorkommen, bilden die Spaltensummen von  $A^*$  offensichtlich einen zu  $p$  konjugierten und somit

auch monotonen Vektor. Des Weiteren gilt

$$\sum_{i=1}^m p_i = \sum_{i=1}^m \sum_{j=1}^n a_{i,j}^* = \sum_{j=1}^n \sum_{i=1}^m a_{i,j}^* = \sum_{j=1}^n p_j^*.$$

Sei nun  $\bar{A}$  eine weitere Matrix in  $\Sigma_{pp^*}$ . Dann müsste  $A^*$  nach Satz 3.5 durch endlich viele  $2 \times 2$ -swaps in  $\bar{A}$  umwandelbar sein. Aufgrund der Bauweise von  $A^*$  existiert aber keine  $2 \times 2$  Untermatrix von  $A^*$  der Form  $A_1$  oder  $A_2$  aus Definition 3.4. Demnach kann es keine andere Matrix  $\bar{A}$  in  $\Sigma_{pp^*}$  geben. ■

**Beispiel 3.12**

Sei  $p = (3, 2, 4, 3, 2, 1)$ , dann gilt

$$A^* = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

und  $p^* = (6, 5, 3, 1)$  ist monoton und konjugiert zu  $p$ . Außerdem ist

$$\sum_{i=1}^6 p_i = \sum_{j=1}^4 p_j^* = 15. \quad \square$$

**Definition 3.13 (Majorisierung)** (nach [28] S. 61)

Seien  $p = (p_1, \dots, p_m) \in \mathbb{N}_0^m$  und  $q = (q_1, \dots, q_m) \in \mathbb{N}_0^m$  zwei  $m$ -Tupel. Es bezeichne

$$p_{\square} := (p_{[1]}, \dots, p_{[m]}) \quad \text{mit } p_{[1]} \geq p_{[2]} \geq \dots \geq p_{[m]} \quad (3.6)$$

die sortierten Komponenten von  $p$  und

$$q_{\square} := (q_{[1]}, \dots, q_{[m]}) \quad \text{mit } q_{[1]} \geq q_{[2]} \geq \dots \geq q_{[m]}$$

die sortierten Komponenten von  $q$ . Dann sei  $p \prec q$  ( $p$  wird von  $q$  majorisiert) definiert, wenn

$$\sum_{i=1}^k p_{[i]} \leq \sum_{i=1}^k q_{[i]} \quad \text{für } k \in \{1, \dots, m-1\} \quad (3.7)$$

und

$$\sum_{i=1}^m p_{[i]} = \sum_{i=1}^m q_{[i]} \quad (3.8)$$

gilt.

**Beispiel 3.14**

Sei  $p = (3, 2, 4, 3, 2, 1)$  und  $q = (1, 1, 5, 3, 4, 1)$ . Dann sind  $p_{\square} = (4, 3, 3, 2, 2, 1)$  und  $q_{\square} = (5, 4, 3, 1, 1, 1)$  die der Größe nach sortierten Einträge von  $p$  und  $q$ . Des Weiteren gilt

$$\begin{aligned} \sum_{i=1}^1 p_{[i]} &= 4 \leq 5 = \sum_{j=1}^1 q_{[j]} \\ \sum_{i=1}^2 p_{[i]} &= 7 \leq 9 = \sum_{j=1}^2 q_{[j]} \\ \sum_{i=1}^3 p_{[i]} &= 10 \leq 12 = \sum_{j=1}^3 q_{[j]} \\ \sum_{i=1}^4 p_{[i]} &= 12 \leq 13 = \sum_{j=1}^4 q_{[j]} \\ \sum_{i=1}^5 p_{[i]} &= 14 \leq 14 = \sum_{j=1}^5 q_{[j]} \\ \sum_{i=1}^6 p_{[i]} &= 15 = 15 = \sum_{j=1}^6 q_{[j]} \end{aligned}$$

und somit  $p \prec q$ . □

**Folgerung 3.15** (nach [7] S. 113)

Sei  $(j_1, \dots, j_m)$  eine Permutation von  $(1, \dots, m)$ . Dann folgt aus  $p \prec q$ , mit  $p$  und  $q$  wie in Definition 3.13,

$$\sum_{i=1}^k p_{j_i} \leq \sum_{i=1}^k q_{[i]} \quad \text{für } k \in \{1, \dots, m-1\} \quad (3.9)$$

und

$$\sum_{i=1}^m p_{j_i} = \sum_{i=1}^m q_{[i]}. \quad (3.10) \quad \square$$

**BEWEIS** (nach [7] S. 113)

Es gilt  $p_{[1]} \geq \dots \geq p_{[m]}$  für die sortierten Komponenten von  $p$ . Da für  $i \in \{1, \dots, k\}$  mit  $k \in \{1, \dots, m-1\}$  alle  $p_{[i]}$ , beginnend mit dem größten Wert, der Größe nach summiert werden und  $p_{j_i}$  für  $j \in \{1, \dots, k\}$  mit  $k \in \{1, \dots, m-1\}$  unsortiert summiert werden, gilt mit einer Permutation  $(j_1, \dots, j_m)$  von  $(1, \dots, m)$  stets

$$\sum_{i=1}^k p_{j_i} \leq \sum_{i=1}^k p_{[i]} \quad \text{für } k \in \{1, \dots, m-1\}.$$

Somit gilt weiter

$$\sum_{i=1}^k p_{j_i} \leq \sum_{i=1}^k p_{[i]} \stackrel{(3.7)}{\leq} \sum_{i=1}^k q_{[i]} \quad \text{für } k \in \{1, \dots, m-1\}$$

und

$$\sum_{i=1}^m p_{j_i} = \sum_{i=1}^m p_{[i]} \stackrel{(3.8)}{=} \sum_{i=1}^m q_{[i]}. \quad \blacksquare$$

Nach Folgerung 3.15 sind also (3.9) und (3.10) notwendige Bedingungen für  $p \prec q$ . Es folgt die schon angekündigte und für diese Arbeit zentrale Eigenschaft von Binärmatrizen.

### 3.2.2 Der Satz von Gale-Ryser

**Satz 3.16 (Gale-Ryser)** (nach [28] S. 63)

Seien  $p \in \mathbb{N}_0^m$  und  $q \in \mathbb{N}_0^n$  beliebig, wobei  $\max_{i \in \{1, \dots, m\}}(p_i) = n$  ist. Sei außerdem  $A^* \in \{0, 1\}^{m \times n}$  die zu  $p$  gehörende maximale  $m \times n$  Binärmatrix mit ZSV  $p$  und SSV  $p^*$ . Dann sind folgende Aussagen äquivalent:

$$\exists A \in \Sigma_{pq} \tag{3.11}$$

$$\Leftrightarrow q \prec p^*. \tag{3.12}$$

Der Beweis des Satzes erfolgt konstruktiv, so dass durch ihn auch gleich ein Schema zur Konstruktion einer validen Binärmatrix mit ZSV  $p$  und SSV  $q$  gegeben ist.

BEWEIS (nach [28] S. 63)

Zunächst wird (3.11) $\Rightarrow$ (3.12) bewiesen:

Es ist nach Voraussetzung  $A^* = (a_{i,j}^*)_{i=1, \dots, m; j=1, \dots, n} \in \Sigma_{pp^*}$ . Angenommen es gibt eine Matrix  $A = (a_{i,j})_{i=1, \dots, m; j=1, \dots, n} \in \Sigma_{pq}$ . Es gilt dann

$$\sum_{i=1}^m p_i = \sum_{i=1}^m \sum_{j=1}^n a_{i,j}^* = \sum_{j=1}^n \sum_{i=1}^m a_{i,j}^* = \sum_{j=1}^n p_j^*$$

und

$$\sum_{i=1}^m p_i = \sum_{i=1}^m \sum_{j=1}^n a_{i,j} = \sum_{j=1}^n \sum_{i=1}^m a_{i,j} = \sum_{j=1}^n q_j$$

und somit auch

$$\sum_{j=1}^n q_j = \sum_{j=1}^n p_j^*.$$

Da sich  $A$  von  $A^*$  also nur in den einzelnen Spaltensummen unterscheidet, kann nun  $A$  aus  $A^*$  konstruiert werden, indem Einsen in den Zeilen von  $A^*$  verschoben

werden. An den Zeilensummen ändert sich durch das Verschieben nichts. Da  $A^*$  nach Voraussetzung maximal ist, können in  $A^*$  Einsen nur von einem niedrigeren Spaltenindex zu einem höheren Spaltenindex verschoben werden, nämlich in eine Spalte, in der noch keine Eins steht. Daraus ergibt sich dann aber

$$\sum_{i=1}^k q_i \leq \sum_{i=1}^k p_i^* \quad \text{für } k \in \{1, \dots, n-1\}$$

und

$$\sum_{i=1}^n q_i = \sum_{i=1}^n p_i^*.$$

Dies ist nach Folgerung 3.15 eine notwendige Bedingung für  $q \prec p^*$ . Nach Folgerung 3.11 ist  $p^*$  monoton und somit gilt  $p_i^* = p_{[i]}^*$  für alle  $i \in \{1, \dots, n\}$ . Es bleibt also zu zeigen, dass auch für die sortierten Komponenten  $q_{[1]} \geq q_{[2]} \geq \dots \geq q_{[n]}$  von  $q$

$$\sum_{i=1}^k q_{[i]} \leq \sum_{i=1}^k p_i^* \quad \text{für } k \in \{1, \dots, n-1\} \quad (3.13)$$

und

$$\sum_{i=1}^n q_{[i]} = \sum_{i=1}^n p_i^* \quad (3.14)$$

gilt. Die Gleichungen (3.13) und (3.14) folgen direkt aus der Konstruktionsweise von  $A$ , denn es können nicht mehr Einsen in den Zeilen von  $A^*$  verschoben werden, als vorhanden sind. Somit ist stets  $q_{[i]} \leq p_i^*$  für  $i \in \{1, \dots, n\}$ .

Da (3.13) und (3.14) erfüllt sind, gilt  $q \prec p^*$ .

Es folgt der Beweis von (3.12) $\Rightarrow$ (3.11):

Angenommen es gilt  $q \prec p^*$ . Ist  $q_i = p_i^*$  für alle  $i \in \{1, \dots, n\}$ , so ist nichts zu tun, denn dann ist  $A^*$  die gesuchte Matrix. Sei also  $q \neq p^*$  und  $q_i \neq p_i^*$  für alle  $i \in \{1, \dots, n\}$ , dann kann eine Matrix  $A$  aus  $A^*$  durch Verschieben von Einsen in den Zeilen von  $A^*$  von links nach rechts konstruiert werden. Ohne Einschränkung der Allgemeinheit kann man voraussetzen, dass  $p$  und  $q$  monoton sind. Man könnte sonst  $A$  mit dem SSV  $q_{\square}$  konstruieren und danach die Spalten vertauschen, bis man  $q$  als SSV erhält. Dies ändert nichts an der Tatsache, dass eine solche Matrix existiert. Die Konstruktion von  $A$  aus  $A^*$  verläuft iterativ.

**Schritt (1)** Sei zunächst  $q_n$  die Spaltensumme der gewünschten letzten ( $n$ -ten) Spalte. Man verschiebe nun jeweils eine Eins aus den  $q_n$  Zeilen von  $A^*$  mit den größten Zeilensummen in die  $n$ -te Spalte. Befinden sich schon  $d_n$  Einsen in der  $n$ -ten Spalte von  $A^*$ , so verbleiben diese natürlich dort und es werden nur  $q_n - d_n$  Einsen verschoben. Wenn mehrere Zeilen von  $A^*$  die gleiche Zeilensumme haben, werden zuerst aus denjenigen Zeilen, die den größeren Index haben, Einsen verschoben,

damit die Monotonie von Zeilen- und Spaltensummen für die entstehende Matrix  $A_{n-1}^*$  erreicht wird. Es entsteht eine Matrix der Form

$$[A_{n-1}^*, A_1],$$

wobei  $A_{n-1}^*$  eine maximale  $m \times (n-1)$  Binärmatrix mit monotonen Zeilen- und Spaltensummen und  $A_1$  eine  $m \times 1$  Binärmatrix mit Spaltensumme  $q_n$  ist. Diese Matrix  $A_1$  bleibt nun unverändert bis zum Ende der Konstruktion. Es wird nur  $A_{n-1}^*$  weiter betrachtet.

**Schritt ( $k \Rightarrow k+1$ )** Angenommen es wurden nach dem Prinzip aus Schritt (1)  $k$  Spalten konstruiert, dann hat die entstandene Matrix die Form

$$[A_{n-k}^*, A_k],$$

wobei  $A_{n-k}^*$  eine maximale  $m \times (n-k)$  Binärmatrix mit monotonem ZSV und SSV und  $A_k$  eine  $m \times k$  Binärmatrix mit monotonem SSV  $(q_{n-k+1}, \dots, q_n)$  ist. Ist es nun möglich den  $(k+1)$ -ten Schritt zu konstruieren, so ist dies für alle  $k$  möglich und der Satz ist bewiesen.

Angenommen es ist nicht möglich aus  $A_{n-k}^*$  Einsen auf die beschriebene Weise so zu verschieben, dass für die  $(n-k)$ -te Spalte die Spaltensumme  $q_{n-k}$  erreicht wird. Mit dem monotonen SSV  $(e_1, \dots, e_{n-k})$  von  $A_{n-k}^*$  würde dies unter zwei Umständen zutreffen. Entweder es ist  $e_1 < q_{n-k}$ , dann wären nicht mehr genügend Einsen vorhanden, oder es ist  $e_{n-k} > q_{n-k}$ , dann wären schon zu viele Einsen in der  $(n-k)$ -ten Spalte vorhanden. Da  $q \prec p^*$  gilt, ist

$$\sum_{i=1}^n q_i = \sum_{i=1}^n p_i^*. \quad (3.15)$$

Weiter gilt durch die in Schritt (1) beschriebene Konstruktionsweise und (3.15)

$$\sum_{i=1}^{n-k} e_i = \sum_{j=1}^n p_j^* - \sum_{j=(n-k+1)}^n q_j \stackrel{(3.15)}{=} \sum_{i=1}^{n-k} q_i. \quad (3.16)$$

Angenommen es ist  $e_1 < q_{n-k}$ , dann folgt mit der Monotonie von  $(e_1, \dots, e_{n-k})$  und  $q$  sowie aus (3.16)

$$\sum_{i=1}^{n-k} q_i \stackrel{(3.16)}{=} \sum_{i=1}^{n-k} e_i \leq (n-k)e_1 < (n-k)q_{n-k} \leq \sum_{i=1}^{n-k} q_i,$$

was zu einem Widerspruch führt.

Sei andererseits  $e_{n-k} > q_{n-k}$ , dann gilt aufgrund der Monotonie von  $q$  auch  $e_{n-k} > q_i$

für alle  $i \in \{(n - k + 1), \dots, n\}$ . Das heißt, dass in der  $(n - k)$ -ten Spalte von  $A_{n-k}^*$  mehr Einsen sind als in jeder Spalte von  $A_k$ . Da aufgrund des monotonen Konstruktionsprinzips zuerst Einsen aus Zeilen mit einem niedrigen Index verschoben werden und da  $A_{n-k}^*$  maximal ist, bedeutet dies, dass in jeder Spalte der ersten  $e_{n-k}$  Zeilen von  $A_k$  mindestens eine Null steht. Daraus folgt dann wiederum, dass die letzten  $m - e_{n-k-1}$  Zeilen von  $A_k$  nur aus Nullen bestehen, denn es werden ja zuerst Einsen in den niedriger indizierten Zeilen verschoben. Das heißt aber auch, dass aus den Spalten  $1, \dots, n - k - 1$  nach dem Konstruktionsprinzip noch keine Einsen verschoben worden sind. Es ergibt sich also

$$\sum_{i=1}^{n-k-1} e_i = \sum_{i=1}^{n-k-1} p_i^*. \quad (3.17)$$

Da  $q$  von  $p^*$  majorisiert wird, folgt

$$\begin{aligned} \left( \sum_{i=1}^{n-k-1} e_i \right) + e_{n-k} &\stackrel{(3.16)}{=} \left( \sum_{i=1}^{n-k-1} q_i \right) + q_{n-k} \\ &\stackrel{q \prec p^*}{\leq} \left( \sum_{i=1}^{n-k-1} p_i^* \right) + q_{n-k} \\ &\stackrel{(3.17)}{=} \left( \sum_{i=1}^{n-k-1} e_i \right) + q_{n-k}. \end{aligned}$$

Dies würde bedeuten, dass  $e_{n-k} \leq q_{n-k}$  ist, was einen Widerspruch zu  $e_{n-k} > q_{n-k}$  darstellt.

Daraus folgt, dass man mit Hilfe dieser Konstruktionsvorgaben eine valide Matrix  $A \in \Sigma_{pq}$  aus  $A^*$  konstruieren kann, womit der Satz bewiesen ist. ■

### Beispiel 3.17

Sei  $p = (4, 3, 3, 2, 2, 1)$  und  $q = (5, 4, 4, 2)$ , dann ist

$$A^* = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

die zu  $p$  gehörende maximale Matrix mit SSV  $p^* = (6, 5, 3, 1)$ . Es ist offensichtlich  $q \prec p^*$ , und somit kann man  $A^*$  nach Satz 3.16 in eine Matrix  $A \in \Sigma_{pq}$  umwandeln.

Es gilt

$$A^* = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} = A. \quad \square$$

### 3.2.3 Anwendung des Satzes von Gale-Ryser auf die Generierung von Binärmatrizen mit festen Zeilen- und Spaltensummen

Nach Satz 3.16 ist  $q \prec p^*$  eine hinreichende Bedingung für die Existenz einer Matrix  $A \in \Sigma_{pq}$ .

In dem in Kapitel 6 folgenden Verfahren werden Binärmatrizen spaltenweise generiert. Dabei handelt es sich um ein sequentielles Importance Sampling Verfahren. Was das bedeutet wird in den späteren Kapiteln erläutert. Das Verfahren wird im Folgenden kurz mit SIS-Verfahren bezeichnet.

Bei der Generierung der  $l$ -ten Spalte einer Matrix  $A = (a_{i,j})_{i=1,\dots,m;j=1,\dots,n} \in \Sigma_{rc}$  mittels des SIS-Verfahrens muss sichergestellt werden, dass nach dieser Generierung eine  $m \times (n-l)$  Binärmatrizen mit den, resultierend aus den in den ersten  $l$  generierten Spalten verteilten Einsen, aktuellen Zeilensummen  $r_1^{(l+1)}, \dots, r_m^{(l+1)}$  und Spaltensummen  $c_1^{(l+1)}, \dots, c_{n-l}^{(l+1)}$  existiert. Sei  $A$  zu Beginn der Generierung eine  $m \times n$  Nullmatrix, die im Laufe des Verfahrens spaltenweise mit einigen Einsen gefüllt wird. Ist eine Spalte generiert, wird im weiteren Verlauf nur noch die Matrix ohne diese Spalte betrachtet. Es sei für die jeweiligen Zeilen- und Spaltensummen

$$r^{(1)} = (r_1^{(1)}, \dots, r_m^{(1)}) := (r_1, \dots, r_m) = r, \quad (3.18)$$

$$c^{(1)} = (c_1^{(1)}, \dots, c_n^{(1)}) := (c_1, \dots, c_n) = c, \quad (3.19)$$

und

$$r_i^{(l+1)} := r_i - \sum_{j=1}^l a_{i,j}, \quad (3.20)$$

$$c_k^{(l+1)} := c_{k+l} \quad (3.21)$$

für alle  $l \in \{1, \dots, n-1\}$ ,  $i \in \{1, \dots, m\}$  und  $k \in \{1, \dots, n-l\}$  definiert. Dabei bezeichne  $a_{i,j}$  die jeweils aktuellen Einträge von  $A$ .

Damit eine  $m \times (n - l)$  Binärmatrix mit ZSV  $r^{(l+1)} = (r_1^{(l+1)}, \dots, r_m^{(l+1)})$  und SSV  $c^{(l+1)} = (c_1^{(l+1)}, \dots, c_{n-l}^{(l+1)})$  existiert, muss nach Satz 3.16

$$r^{(l+1)} \prec c^{(l+1)*} \quad (3.22)$$

gewährleistet sein. Da die Anzahl der Elemente von  $\Sigma_{rc}$  nicht von der Reihenfolge der Zeilensummen abhängt, kann man ohne Einschränkung der Allgemeinheit den ZSV  $r$  als monoton annehmen. Um für jeden neuen Schritt des SIS-Verfahrens sicherzustellen, dass  $r_1^{(l+1)} \geq \dots \geq r_m^{(l+1)}$  gilt, werden die Zeilen nach jeder fertig generierten Spalte der Größe nach sortiert. Einfaches Umsortieren der Zeilen am Ende des SIS-Verfahrens erzeugt dann eine Matrix mit den anfangs geforderten Zeilensummen.

Seien nun  $(l - 1)$  Spalten erfolgreich generiert worden, dann ist nach Definition  $(c_1^{(l)}, \dots, c_{n-l+1}^{(l)}) = (c_l, \dots, c_n)$  und  $(c_1^{(l)*}, \dots, c_m^{(l)*})$  sei dazu konjugiert. Weiter ist  $(c_1^{(l+1)*}, \dots, c_m^{(l+1)*})$  mit

$$c_j^{(l+1)*} = \begin{cases} c_j^{(l)*} - 1 & \text{für } 1 \leq j \leq c_l \\ c_j^{(l)*} & \text{für } j > c_l \end{cases} \quad (3.23)$$

konjugiert zu  $c^{(l+1)} \stackrel{(3.21)}{=} (c_{l+1}, \dots, c_n)$ . Gilt nun  $r^{(l+1)} \prec c^{(l+1)*}$ , so existiert nach Satz 3.16 eine  $m \times (n - l)$  Binärmatrix mit Zeilensummen  $r_1^{(l+1)}, \dots, r_m^{(l+1)}$  und Spaltensummen  $c_1^{(l+1)}, \dots, c_{n-l}^{(l+1)}$ . Es muss also mit (3.7) und (3.8)

$$\sum_{i=1}^k r_{[i]}^{(l+1)} \leq \sum_{i=1}^k c_i^{(l+1)*} \quad \text{für } k = 1, \dots, m - 1$$

und

$$\sum_{i=1}^m r_{[i]}^{(l+1)} = \sum_{i=1}^m c_i^{(l+1)*}$$

gelten, wobei  $r_{[i]}^{(l+1)}$  für  $i \in \{1, \dots, m\}$  analog zu (3.6) die der Größe nach sortierten Komponenten von  $r^{(l+1)}$  bezeichnen. Nach Folgerung 3.15 ist

$$\sum_{i=1}^k r_i^{(l+1)} \leq \sum_{i=1}^k c_i^{(l+1)*} \quad \text{für } k = 1, \dots, m - 1 \quad (3.24)$$

und

$$\sum_{i=1}^m r_i^{(l+1)} = \sum_{i=1}^m c_i^{(l+1)*}, \quad (3.25)$$

eine notwendige Bedingung dafür, dass  $r^{(l+1)} \prec c^{(l+1)}$  gilt und somit eine  $m \times (n - l)$  Binärmatrix mit Zeilensummen  $r_1^{(l+1)}, \dots, r_m^{(l+1)}$  und Spaltensummen  $c_1^{(l+1)}, \dots, c_{n-l}^{(l+1)}$

existiert. Der folgende Satz 3.18 zeigt, dass (3.24) und (3.25) im Falle der Binärmatrixgenerierung auch schon eine hinreichende Bedingung für  $r^{(l+1)} \prec c^{(l+1)}$  ist.

**Satz 3.18** (nach [7] S. 113)

Seien  $p = (p_1, \dots, p_m) \in \mathbb{N}_0^m$  und  $q = (q_1, \dots, q_m) \in \mathbb{N}_0^m$  monotone Vektoren und seien  $d \geq 1$  Einträge in  $p$  ungleich Null. Man wähle nun  $d'$  dieser Einträge aus  $p$ , wobei  $d' \in \{1, \dots, d\}$  ist. Seien  $p_{k_1} \neq 0, \dots, p_{k_{d'}} \neq 0$  diese Einträge, dann definiere man  $p' := (p'_1, \dots, p'_m)$  mit

$$p'_j = \begin{cases} p_j - 1 & \text{für } j \in \{k_i \mid 1 \leq i \leq d'\} \\ p_j & \text{andernfalls.} \end{cases} \quad (3.26)$$

Gilt

$$\sum_{i=1}^k p'_i \leq \sum_{i=1}^k q_i \quad \text{für } k = 1, \dots, m-1 \quad (3.27)$$

und

$$\sum_{i=1}^m p'_i = \sum_{i=1}^m q_i, \quad (3.28)$$

so ist  $p' \prec q$ .

BEWEIS (nach [7] S. 119)

Angenommen  $p_1, \dots, p_m$  nehmen  $l$  unterschiedliche Werte aus  $\mathbb{N}_0$  an, wobei  $l \leq m$  gilt. Es gibt dann aufgrund der Monotonie von  $p$  Übergangsstellen  $i_1 < \dots < i_l$  von verschiedenen Werten in  $p$ , für die

$$p_{i_{k-1}+1} = p_{i_{k-1}+2} = \dots = p_{i_k} > p_{i_k+1} \quad \text{für } k = 1, \dots, l-1, \quad (3.29)$$

$$p_{i_{l-1}+1} = p_{i_{l-1}+2} = \dots = p_{i_l} = p_m \quad (3.30)$$

gilt, wobei  $i_0 = 0$  sei. Es bezeichne  $b_k = \{p_j \mid i_{k-1} < j \leq i_k\}$  den  $k$ -ten Block für  $k \in \{1, \dots, l\}$ . Da nun nach (3.26)  $p'_i = p_i$  oder  $p'_i = p_i - 1$  für alle  $i \in \{1, \dots, m\}$  gilt und  $p$  monoton ist, ist offensichtlich für  $p'_i \in b_k$  und  $p'_j \in b_{k+1}$  mit  $i, j \in \{1, \dots, m\}$  und  $k \in \{1, \dots, l-1\}$  immer  $p'_i \geq p'_j$ . Es sind also die Werte in jedem Block von  $p'$  größer oder gleich den Werten in den höher indizierten Blöcken von  $p'$ . Jedoch kann es sein, dass die Werte in einem Block von  $p'$  nicht der Größe nach sortiert sind. Dies ist beispielsweise der Fall, wenn  $p'_j = p_j - 1 = p_{i_k} - 1$  und  $p'_{j+1} = p_j = p_{i_k}$  beziehungsweise  $p'_j = p'_{j+1} - 1$  für  $p'_j \in b_k$  und  $p'_{j+1} \in b_k$  mit  $k \in \{1, \dots, l-1\}$  gilt. Könnte man stets in einem Block  $p'_j$  mit  $p'_{j+1}$  vertauschen, ohne dabei die Gültigkeit von (3.27) und (3.28) zu beeinträchtigen, so erreichte man, dass  $p'$  monoton ist und gleichzeitig (3.27) und (3.28) gilt. Dann würde sofort  $p' \prec q$  folgen.

Im Folgenden wird gezeigt, dass solche Vertauschungen (3.27) und (3.28) nicht beeinträchtigen. Ist  $p'_j = p'_{j+1}$ , so bewirkt eine Vertauschung von  $p'_j$  und  $p'_{j+1}$  natürlich keine Veränderung. Es wird nun für ein  $k \in \{1, \dots, l-1\}$  das Paar  $p'_j \in b_k$  und  $p'_{j+1} \in b_k$  mit  $p'_{j+1} = p'_j - 1$  betrachtet. Dabei muss man zwei Fälle unterscheiden.

Fall 1: Angenommen es sei

$$\sum_{i=1}^j p'_i < \sum_{i=1}^j q_i, \quad (3.31)$$

dann gilt (3.27) und (3.28) auch noch, wenn  $p'_j$  mit  $p'_{j+1}$  getauscht wird, denn es gilt

$$\begin{aligned} \sum_{i=1}^j p'_i &\stackrel{(3.31)}{<} \sum_{i=1}^j q_i \\ \Rightarrow \sum_{i=1}^j p'_i &\leq \left( \sum_{i=1}^j q_i \right) - 1 \\ \Rightarrow \left( \sum_{i=1}^{j-1} p'_i \right) + p'_j &\leq \left( \sum_{i=1}^j q_i \right) - 1 \\ \Rightarrow \left( \sum_{i=1}^{j-1} p'_i \right) + p'_j - 1 &\leq \sum_{i=1}^j q_i \\ \Rightarrow \left( \sum_{i=1}^{j-1} p'_i \right) + p'_{j+1} &\leq \sum_{i=1}^j q_i \end{aligned}$$

und  $\sum_{i=1}^{j+1} p'_i$  wird durch das Vertauschen nicht verändert.

Fall 2: Angenommen es sei

$$\sum_{i=1}^j p'_i = \sum_{i=1}^j q_i. \quad (3.32)$$

Es wird im Folgenden gezeigt, dass dieser Fall nie eintritt.

Es muss nach (3.27) und (3.28)

$$\sum_{i=1}^{j+1} p'_i \leq \sum_{i=1}^{j+1} q_i$$

gelten. Mit (3.32) muss dann  $q_{j+1} \geq p'_{j+1} = p_{i_k}$  gelten. Daraus und aus der Monotonie von  $q$  folgt dann

$$q_{i_{k-1}+1} \geq \dots \geq q_j \geq q_{j+1} \geq p'_{j+1}. \quad (3.33)$$

Außerdem gilt mit (3.26) und (3.29)  $p'_i \leq p_{i_k}$  für  $i_{k-1} + 1 \leq i < j$  und  $p'_j = p_{i_k} - 1$ .

Damit und mit (3.33) folgt

$$\sum_{i=i_{k-1}+1}^j p'_i < \sum_{i=i_{k-1}+1}^j q_i. \quad (3.34)$$

Des Weiteren gilt nach (3.27) und (3.28)

$$\sum_{i=1}^{i_{k-1}} p'_i \leq \sum_{i=1}^{i_{k-1}} q_i. \quad (3.35)$$

Aus (3.34) und (3.35) folgt nun

$$\sum_{i=1}^j p'_i < \sum_{i=1}^j q_i,$$

was der Annahme (3.32) widerspricht.

Wie gezeigt wurde, ist es also möglich  $p'_j$  mit  $p'_{j+1}$  in einem Block zu vertauschen, wobei  $p'_j = p'_{j+1} - 1$  ist. Die Gültigkeit von (3.27) und (3.28) wird dabei nicht beeinträchtigt. Da es aber nur endlich viele solcher Paare in  $p'$  geben kann, ist nach endlich vielen Vertauschungen  $p'$  monoton und (3.27) und (3.28) sind aufgrund der Monotonie von  $q$  äquivalent zu (3.7) und (3.8) und es ist  $p' \prec q$ . ■

Satz 3.18 kann man nun auf das Problem der Matrix-Generierung anwenden. Während des SIS-Verfahrens ist es möglich, nach jeder generierten Spalte die Zeilen den Zeilensummen nach neu zu ordnen, sodass die Zeilensummen nach jedem Schritt des SIS-Verfahrens monoton sind.

Setzt man in der Notation von Satz 3.18  $q := c^{(l+1)*}$ ,  $p := r^{(l)}$  und  $p' := r^{(l+1)}$ , dann ist sofort ersichtlich, dass die Anforderungen für Satz 3.18 erfüllt sind. Es müssen also die Gleichungen (3.24) und (3.25) erfüllt sein, damit  $r^{(l+1)} \prec c^{(l+1)*}$  gilt, und man nach Satz 3.16 weiß, dass eine  $m \times (n - l)$  Binärmatrix mit Zeilensummen  $r_1^{(l+1)}, \dots, r_m^{(l+1)}$  und Spaltensummen  $c_1^{(l+1)}, \dots, c_{n-l}^{(l+1)}$  existiert. Zwar ist  $r^{(l+1)}$  unbekannt, solange die  $l$ -te Spalte nicht komplett generiert ist, jedoch kann auch aus  $r^{(l)}$  und  $c^{(l+1)*}$  eine leicht zu überprüfende Alternative für (3.24) und (3.25) formuliert werden. Dazu vergleicht man  $\sum_{i=1}^k r_i^{(l)}$  mit  $\sum_{i=1}^k c_i^{(l+1)*}$  für  $1 \leq k \leq n$  und erhält folgendes Schema.

Es ist

$$r_i^{(l+1)} = r_i^{(l)} - a_{i,(l+1)} \quad \text{für } i \in \{1, \dots, m\}, \quad (3.36)$$

wobei  $a_{i,(l+1)} \in \{0, 1\}$  ist, je nachdem, ob in die  $(l + 1)$ -te Spalte und die  $i$ -te Zeile von  $A$  eine Eins eingetragen wird oder nicht. Ist  $\sum_{i=1}^k r_i^{(l)} > \sum_{i=1}^k c_i^{(l+1)*}$  für ein

$k \in \{1, \dots, m\}$ , so muss für dieses  $k$

$$\sum_{i=1}^k a_{i,(l+1)} \geq \sum_{i=1}^k r_i^{(l)} - \sum_{i=1}^k c_i^{(l+1)*} \quad (3.37)$$

im Laufe der Spaltengenerierung erreicht werden, damit

$$\sum_{i=1}^k r_i^{(l+1)} \leq \sum_{i=1}^k c_i^{(l+1)*} \quad (3.38)$$

erfüllt ist, denn es ist mit (3.37)

$$\sum_{i=1}^k r_i^{(l+1)} \stackrel{(3.36)}{=} \sum_{i=1}^k r_i^{(l)} - \sum_{i=1}^k a_{i,(l+1)} \quad (3.39)$$

$$\stackrel{(3.37)}{\leq} \sum_{i=1}^k r_i^{(l)} - \left( \sum_{i=1}^k r_i^{(l)} - \sum_{i=1}^k c_i^{(l+1)*} \right) \quad (3.40)$$

$$= \sum_{i=1}^k c_i^{(l+1)*}. \quad (3.41)$$

Die Ungleichung in (3.37) bedeutet, dass in der  $(l+1)$ -ten Spalte von  $A$  bis zur  $k$ -ten Zeile mindestens  $\sum_{i=1}^k r_i^{(l)} - \sum_{i=1}^k c_i^{(l+1)*}$  Einsen eingetragen werden müssen. Ansonsten wäre (3.38) nicht erfüllt und somit wäre  $r^{(l+1)} \prec c^{(l+1)*}$  nicht gegeben. Ist andererseits  $\sum_{i=1}^k r_i^{(l)} \leq \sum_{i=1}^k c_i^{(l+1)*}$  für ein  $k \in \{1, \dots, m\}$ , so ist die Anforderung (3.7) für dieses  $k$  erfüllt und es müssen für die  $k$ -te Zeile keine Einschränkungen eingeführt werden.

Dass  $r^{(l)} \prec c^{(l)*}$  gilt, wurde bei der Generierung der  $l$ -ten Spalte von  $A$  während des SIS-Verfahrens sichergestellt. Deshalb gilt

$$\sum_{i=1}^m c_i^{(l)*} \stackrel{(3.8)}{=} \sum_{i=1}^m r_i^{(l)}. \quad (3.42)$$

Damit folgt weiter

$$\begin{aligned} \sum_{i=1}^m c_i^{(l+1)*} &\stackrel{(3.23)}{=} \left( \sum_{i=1}^m c_i^{(l)*} \right) - c_l \\ &\stackrel{(3.42)}{=} \left( \sum_{i=1}^m r_i^{(l)} \right) - c_l \\ &\stackrel{(3.20)}{=} \sum_{i=1}^m r_i^{(l+1)}. \end{aligned} \quad (3.43)$$

Gilt (3.38) für alle  $k \in \{1, \dots, m\}$ , so folgt aufgrund der sichergestellten Monotonie von  $r^{(l+1)}$  und  $c^{(l+1)*}$  und wegen (3.43)  $r^{(l+1)} \prec c^{(l+1)*}$ . Somit existiert eine  $m \times (n-l)$

Binärmatrix mit ZSV  $r^{(l+1)}$  und SSV  $c^{(l+1)}$  und das SIS-Verfahren kann fortgeführt werden.

Im Folgenden wird ausgeführt, wie sich die oben genannten Zusammenhänge für die Praxis nutzen lassen.

Ein  $k$ , für das  $\sum_{i=1}^k r_i^{(l)} > \sum_{i=1}^k c_i^{(l+1)*}$  ist, nennt man einen Knoten. Sammelt man diese Knoten in einem Tupel  $K = (k_1, \dots, k_t)$  und die Anzahl an Einsen, die bis einschließlich zu den jeweiligen Knoten eingetragen werden müssen, in einem zweiten Tupel  $V = (v_1, \dots, v_t)$ , kann man mit Hilfe dieser Tupel während des später folgenden SIS-Verfahrens einfach gewährleisten, dass eine valide Binärmatrix erzeugt wird. Zur weiteren Vereinfachung kann man einige redundante Knoten aus der Betrachtung entfernen. So zum Beispiel diejenigen Knoten  $k_j$  ( $j \in \{1, \dots, t\}$ ), für die  $v_j \leq v_i$  ( $i \in \{1, \dots, t\}$ ) für ein  $i < j$  ist, denn dann wären vor Knoten  $k_j$  ohnehin schon mindestens so viele Einsen eingetragen worden, wie gefordert sind. Außerdem kann man Knoten  $k_j$  entfernen, wenn  $v_i - v_j \geq k_i - k_j$  für ein  $i > j$  ist, denn hier werden durch  $k_i$  ohnehin vor  $k_j$  mehr Einsen eingetragen, als  $k_j$  erfordert.

**Beispiel 3.19**

Sei  $r = (4, 4, 2, 1) \in \mathbb{N}^4$ ,  $c = (3, 3, 3, 1, 1) \in \mathbb{N}^5$  und

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} \in \Sigma_{rc}.$$

Es ist dann  $r^{(1)} \stackrel{(3.18)}{=} r$ ,  $c^{(2)} \stackrel{(3.21)}{=} (3, 3, 1, 1)$  und  $c^{(2)*} = (4, 2, 2, 0)$ .

Angenommen man möchte nun die erste Spalte einer weiteren Matrix  $A' \in \Sigma_{rc}$  generieren. Damit es nach der Generierung der ersten Spalte von  $A'$  auch noch möglich ist, die restlichen Spalten zu generieren, muss nach Satz 3.16  $r^{(2)} \prec c^{(2)*}$  gelten, wobei  $r^{(2)}$  sich aus (3.20) ergibt. Nach Satz 3.18 genügt es nun zu gewährleisten, dass (3.24) und (3.25) erfüllt sind. Dazu müssen nun sämtliche Knoten für die erste Spalte bestimmt werden. Es gilt

$$\begin{aligned} \sum_{i=1}^1 r_i &= 4 = 4 = \sum_{i=1}^1 c_i^{(2)*} \\ \sum_{i=1}^2 r_i &= 8 > 6 = \sum_{i=1}^2 c_i^{(2)*} \\ \sum_{i=1}^3 r_i &= 10 > 8 = \sum_{i=1}^3 c_i^{(2)*} \\ \sum_{i=1}^4 r_i &= 11 > 8 = \sum_{i=1}^4 c_i^{(2)*}. \end{aligned}$$

Somit ist  $K = (2, 3, 4)$  der Vektor aller Knoten und die Anzahl der Einsen, die bis zu den jeweiligen Knoten mindestens eingesetzt werden müssen, wird in dem Vektor  $V = (v_1, v_2, v_3, v_4)$  angegeben, wobei

$$v_1 = \sum_{i=1}^2 r_i - \sum_{i=1}^2 c_i^{(2)*} = 8 - 6 = 2,$$

$$v_2 = \sum_{i=1}^3 r_i - \sum_{i=1}^3 c_i^{(2)*} = 10 - 8 = 2,$$

$$v_3 = \sum_{i=1}^4 r_i - \sum_{i=1}^4 c_i^{(2)*} = 11 - 8 = 3$$

ist. Da  $v_2 = 2 = v_1$  ist, kann der Knoten  $k_2$  aus der weiteren Betrachtung ausgeschlossen werden.

Es ist jetzt also nötig, 2 Einsen in die ersten 2 Zeilen der ersten Spalte von  $A'$  und eine weitere Eins entweder in die dritte oder die vierte Zeile der ersten Spalte von  $A'$  einzutragen, um zu gewährleisten, dass die anderen Spalten von  $A'$  auch noch generierbar sind. Auf analoge Weise können alle weiteren Spalten der Matrix  $A' \in \Sigma_{rc}$  generiert werden. □

## 4 Importance Sampling

In vielen Fällen ist man daran interessiert, *zufällige* Stichproben einer gegebenen Verteilung zu simulieren. Bei der praktischen Anwendung des in Kapitel 6 folgenden Verfahrens wird es darum gehen, auf ökologischen Daten basierende Binärmatrizen auf einen möglichen Konkurrenzzusammenhang bei der Verbreitung der durch die Binärmatrix widergegebenen Arten aufzudecken. Dies wird durch das Monte-Carlo Prinzip erreicht. Dabei wird zu konkreten erhobenen Daten mittels geeigneter Verfahren mit einem Computer eine unabhängige identisch verteilte Stichprobe der entsprechenden Grundgesamtheit generiert. Diese Stichprobe wird dann mit den original Daten verglichen.

Wurde eine Presence-Absence Matrix erhoben, ist man nun daran interessiert, eine auf der Menge aller möglichen Binärmatrizen mit gegebenen Zeilen- und Spaltensummen gleichverteilte Stichprobe von Binärmatrizen zu generieren. Diese sollen dann mit Hilfe geeigneter Statistiken mit der durch reale Beobachtungen gewonnenen Binärmatrix verglichen werden.

Im Allgemeinen ist es schwierig, die Gleichverteilung über der Menge  $\Sigma_{rc}$  aller  $m \times n$  Binärmatrizen mit Zeilensummenvektor  $r = (r_1, \dots, r_m)$  und Spaltensummenvektor  $c = (c_1, \dots, c_n)$  explizit zu bestimmen. Hierzu müsste man die genaue Anzahl dieser Matrizen kennen. Ein Ansatz zur exakten Berechnung der Anzahl solcher Matrizen wird in [24] geliefert. Es würde allerdings den Rahmen dieser Arbeit sprengen, diesen einzubringen. Der interessierte Leser sei an dieser Stelle auf [24] verwiesen.

Ist  $|\Sigma_{rc}|$  nicht bekannt, so kann man Matrizen von einer anderen Verteilung generieren, die der Gleichverteilung hinreichend nahe ist. Diese Methode wird durch das Importance Sampling beschrieben.

### 4.1 Funktionsweise des Importance Sampling

Angenommen man möchte eine Stichprobe von einer diskreten Zufallsvariablen  $Y$  mit der Verteilung  $\pi$  generieren. Dazu ist es notwendig die Verteilung  $\pi$  explizit zu kennen. Ist es aber entweder nicht möglich oder viel zu aufwändig, diese Verteilung zu bestimmen, kann man eventuell eine andere Zufallsvariable  $X$  beziehungsweise

deren Verteilung, die einfacher zu bestimmen und gleichzeitig  $\pi$  möglichst ähnlich ist, für die Generierung der Stichprobe nutzen.

Dieses Verfahren wird mit Importance Sampling bezeichnet. Der Name entspringt einem weiteren Anwendungsbereich des Importance Sampling. Dabei wird eine Stichprobe von einer Verteilung generiert, die  $\pi$  für Ereignisse mit hoher Eintrittswahrscheinlichkeit gleicht, jedoch Ereignisse mit geringer Eintrittswahrscheinlichkeit ignoriert. Man konzentriert sich also auf die „wichtigen“ Ereignisse und kann somit die Varianz reduzieren.

Im Folgenden bezeichne  $g$  die Verteilung einer diskreten Zufallsvariablen  $X$ , die zur Generierung einer Stichprobe anstatt von  $\pi$  genutzt werden soll, und  $g_X$  die entsprechende diskrete Dichte. Für  $g$  sollten dabei natürlich geeignete Anforderungen erfüllt sein. Man nennt  $g$  auch *Probeverteilung* (engl.: trial distribution). Eine Anforderung an die Probeverteilung ist, dass der so genannte *Träger* der Probeverteilung den Träger der *Zielverteilung* (engl.: target distribution)  $\pi$  enthält. Ein Träger ist wie folgt definiert.

**Definition 4.1 (Träger)** (nach [8] S. 157)

Sei  $(\Omega, P)$  ein diskreter Wahrscheinlichkeitsraum. Dann bezeichnet man die Menge  $T_P = \{\omega \in \Omega \mid P(\omega) > 0\}$  aller Nichtnullstellen von  $P$  als Träger von  $P$ .

Es muss also für die Zielverteilung  $\pi$  und die Probeverteilung  $g$

$$T_\pi \subseteq T_g \tag{4.1}$$

gelten, damit sämtliche Ereignisse, die unter  $\pi$  eintreten können, auch unter  $g$  eintreten können.

**Beispiel 4.2**

Sei  $\Sigma_{rc}$  die Menge der Binärmatrizen mit ZSV  $r$  und SSV  $c$  und  $Y$  eine diskrete Zufallsvariable mit Werten in  $\Sigma_{rc}$ . Sei weiter  $\pi_Y(x) = \frac{1}{|\Sigma_{rc}|}$  die diskrete Dichte der Gleichverteilung  $\pi$  von  $Y$  auf  $\Sigma_{rc}$  und somit insbesondere  $\pi_Y(x) > 0$  für alle  $x \in \Sigma_{rc}$ . Es ist folglich  $T_\pi = \Sigma_{rc}$ . Die Menge  $\Sigma_{rc}$  ist in den meisten Fällen nur sehr schwer zu bestimmen. Sei  $g$  eine (diskrete) Probeverteilung für ein Importance Sampling Verfahren. Es muss also (4.1) gelten, woraus folgt, dass  $g_X(x) \neq 0$  für alle  $x \in \Sigma_{rc}$  und die diskrete Dichte  $g_X$  zu der Zufallsvariablen  $X$  gelten muss. □

Hat man eine Probeverteilung  $g$  gefunden, kann man eine Stichprobe  $x_1, \dots, x_n$  als unabhängige Realisierungen der diskreten,  $g$ -verteilten Zufallsvariablen  $X$  mit Werten in  $\mathcal{X}$  generieren. In der Regel möchte man mit Hilfe dieser Stichprobe  $\mu_\pi := E_\pi(h) = \sum_{x \in \mathcal{X}} \pi_Y(x)h(x)$  zu einer Funktion (z.B. Test)  $h : \mathcal{X} \rightarrow \mathbb{R}$  berechnen. Mit  $x_1, \dots, x_n$  ist es aber lediglich möglich  $\mu_g := E_g(h) = \sum_{x \in \mathcal{X}} g_X(x)h(x)$

zu schätzen.

Eine erwartungstreue Schätzung wäre zum Beispiel  $\widehat{\mu}_g := \frac{1}{n} \sum_{i=1}^n h(x_i)$ . Durch Anpassung von  $h$  ist es möglich auch  $\mu_\pi$  zu bestimmen. Man definiere  $\tilde{h}(x) := \frac{\pi_Y(x)}{g_X(x)} h(x)$  für alle  $x \in \mathcal{X}$  mit  $g_X(x) \neq 0$  und  $\tilde{h}(x) = \pi_Y(x)$  für  $g_X(x) = 0$ . Wenn  $g_X(x) = 0$  ist, folgt mit (4.1) aber  $\pi_Y(x) = 0$ , und es gilt somit

$$\begin{aligned} E_g(\tilde{h}) &= \sum_{x \in \mathcal{X}} g_X(x) \tilde{h}(x) \\ &= \sum_{\substack{x \in \mathcal{X} \\ g_X(x) \neq 0}} g_X(x) \frac{\pi_Y(x)}{g_X(x)} h(x) \\ &= \sum_{x \in \mathcal{X}} \pi_Y(x) h(x) = E_\pi(h). \end{aligned}$$

Definiert man eine Zufallsvariable  $W : \mathcal{X} \rightarrow \mathbb{R}$  auf  $\mathcal{X}$  mit  $W(x) := \frac{\pi_Y(x)}{g_X(x)}$ , so gilt  $\tilde{h} = hW$  mit  $\tilde{h}(x) = h(x)W(x)$  für alle  $x \in \mathcal{X}$ . Die Funktion  $W$  wird dann als Gewichtsfunktion bezeichnet und Realisierungen  $w$  von  $W$  werden *Importance weights* genannt.

Damit lässt sich eine erwartungstreue Schätzung für  $\mu_\pi = E_\pi(h) = E_g(\tilde{h}) = E_g(hW)$  durch  $\widehat{\mu}_\pi = \frac{1}{n} \sum_{i=1}^n w_i h(x_i)$  geben, wobei die  $w_i$  durch  $w_i = \frac{\pi_Y(x_i)}{g_X(x_i)}$ , für  $1 \leq i \leq n$  gegeben sind.

Die Importance weights können allerdings nur exakt berechnet werden, wenn man  $\pi$  kennt. Daher ist die Schätzung  $\widehat{\mu}_\pi$  im Fall der Binärmatrizen nicht anwendbar, weshalb auf eine andere Schätzung zurückgegriffen werden muss.

**Bemerkung 4.3** (angelehnt an [20] S. 33)

Für zwei diskrete Dichten  $\pi_Y$  und  $g_X$  von diskreten Zufallsvariablen  $Y$  und  $X$  mit Werten in  $\mathcal{X}$  gilt stets

$$\sum_{x \in \mathcal{X}} \pi_Y(x) = \sum_{x \in \mathcal{X}} g_X(x) = 1. \quad (4.2)$$

Ist  $g_X(x) \neq 0$  für alle  $x \in \mathcal{X}$ , so ist die Zufallsvariable  $W : \mathcal{X} \rightarrow \mathbb{R}$  mit  $W(x) = \frac{\pi_Y(x)}{g_X(x)}$  wohldefiniert und es gilt für alle  $x \in \mathcal{X}$

$$E_g(W) = \sum_{x \in \mathcal{X}} \frac{\pi_Y(x)}{g_X(x)} g_X(x) = \sum_{x \in \mathcal{X}} \pi_Y(x) \stackrel{(4.2)}{=} 1. \quad (4.3)$$

Somit gilt für  $\mu_\pi = E_\pi(h)$  zu einer Funktion  $h : \mathcal{X} \rightarrow \mathbb{R}$  mit der Gewichtsfunktion  $W$

$$\mu_\pi = E_\pi(h) = E_g(hW) \stackrel{(4.3)}{=} \frac{E_g(hW)}{E_g(W)}.$$

Eine weitere Schätzung für  $\mu_\pi$  ist dann durch

$$\widehat{\mu}_\pi = \frac{\frac{1}{n} \sum_{i=1}^n w_i h(x_i)}{\frac{1}{n} \sum_{i=1}^n w_i} = \frac{\sum_{i=1}^n w_i h(x_i)}{\sum_{i=1}^n w_i}$$

möglich, wobei  $x_1, \dots, x_n$  unabhängige Realisierungen der gemäß  $g$  verteilten Zufallsvariablen  $X$  mit Werten in  $\mathcal{X}$  und  $w_i = \frac{\pi_Y(x_i)}{g_X(x_i)}$  für alle  $i \in \{1, \dots, n\}$  gilt. Angenommen man kennt  $\pi_Y(x_i)$  bis auf eine normierende Konstante  $c$ , die für alle  $\pi_Y(x_i)$  gleich ist. Dann gilt  $w_i = c \frac{\tilde{\pi}_Y(x_i)}{g_X(x_i)}$  mit  $\pi_Y(x_i) = c \tilde{\pi}_Y(x_i)$  und somit

$$\begin{aligned} \widehat{\mu}_\pi &= \frac{\sum_{i=1}^n w_i h(x_i)}{\sum_{i=1}^n w_i} \\ &= \frac{\sum_{i=1}^n c \frac{\tilde{\pi}_Y(x_i)}{g_X(x_i)} h(x_i)}{\sum_{i=1}^n c \frac{\tilde{\pi}_Y(x_i)}{g_X(x_i)}} \\ &= \frac{c \sum_{i=1}^n \frac{\tilde{\pi}_Y(x_i)}{g_X(x_i)} h(x_i)}{c \sum_{i=1}^n \frac{\tilde{\pi}_Y(x_i)}{g_X(x_i)}} \\ &= \frac{\sum_{i=1}^n \frac{\tilde{\pi}_Y(x_i)}{g_X(x_i)} h(x_i)}{\sum_{i=1}^n \frac{\tilde{\pi}_Y(x_i)}{g_X(x_i)}}. \end{aligned}$$

Für die Berechnung von  $\widehat{\mu}_\pi$  muss somit  $\pi_Y(x_i)$  für  $1 \leq i \leq n$  nur bis auf eine gemeinsame Konstante  $c$  bekannt sein. □

#### Beispiel 4.4

Ist  $\pi_Y(x) = \frac{1}{|\Sigma_{rc}|}$  für die Zufallsvariable  $Y$  die Gleichverteilung über  $\Sigma_{rc}$ , so ist  $c = \pi_Y(x)$  eine Konstante, die für alle  $x \in \Sigma_{rc}$  gleich ist. Nun kann man mit einer Stichprobe  $x_1, \dots, x_n$  von  $n$  unabhängigen Realisierungen einer geeigneten, gemäß einer Probeverteilung  $g$  verteilten Zufallsvariablen  $X$ , für die  $g_X(x_i) > 0$  für alle  $1 \leq i \leq n$  gilt,  $\mu_\pi = E_\pi(h)$  durch

$$\widehat{\mu}_\pi = \frac{\sum_{i=1}^n \frac{\pi_Y(x_i)}{g_X(x_i)} h(x_i)}{\sum_{i=1}^n \frac{\pi_Y(x_i)}{g_X(x_i)}} = \frac{c \sum_{i=1}^n \frac{1}{g_X(x_i)} h(x_i)}{c \sum_{i=1}^n \frac{1}{g_X(x_i)}} = \frac{\sum_{i=1}^n \frac{1}{g_X(x_i)} h(x_i)}{\sum_{i=1}^n \frac{1}{g_X(x_i)}}$$

schätzen und benötigt dafür keinerlei Kenntnis von  $\pi$ . □

Ein anderer Vorteil durch die Verwendung von  $\widehat{\mu}_\pi$  anstatt von  $\widetilde{\mu}_\pi$  entsteht dadurch, dass häufig der mittlere quadratische Fehler (MSE) verkleinert wird. Dies ist zum

Beispiel der Fall, wenn die Zufallsvariablen  $W$  und  $Z$  stark korreliert sind, wobei  $W(x) = \frac{\pi_Y(x)}{g_X(x)}$  und  $Z(x) = h(x) \frac{\pi_Y(x)}{g_X(x)}$  für alle  $x \in \mathcal{X}$  gelte. Eine genaue Gegenüberstellung der MSE von  $\widehat{\mu}_\pi$  und  $\widetilde{\mu}_\pi$  ist für die weiteren Betrachtungen irrelevant aber zum Beispiel in [20] nachzulesen.

## 4.2 Bestimmung der Güte beim Importance Sampling

Das größte Problem beim Importance Sampling ist das Finden einer geeigneten Probeverteilung  $g$ . Hat man eine Probeverteilung gefunden, so muss man sicherstellen, dass diese die Zielverteilung möglichst gut wiedergibt. Ein Maß, das die Effektivität des angewendeten Verfahrens wiedergibt, ist der in Definition 4.5 angegebene effektive Stichprobenumfang (engl.: effective sample size) kurz *ESS*.

**Definition 4.5 (ESS)** (nach [17] S. 284)

Sei  $\pi_Y$  die diskrete Dichte einer diskreten Zufallsvariablen  $Y$  mit Werten in  $\mathcal{X}$  und  $g$  eine mögliche Probeverteilung eines Importance Sampling Verfahrens. Hat man eine Stichprobe  $x = (x_1, \dots, x_n)$  mit  $n \in \mathbb{N}$  Elementen als unabhängige Realisierungen einer diskreten,  $g$ -verteilten Zufallsvariablen  $X$  genommen, so ist der effektive Stichprobenumfang durch

$$ESS = \frac{n}{1 + \text{Var}_g(W)} \in (0, n] \quad (4.4)$$

gegeben, wobei  $W(x) = \frac{\pi_Y(x)}{g_X(x)}$  für alle  $x \in \mathcal{X}$  ist. Je größer der ESS ist, desto besser ist die Probeverteilung  $g$ , wobei  $ESS = n$  bedeutet, dass  $g = \pi$  ist.

Wie aus Bemerkung 4.8 hervorgehen wird, beschreibt  $\text{Var}_g(W)$  dabei nichts anderes als den Chi-Quadrat-Abstand zwischen den Verteilungen  $\pi$  und  $g$  gemäß der folgenden Definition.

**Definition 4.6 (Chi-Quadrat-Abstand)** (nach [6] S. 280)

Es seien  $\pi$  und  $g$  die Verteilungen der diskreten Zufallsvariablen  $Y$  und  $X$ , wobei  $Y$  und  $X$  Werte in  $\mathcal{X}$  annehmen. Die jeweiligen diskreten Dichten seien für alle  $x \in \mathcal{X}$  durch  $\pi_Y(x)$  und  $g_X(x)$  gegeben. Es gelte  $g_X(x) \neq 0$  für alle  $x \in \mathcal{X}$ . Dann bezeichnet

$$\chi^2(\pi, g) = \sum_{x \in \mathcal{X}} \frac{(\pi_Y(x) - g_X(x))^2}{g_X(x)} \quad (4.5)$$

den Chi-Quadrat-Abstand zwischen den Verteilungen  $\pi$  und  $g$ .

Der Chi-Quadrat-Abstand beschreibt dabei keinen Abstand im metrischen Sinn, sondern gibt die Ähnlichkeit beider Verteilungen wieder. Je kleiner  $\chi^2(\pi, g)$  ist, desto ähnlicher sind sich die beiden Verteilungen  $\pi$  und  $g$ . Dabei bedeutet  $\chi^2(\pi, g) = 0$  die Gleichheit beider Verteilungen für alle  $x \in \mathcal{X}$ .

$Var_g(W)$  beziehungsweise  $\chi^2(\pi, g)$  setzt die Kenntnis sowohl von  $g$  als auch von  $\pi$  voraus. Da bei einem Importance Sampling Verfahren meist jedoch lediglich  $g$  bekannt ist, wird dafür noch eine weitere Größe, die bei der Berechnung von  $\chi^2(\pi, g)$  beziehungsweise  $Var_g(W)$  und  $ESS$  behilflich ist, benötigt.

**Definition 4.7 (Variationskoeffizient)** (nach [7] S. 111)

Sei  $X$  eine diskrete Zufallsvariable. Der Variationskoeffizient (engl.: coefficient of variation) kurz  $cv$  setzt nun die Standardabweichung von  $X$  in Bezug zu dem Erwartungswert von  $X$  und wird durch

$$cv = \frac{\sqrt{Var(X)}}{E(X)}. \quad (4.6)$$

definiert.

**Bemerkung 4.8** (nach [17] S. 284)

Es seien  $\pi$  und  $g$  Verteilungen der diskreten,  $\mathcal{X}$ -wertigen Zufallsvariablen  $Y$  und  $X$ , wobei  $g(X = x) \neq 0$  für alle  $x \in \mathcal{X}$  sei und  $\pi_Y(x)$  beziehungsweise  $g_X(x)$  die jeweiligen diskreten Dichten bezeichne. Mit der Zufallsvariablen  $W$  aus Definition 4.5 gilt dann

$$cv^2 \stackrel{(4.6)}{=} \frac{Var_g(W)}{(E_g(W))^2} \stackrel{(4.3)}{=} Var_g(W). \quad (4.7)$$

□

**Folgerung 4.9** (nach [7] S. 111)

Seien  $\pi$  und  $g$  die Verteilungen der diskreten Zufallsvariablen  $Y$  und  $X$  mit Werten in  $\mathcal{X}$ , wobei  $g(X = x) \neq 0$  für alle  $x \in \mathcal{X}$  sei und  $\pi_Y(x)$  beziehungsweise  $g_X(x)$  die jeweiligen diskreten Dichten bezeichne. Mit der Zufallsvariablen  $W$  aus Definition 4.5 gilt dann

$$cv^2 \stackrel{(4.6)}{=} \frac{Var_g(W)}{(E_g(W))^2} = \chi^2(\pi, g). \quad (4.8)$$

□

BEWEIS

Die Gleichung (4.8) ergibt sich durch einfaches nachrechnen. Es gilt zunächst

$$cv^2 \stackrel{(4.7)}{=} Var_g(W).$$

Damit folgt weiter

$$\begin{aligned}
 cv^2 &\stackrel{(2.8)}{=} E_g(W^2) - (E_g(W))^2 \\
 &\stackrel{(4.3)}{=} E_g(W^2) - 1 \\
 &\stackrel{(2.7)}{=} \left( \sum_{x \in \mathcal{X}} \frac{(\pi_Y(x))^2}{(g_X(x))^2} g_X(x) \right) - 1 \\
 &= \left( \sum_{x \in \mathcal{X}} \frac{(\pi_Y(x))^2}{g_X(x)} \right) - 1 \\
 &\stackrel{(4.2)}{=} \sum_{x \in \mathcal{X}} \frac{(\pi_Y(x))^2}{g_X(x)} - 2 \sum_{x \in \mathcal{X}} \pi_Y(x) + \sum_{x \in \mathcal{X}} g_X(x) \\
 &= \sum_{x \in \mathcal{X}} \left( \frac{(\pi_Y(x))^2}{g_X(x)} - 2 \frac{\pi_Y(x) g_X(x)}{g_X(x)} + \frac{(g_X(x))^2}{g_X(x)} \right) \\
 &= \sum_{x \in \mathcal{X}} \frac{(\pi_Y(x))^2 - 2\pi_Y(x)g_X(x) + (g_X(x))^2}{g_X(x)} \\
 &= \sum_{x \in \mathcal{X}} \frac{(\pi_Y(x) - g_X(x))^2}{g_X(x)} \\
 &\stackrel{(4.5)}{=} \chi^2(\pi, g). \quad \blacksquare
 \end{aligned}$$

Es ist also  $cv^2$  für  $W$  gleichbedeutend mit dem Chi-Quadrat-Abstand zwischen den Verteilungen  $\pi$  und  $g$ . Im Folgenden wird deshalb auch  $cv^2$  als Bezeichnung für den Chi-Quadrat-Abstand genutzt. Für eine Stichprobe der Größe  $n$  kann der *ESS* somit gleichbedeutend auch durch

$$ESS \stackrel{(4.7)}{=} \frac{n}{1 + cv^2}$$

beziehungsweise

$$ESS \stackrel{(4.8)}{=} \frac{n}{1 + \chi^2(\pi, g)}$$

definiert werden. Je kleiner  $cv^2$  für  $W$  beziehungsweise  $\chi^2(\pi, g)$  ist, um so größer ist demnach der *ESS*.

**Bemerkung 4.10** (angelehnt an [20] S. 33)

Seien  $\pi$  und  $g$  die Verteilungen zu den diskreten,  $\mathcal{X}$ -wertigen Zufallsvariablen  $Y$  und  $X$ , wobei die diskrete Dichte  $g_X(x)$  für alle  $x \in \mathcal{X}$  bekannt und  $g_X(x) \neq 0$  sei. Die diskrete Dichte  $\pi_Y(x)$  sei für alle  $x \in \mathcal{X}$  bis auf eine Konstante  $c$  bekannt. Es lässt sich dann die Berechnung von  $cv^2$  für die Zufallsvariable  $W$  aus Folgerung 4.9 vereinfachen.

Ist  $\pi_Y(x) = c \cdot \tilde{\pi}_Y(x)$  für alle  $x \in \mathcal{X}$ , so gilt

$$\begin{aligned}
 cv^2 &\stackrel{(4.6)}{=} \frac{\text{Var}_g(W)}{(E_g(W))^2} \\
 &\stackrel{(2.8)}{=} \frac{E_g(W^2) - (E_g(W))^2}{(E_g(W))^2} \\
 &\stackrel{(2.7)}{=} \frac{\sum_{x \in \mathcal{X}} \frac{\pi_Y(x)^2}{g_X(x)^2} g(x) - \left( \sum_{x \in \mathcal{X}} \frac{\pi_Y(x)}{g_X(x)} g(x) \right)^2}{\left( \sum_{x \in \mathcal{X}} \frac{\pi_Y(x)}{g_X(x)} g(x) \right)^2} \\
 &= \frac{\sum_{x \in \mathcal{X}} \frac{c^2 \tilde{\pi}_Y(x)^2}{g_X(x)^2} g(x) - \left( \sum_{x \in \mathcal{X}} \frac{c \tilde{\pi}_Y(x)}{g_X(x)} g(x) \right)^2}{\left( \sum_{x \in \mathcal{X}} \frac{c \tilde{\pi}_Y(x)}{g_X(x)} g(x) \right)^2} \\
 &= \frac{c^2 \left( \sum_{x \in \mathcal{X}} \frac{\tilde{\pi}_Y(x)^2}{g_X(x)^2} g(x) - \left( \sum_{x \in \mathcal{X}} \frac{\tilde{\pi}_Y(x)}{g_X(x)} g(x) \right)^2 \right)}{c^2 \left( \sum_{x \in \mathcal{X}} \frac{\tilde{\pi}_Y(x)}{g_X(x)} g(x) \right)^2} \\
 &= \frac{\text{Var}_g(\tilde{W})}{(E_g(\tilde{W}))^2}
 \end{aligned}$$

mit  $\tilde{W}(x) := \frac{\tilde{\pi}_Y(x)}{g_X(x)}$  für alle  $x \in \mathcal{X}$ .

$cv^2$  ist so zwar theoretisch immer noch eine unbekannte Größe, lässt sich aber bei einer gegebenen Stichprobe  $x = (x_1, \dots, x_n)$  der mittels der erwartungstreuen Schätzungen Stichprobenvarianz und arithmetisches Mittel durch

$$\widehat{cv^2} = \frac{S^2(\tilde{W}(x))}{(\overline{\tilde{W}(x)})^2} = \frac{\frac{1}{n-1} \sum_{i=1}^n \left( \frac{\tilde{\pi}(x_i)}{g(x_i)} - \frac{1}{n} \sum_{j=1}^n \frac{\tilde{\pi}(x_j)}{g(x_j)} \right)^2}{\left( \frac{1}{n} \sum_{i=1}^n \frac{\tilde{\pi}(x_i)}{g(x_i)} \right)^2} \quad (4.9)$$

schätzen, wobei  $x_1, \dots, x_n$  unabhängige Realisierungen von  $X$  sind. Damit kann nun die Güte einer Probeverteilung  $g$  mittels  $\widehat{cv^2}$  gemessen werden. Die Genauigkeit dieser Schätzung hängt dabei natürlich vom Stichprobenumfang  $n$  ab.  $\widehat{cv^2}$  ist dabei nicht robust gegen Ausreißer. Das heißt, ein einziger extrem von allen anderen abweichender Wert  $g(x_i)$  kann  $\widehat{cv^2}$  stark verändern. □

#### Beispiel 4.11

Sei  $\Sigma_{rc}$  die Menge aller Binärmatrizen mit ZSV  $r$  und SSV  $c$ . Ist in der Notation von Bemerkung 4.10  $\pi_Y(x) = \frac{1}{|\Sigma_{rc}|}$  für alle  $x \in \Sigma_{rc}$ , so ist  $\pi_Y(x)$  konstant für alle  $x \in \mathcal{X}$

und es lässt sich  $\tilde{\pi}_Y = 1$  setzen. Damit ist  $cv^2$  in dem Fall sehr leicht zu schätzen. Mit einer Probeverteilung  $g$  und einer Stichprobe  $x = (x_1, \dots, x_n)$ , die Realisierung einer diskreten,  $g$ -verteilten Zufallsvariable  $X$  ist, gilt dann

$$\widehat{cv^2} = \frac{\frac{1}{n-1} \sum_{i=1}^n \left( \frac{1}{g_X(x_i)} - \frac{1}{n} \sum_{j=1}^n \frac{1}{g_X(x_j)} \right)^2}{\left( \frac{1}{n} \sum_{i=1}^n \frac{1}{g_X(x_i)} \right)^2}.$$

Sei  $n = 10$ ,  $x = (x_1, \dots, x_{10})$  und

$$(g_X(x_1), \dots, g_X(x_{10})) = (0.01, 0.02, 0.01, 0.02, 0.01, 0.02, 0.01, 0.02, 0.01, 0.02).$$

Es ist dann  $\widehat{cv^2} = 0.1235$  und  $ESS = 8.9011$ . Da  $ESS \in (0, 10]$  gilt, ist die Verteilung  $g$  für diese Stichprobe nahe am Optimum. Ist hingegen  $g_X(x_1) = 0.0001$ , so ergibt sich als Schätzung  $\widehat{cv^2} = 8.6902$  und  $ESS = 1.0320$ . In diesem Fall ist der  $ESS$  sehr klein und die Probeverteilung  $g$  für die Generierung einer Stichprobe nicht geeignet.  $\square$

### 4.3 Sequentielles Importance Sampling

Das größte Problem eines Importance Sampling Verfahrens ist, eine geeignete Probeverteilung  $g$ , mittels der dann eine Stichprobe erzeugt werden kann. Eine Möglichkeit ist,  $g$  sequentiell, also in mehreren Schritten, zu ermitteln. Dabei macht man sich die Eigenschaft (2.5) zu nutze. Ist es möglich  $g$  aufzuteilen, so lässt sich  $g$  eventuell leichter bestimmen. Dieses Vorgehen nennt man dann sequentielles Importance Sampling.

**Bemerkung 4.12** (nach [20] S. 46)

Sei  $X = (X_1, \dots, X_N)$  ein diskreter,  $N$ -dimensionaler Zufallsvektor mit Werten in  $\mathcal{X} = \otimes_{i=1}^N \mathcal{X}_i$  und Verteilung  $g$ , so kann man (2.5) anwenden. Es ist dann für alle  $x \in \mathcal{X}$

$$\begin{aligned} g(X = (x_1, \dots, x_N)) &= g(X_1 = x_1)g(X_2 = x_2 \mid X_1 = x_1) \\ &\quad g(X_3 = x_3 \mid X_1 = x_1, X_2 = x_2) \\ &\quad \dots \\ &\quad g(X_N = x_N \mid X_1 = x_1, \dots, X_{N-1} = x_{N-1}). \end{aligned} \tag{4.10}$$

Ist es möglich, Verteilungen  $g_1, \dots, g_N$  zu den Zufallsvariablen  $X_1, \dots, X_N$  zu finden,

für die

$$\begin{aligned}
 g_1(X_1 = x_1) &= g(X_1 = x_1), \\
 g_2(X_2 = x_2) &= g(X_2 = x_2 \mid X_1 = x_1), \\
 &\vdots \\
 g_N(X_N = x_N) &= g(X_N = x_N \mid X_1 = x_1, \dots, X_{N-1} = x_{N-1})
 \end{aligned}$$

gilt,  $g(X = x)$  für alle  $x \in \mathcal{X}$  durch

$$g(X = x) = \prod_{i=1}^N g_i(X_i = x_i) \tag{4.11}$$

zu bestimmen. □

**Beispiel 4.13**

Angenommen es ist  $\mathcal{X} = \Sigma_{rc}$  mit  $r \in \{1, \dots, n\}^m$ ,  $c \in \{1, \dots, m\}^n$  und  $m, n \in \mathbb{N}$ . Es bezeichne  $x_1, \dots, x_n$  die Spalten von  $x \in \mathcal{X}$ . Angenommen es ist  $g$  eine Probeverteilung für ein Importance Sampling Verfahren zur Generierung unabhängiger, auf  $\Sigma_{rc}$  gleichverteilter Binärmatrizen. Es bezeichne  $g_1$  eine Verteilung, die die Wahrscheinlichkeit für die erste Spalte  $x_1$  von  $x$  bei den gegebenen Zeilen- und Spaltensummen aufzutreten wiedergibt. Hat man dann die erste Spalte  $x_1$  mittels  $g_1$  generiert, betrachtet man nur noch die Spalten 2 bis  $n$  von  $x$  mit den entsprechend der in  $x_1$  auftretenden Einsen geänderten Zeilen- und Spaltensummen. Es bezeichne dann  $g_2$  eine Verteilung, die die Wahrscheinlichkeit für  $x_2$  wiedergibt, bei den nun auch von  $x_1$  abhängenden gegebenen Zeilen- und Spaltensummen aufzutreten. Führt man dieses Verfahren fort, erhält man die Verteilungen  $g_1, \dots, g_n$  und kann  $g$  entsprechend (4.11) durch  $g(X = x) = \prod_{i=1}^n g_i(X_i = x_i)$  berechnen. Es bleibt dabei dennoch die Frage nach geeigneten Verteilungen  $g_1, \dots, g_n$ . Wie sich in Kapitel 5 zeigen wird, ist die dort erläuterte bedingte Bernoulli Verteilung eine geeignete Verteilungsfunktion, mittels der man  $g$  im Falle von Binärmatrizen mit gegebenen ZSV und SSV spaltenweise erschließen kann. □

# 5 Bedingte Bernoulli-Modelle

Wendet man Importance Sampling an, um Binärmatrizen sequentiell, also spaltenweise zu generieren, bleibt die Frage, nach welcher Verteilung man die einzelnen Spalten generieren sollte. Ist  $N$  die Anzahl der Zeilen der zu generierenden Matrizen, so sei  $\Omega = \{1, \dots, N\}$ . Ist  $n$  die Spaltensumme der zu generierenden Spalte, so benötigt man nun ein Verfahren zur Ziehung von  $n$  Einheiten aus der Population  $\Omega$  ohne Zurücklegen. Da man tendenziell eher eine Eins in eine Zeile mit großer Zeilensumme eintragen möchte, muss diese Ziehung mit unterschiedlichen Wahrscheinlichkeiten erfolgen, also die einzelnen Ereignisse unterschiedlich gewichtet sein. Solche Ziehungen werden durch bedingte Bernoulli-Modelle beschrieben.

## 5.1 Bedingte Bernoulli-Modelle

**Definition 5.1 (Bedingte Bernoulli-Verteilung)** (nach [4] S. 875)

Sei  $Z = (Z_1, \dots, Z_N)$  Zufallsvektor aus unabhängigen, Bernoulli-verteilten Zufallsvariablen mit den Erfolgswahrscheinlichkeiten  $p_1, \dots, p_N$ . Die Verteilung von  $Z$  unter der Voraussetzung, dass  $\sum_{i=1}^N Z_i = n$  gilt, heißt bedingte Bernoulli-Verteilung.

Es sei  $\Omega = \{1, \dots, N\}$  eine endliche Population der Größe  $N$  und  $s$  eine ungeordnete Stichprobe ohne Zurücklegen der Größe  $n$  aus  $\Omega$ .  $s = (s_1, \dots, s_n)$  sei eine Realisierung der Zufallsvariablen  $Y = (Y_1, \dots, Y_n)$ .  $Y$  lässt sich auch mit Hilfe von Indikator-Variablen beschreiben. Es sei

$$X = (X_1, \dots, X_N),$$

wobei  $X_i : \{i\} \rightarrow \{0, 1\}$  für  $i \in \Omega$  mit

$$X_i(i) = \begin{cases} 1 & \text{falls } Y_j = i \text{ für ein } j \in \{1, \dots, n\} \\ 0 & \text{sonst.} \end{cases} \quad (5.1)$$

$X_i$  gibt dabei an, ob die  $i$ -te Einheit der Menge  $\Omega$  in der Stichprobe  $s$  ist oder nicht. Eine Realisierung  $s^I = (s_1^I, \dots, s_N^I)$  von  $X$  wird von nun an Indikatorstichprobe zu  $s = (s_1, \dots, s_n)$  genannt.

Es sei

$$D^n = \left\{ x = (x_1, \dots, x_N) \mid x_i \in \{0, 1\} \text{ und } \sum_{i=1}^N x_i = n \right\} \quad (5.2)$$

der Stichprobenraum aller Indikatorstichproben, die zu ungeordneten Stichproben  $s$  ohne Zurücklegen von  $n$  Elementen aus der Population  $\Omega$  gebildet werden können.

Die Anzahl der Elemente von  $D^n$  ist dabei  $\binom{N}{n} = \frac{N!}{(N-n)!n!}$ .

Soll  $s$  eine ungeordnete Stichprobe von  $n$  Elementen der Population  $\Omega$  sein, so nimmt die Indikatorvariable  $X$  Werte in  $D^n$  an. Für ein beliebiges Stichprobenverfahren kann nun die dazu gehörige Dichte hergeleitet werden. Es bezeichne  $p_X(x)$  eine solche Dichte für beliebige  $x \in D^n$ . Es soll dabei stets

$$p_X(x) > 0 \quad (5.3)$$

für alle  $x \in D^n$  gelten, damit der Stichprobenraum  $D^n$  voll ausgeschöpft wird. Nach Satz 2.6 muss außerdem

$$\sum_{x \in D^n} p_X(x) = 1 \quad (5.4)$$

gelten. Durch jede Dichtefunktion, die (5.3) und (5.4) genügt, ist eine Dichte für ein bedingtes Bernoulli-Modell gegeben.

Es ist

$$E(X_i) = \sum_{x \in D^n} x_i p_X(x). \quad (5.5)$$

die Wahrscheinlichkeit dafür, dass die  $i$ -te Einheit von  $\Omega$  in die Stichprobe aufgenommen wird.

Um die Dichte  $p_X(\cdot)$  eines bedingten Bernoulli-Modells zu bestimmen, werden nun drei verschiedene Methoden angegeben, die jeweils unterschiedliche Voraussetzungen benötigen. Bei der ersten Methode werden geeignete *Gewichte* zum jeweiligen Bernoulli-Modell erschlossen, um die Dichte zu bestimmen. Bei der zweiten Methode wird die Dichte anhand von Erfolgswahrscheinlichkeiten von geeigneten Bernoulli-Experimenten bestimmt. Für die dritte Methode wird die sogenannte Entropie maximiert.

**Methode 1** (nach [5] S. 458)

Es sei  $w = (w_1, \dots, w_N)$  ein beliebiger Vektor von Gewichten, wobei  $w_i > 0$  für alle  $i \in \Omega = \{1, \dots, N\}$  ist. Dann ist durch

$$p_X^{(1)}(x) := \frac{\prod_{i \in \Omega} w_i^{x_i}}{\sum_{y \in D^n} \left( \prod_{i \in \Omega} w_i^{y_i} \right)} \quad \text{für } x \in D^n \quad (5.6)$$

eine Dichte für ein bedingtes Bernoulli-Modell gegeben. Es sind nämlich (5.3) und (5.4) erfüllt, da zum einen für  $w_i > 0$  mit  $i \in \Omega$  stets  $p_X^{(1)}(x) > 0$  für alle  $x \in D^n$  ist und zum anderen

$$\sum_{x \in D^n} p_X^{(1)}(x) = \frac{\sum_{x \in D^n} \left( \prod_{i \in \Omega} w_i^{x_i} \right)}{\sum_{y \in D^n} \left( \prod_{i \in \Omega} w_i^{y_i} \right)} = 1$$

gilt. □

Am Ende dieses Abschnitts wird noch auf die richtige Wahl der Gewichte eingegangen.

**Methode 2** (nach [5] S. 458)

Es seien  $p_1, \dots, p_N$  beliebige Erfolgswahrscheinlichkeiten von  $N$  unabhängigen Bernoulli-Experimenten  $Z_1, \dots, Z_N$  mit  $0 < p_i < 1$  für alle  $i \in \Omega = \{1, \dots, N\}$  sind. Man setze nun  $Z = (Z_1, \dots, Z_N)$ . Sei  $x = (x_1, \dots, x_N) \in \{0, 1\}^N$  eine Realisierung von  $Z$ . Aufgrund der Unabhängigkeit von  $Z_1, \dots, Z_N$  ist dann

$$\begin{aligned} P(Z_1 = x_1, Z_2 = x_2, \dots, Z_N = x_N) &\stackrel{\text{Unabh.}}{=} \prod_{i \in \Omega} P(Z_i = x_i) \\ &\stackrel{(2.10)}{=} \prod_{i \in \Omega} p_i^{x_i} (1 - p_i)^{1-x_i} \\ &= \prod_{i \in \Omega} p_i^{x_i} (1 - p_i)^{-x_i} (1 - p_i) \\ &= \prod_{i \in \Omega} (1 - p_i) \prod_{i \in \Omega} \left( \frac{p_i}{(1 - p_i)} \right)^{x_i}. \end{aligned}$$

Man setze nun

$$\prod_{i \in \Omega} (1 - p_i) \prod_{i \in \Omega} \left( \frac{p_i}{(1 - p_i)} \right)^{x_i} =: p_Z(x). \quad (5.7)$$

Damit und mit  $S_Z := \sum_{i \in \Omega} Z_i$  gilt dann

$$\begin{aligned} P(S_Z = n) &= P\left(\sum_{i \in \Omega} Z_i = n\right) \\ &\stackrel{(5.2)}{=} \sum_{x \in D^n} p_Z(x) \\ &\stackrel{(5.7)}{=} \sum_{x \in D^n} \prod_{i \in \Omega} (1 - p_i) \prod_{i \in \Omega} \left( \frac{p_i}{(1 - p_i)} \right)^{x_i} \\ &= \prod_{i \in \Omega} (1 - p_i) \sum_{x \in D^n} \prod_{i \in \Omega} \left( \frac{p_i}{(1 - p_i)} \right)^{x_i} =: p_{S_Z}(n). \end{aligned}$$

Nach Definition 2.14 beschreibt  $p_{(Z, S_Z)}(x, n)$  die gemeinsame Dichte von  $Z$  und  $S_Z$ . Ist  $x \notin D^n$ , so ist  $p_{S_Z} = 0$  und damit offensichtlich  $p_{(Z, S_Z)}(x, n) = 0$ . Für  $x \in D^n$  ist

$p_{S_Z} = 1$  und damit  $p_{(Z, S_Z)}(x, n) = p_Z(x)$ . Es sei

$$\begin{aligned}
 p_X^{(2)}(x) &:= p_{Z|S_Z=n}(x) \\
 &\stackrel{(2.6)}{=} \frac{p_{(Z, S_Z)}(x, n)}{p_{S_Z}(n)} \\
 &= \frac{\prod_{i \in \Omega} (1 - p_i) \prod_{i \in \Omega} \left(\frac{p_i}{(1-p_i)}\right)^{x_i}}{\prod_{i \in \Omega} (1 - p_i) \sum_{y \in D^n} \left(\prod_{i \in \Omega} \left(\frac{p_i}{(1-p_i)}\right)^{y_i}\right)} \\
 &= \frac{\prod_{i \in \Omega} \left(\frac{p_i}{(1-p_i)}\right)^{x_i}}{\sum_{y \in D^n} \left(\prod_{i \in \Omega} \left(\frac{p_i}{(1-p_i)}\right)^{y_i}\right)}. \tag{5.8}
 \end{aligned}$$

Da nach Voraussetzung  $p_i > 0$  für alle  $i \in \Omega$  ist, ist (5.3) für  $p_X^{(2)}$  erfüllt. Außerdem gilt

$$\sum_{x \in D^n} p_X^{(2)}(x) = \frac{\sum_{x \in D^n} \left(\prod_{i \in \Omega} \left(\frac{p_i}{(1-p_i)}\right)^{x_i}\right)}{\sum_{y \in D^n} \left(\prod_{i \in \Omega} \left(\frac{p_i}{(1-p_i)}\right)^{y_i}\right)} = 1,$$

womit auch (5.4) erfüllt ist. Durch  $p_X^{(2)}(\cdot)$  ist somit für  $x \in D^n$  eine Dichte für ein bedingtes Bernoulli-Modell gegeben.  $\square$

Für die dritte Methode sind noch weitere Voraussetzungen nötig, sie basiert auf der Entropie einer Verteilung. Die Entropie bezeichnet den Informationsgehalt.

**Definition 5.2 (Entropie)** (angelehnt an [30] S. 981)

Sei  $P$  ein Wahrscheinlichkeitsmaß über der diskreten Menge  $\Omega$  mit existierender diskreter Dichte  $p$ . Die Entropie von  $P$  wird durch

$$H(P) := - \sum_{\omega \in \Omega} p(\omega) \ln p(\omega) \tag{5.9}$$

beschrieben.

**Definition 5.3 (Relative Entropie)** (nach [14] S. 211)

Seien  $P$  und  $Q$  zwei Wahrscheinlichkeitsmaße über der diskreten Menge  $\Omega$  mit existierenden diskreten Dichten  $p$  und  $q$ . Es bezeichne

$$H(P, Q) := \begin{cases} \sum_{\omega \in \Omega} p(\omega) \ln \left(\frac{p(\omega)}{q(\omega)}\right) & \text{falls } q(\omega) \neq 0 \text{ für alle } \omega \in \Omega \\ \infty & \text{sonst.} \end{cases} \tag{5.10}$$

$H(P, Q)$  heißt relative Entropie oder Kullback-Leibler-Divergenz von  $P$  zu  $Q$ .

**Lemma 5.4** (nach [14] S. 211)

Seien  $P$  und  $Q$  Wahrscheinlichkeitsmaße über der diskreten Menge  $\Omega$  mit existierenden diskreten Dichten  $p$  und  $q$ . Dann ist  $H(P, Q)$  wohldefiniert. Das heißt, es gilt

$$H(P, Q) \geq 0 \tag{5.11}$$

und

$$H(P, Q) = 0 \Leftrightarrow p(\omega) = q(\omega) \quad \text{für alle } \omega \in \Omega. \quad \square$$

BEWEIS (nach [14] S. 211)

Für alle  $\omega \in \Omega$  sei

$$f(\omega) := \begin{cases} \frac{p(\omega)}{q(\omega)} & \text{für } q(\omega) > 0, \\ 1 & \text{für } q(\omega) = 0. \end{cases}$$

Es gilt  $q(\omega) \cdot f(\omega) \in [0, 1]$  für alle  $\omega \in \Omega$  und  $\sum_{\omega \in \Omega} q(\omega) \cdot f(\omega) = \sum_{\omega \in \Omega} p(\omega) = 1$ , somit ist auch  $q \cdot f$  eine diskrete Dichte von  $P$  und man kann  $p = q \cdot f$  annehmen. Sei  $g(x) := 1 - x + x \ln(x)$  für beliebige  $x \in \mathbb{R}_{\geq 0}$ , wobei  $0 \ln(0) = 0$  gelte.

Wie sich leicht nachrechnen lässt, ist  $\min_{x \in \mathbb{R}_{\geq 0}}(g(x)) = g(1) = 0$  das globale Minimum von  $g$  und somit gilt  $g(x) \geq 0$  für beliebige  $x \in \mathbb{R}_{\geq 0}$ . Somit gilt aber auch  $\sum_{\omega \in \Omega} g(f(\omega))q(\omega) \geq 0$ . Daraus folgt

$$\begin{aligned} 0 &\leq \sum_{\omega \in \Omega} (g(f(\omega))q(\omega)) \\ &= \sum_{\omega \in \Omega} ((1 - f(\omega) + f(\omega) \ln(f(\omega))) q(\omega)) \\ &= \sum_{\omega \in \Omega} (q(\omega) - f(\omega)q(\omega) + f(\omega) \ln(f(\omega)) q(\omega)) \\ &= \sum_{\omega \in \Omega} \left( q(\omega) - \frac{p(\omega)}{q(\omega)} q(\omega) + \frac{p(\omega)}{q(\omega)} q(\omega) \ln \left( \frac{p(\omega)}{q(\omega)} \right) \right) \\ &= \sum_{\omega \in \Omega} q(\omega) - \sum_{\omega \in \Omega} p(\omega) + \sum_{\omega \in \Omega} p(\omega) \ln \left( \frac{p(\omega)}{q(\omega)} \right) \\ &= 1 - 1 + \sum_{\omega \in \Omega} p(\omega) \ln \left( \frac{p(\omega)}{q(\omega)} \right) \\ &= \sum_{\omega \in \Omega} p(\omega) \ln \left( \frac{p(\omega)}{q(\omega)} \right) = H(P, Q). \end{aligned}$$

Ist  $H(P, Q) = 0$ , so gilt demnach auch

$$\sum_{\omega \in \Omega} g(f(\omega))q(\omega) = 0. \tag{5.12}$$

Da aber sowohl  $g(x) \geq 0$  für beliebige  $x \in \mathbb{R}_{\geq 0}$  als auch  $q(\omega) \geq 0$  für alle  $\omega \in \Omega$

gilt, muss in (5.12) für jeden Summanden  $g(f(\omega))q(\omega) = 0$  gelten. Nun ist die Eins die einzige Nullstelle von  $g$  und es muss somit  $f(\omega) = 1$  für alle  $\omega \in \Omega$  gelten. Dies ist entweder der Fall, wenn  $q(\omega) = 0$  oder  $q(\omega) = p(\omega)$  für  $\omega \in \Omega$  ist. Da aber  $\sum_{\omega \in \Omega} q(\omega) = 1 = \sum_{\omega \in \Omega} p(\omega)$  gilt, folgt  $p(\omega) = q(\omega)$  für alle  $\omega \in \Omega$ . ■

Da  $H(P, Q) = 0$  nur gilt, wenn  $p(\omega) = q(\omega)$  für alle  $\omega \in \Omega$  ist, wird  $H(P, Q)$  manchmal irritierender Weise auch Kullback-Leibler-Abstand genannt, obwohl es sich nicht um eine Abstandsmetrik handelt, da  $H(P, Q)$  nicht symmetrisch ist.

Es sei  $\Omega = \{1, \dots, N\}$  und  $\pi = (\pi_1, \dots, \pi_N)$  ein Vektor von Wahrscheinlichkeiten, wobei  $\pi_i$  für  $i \in \Omega$  die Wahrscheinlichkeit bezeichne, dass die  $i$ -te Einheit von  $\Omega$  in eine Stichprobe  $s$  der Länge  $n$  aufgenommen wird. Es muss dabei

$$1) \quad 0 < \pi_i < 1 \quad \text{für alle } i \in \Omega \quad (5.13)$$

$$2) \quad \sum_{i \in \Omega} \pi_i = n \quad (5.14)$$

gelten. Ist  $s = (s_1, \dots, s_n)$  Realisierung eines Zufallsvektors  $Y = (Y_1, \dots, Y_n)$  und ist  $X = (X_1, \dots, X_N)$  wie in (5.1), so sei

$$\pi_i = E(X_i).$$

Für eine geeignete diskrete Dichte  $p$  für ein bedingtes Bernoulli-Modell gelte dann

$$\pi_i \stackrel{(5.5)}{=} \sum_{x \in D^n} x_i p(x) \quad \text{für alle } i \in \Omega. \quad (5.15)$$

Eine Möglichkeit, eine solche geeignete diskrete Dichte  $p$  zu finden, besteht darin, die Entropie zu maximieren. Man wähle dann diejenige diskrete Dichte  $p^*$ , für deren Wahrscheinlichkeitsmaß  $P^*$  die Entropie  $H(P^*)$  maximal unter allen möglichen Wahrscheinlichkeitsmaßen ist, für deren diskrete Dichte (5.15) erfüllt ist.

**Satz 5.5** (nach [30] S. 981)

Die diskrete Dichte  $p^*$  des Wahrscheinlichkeitsmaßes  $P^*$  über der Menge  $D^n$  aus (5.2), das  $H(P)$  über alle möglichen Wahrscheinlichkeitsmaße maximiert, deren diskrete Dichten Eigenschaft (5.15) erfüllen, ist für alle  $x \in D^n$  durch

$$p^*(x) = c \prod_{i=1}^N w_i^{x_i} \quad (5.16)$$

gegeben, wobei  $c = \exp(\lambda_0)$  und  $w_i = \exp(\lambda_i)$  für  $i \in \{1, \dots, N\}$  ist.  $\lambda_0, \dots, \lambda_N$  beziehungsweise  $c$  und  $w_1, \dots, w_N$  seien dabei so gewählt, dass Eigenschaft (5.15) und  $\sum_{x \in D^n} p^*(x) = 1$  gilt.

BEWEIS (nach [30] S. 982)

Es gilt zunächst

$$\begin{aligned}
 c \prod_{i=1}^N w_i^{x_i} &= \exp(\lambda_0) \prod_{i=1}^N \exp(\lambda_i)^{x_i} \\
 &= \exp(\lambda_0) \prod_{i=1}^N \exp(\lambda_i x_i) \\
 &= \exp(\lambda_0) \exp\left(\sum_{i=1}^N \lambda_i x_i\right) \\
 &= \exp\left(\lambda_0 + \sum_{i=1}^N \lambda_i x_i\right). \tag{5.17}
 \end{aligned}$$

Angenommen es ist  $q$  eine andere diskrete Dichte, die den Anforderungen (5.15) und  $\sum_{x \in D^n} q(x) = 1$  genügt, dann bleibt zu zeigen, dass  $H(Q) \leq H(P^*)$  für das zu  $q$  gehörende Wahrscheinlichkeitsmaß  $Q$  gilt. Es gilt

$$\begin{aligned}
 0 &\stackrel{(5.11)}{\leq} H(Q, P^*) \\
 &= \sum_{x \in D^n} q(x) \ln\left(\frac{q(x)}{p^*(x)}\right) \\
 &= \sum_{x \in D^n} q(x) \left(\ln(q(x)) + \ln\left(\frac{1}{p^*(x)}\right)\right) \\
 &= \sum_{x \in D^n} q(x) \ln(q(x)) - \sum_{x \in D^n} q(x) \ln(p^*(x)).
 \end{aligned}$$

Somit folgt

$$- \sum_{x \in D^n} q(x) \ln(q(x)) \leq - \sum_{x \in D^n} q(x) \ln(p^*(x)). \tag{5.18}$$

Es gilt für alle diskreten Dichten  $q$  über  $D^n$ , wenn die Anforderung (5.15) erfüllt ist,

$$\begin{aligned}
 \sum_{x \in D^n} q(x) \left(\lambda_0 + \sum_{i=1}^N \lambda_i x_i\right) &= \sum_{x \in D^n} \left(q(x) \lambda_0 + \sum_{i=1}^N \lambda_i q(x) x_i\right) \\
 &= \lambda_0 \sum_{x \in D^n} q(x) + \sum_{x \in D^n} \sum_{i=1}^N \lambda_i q(x) x_i \\
 &= \lambda_0 + \sum_{i=1}^N \sum_{x \in D^n} \lambda_i q(x) x_i \\
 &= \lambda_0 + \sum_{i=1}^N \lambda_i \sum_{x \in D^n} q(x) x_i \\
 &\stackrel{(5.15)}{=} \lambda_0 + \sum_{i=1}^N \lambda_i \pi_i. \tag{5.19}
 \end{aligned}$$

Die Gleichung (5.19) hängt dabei nicht von der Wahl einer bestimmten diskreten Dichte  $q$  über  $D^n$  ab. Somit gilt die Gleichung (5.19) auch für  $p^*$  und es ergibt sich

$$\begin{aligned}
 H(Q) &\stackrel{(5.9)}{=} - \sum_{x \in D^n} q(x) \ln(q(x)) \\
 &\stackrel{(5.18)}{\leq} - \sum_{x \in D^n} q(x) \ln(p^*(x)) \\
 &\stackrel{(5.17)}{=} - \sum_{x \in D^n} q(x) \left( \lambda_0 + \sum_{i=1}^N \lambda_i x_i \right) \\
 &\stackrel{(5.19)}{=} - \left( \lambda_0 + \sum_{i=1}^N \lambda_i \pi_i \right) \\
 &\stackrel{(5.19)}{=} - \sum_{x \in D^n} p^*(x) \left( \lambda_0 + \sum_{i=1}^N \lambda_i x_i \right) \\
 &\stackrel{(5.17)}{=} - \sum_{x \in D^n} p^*(x) \ln(p^*(x)) = H(P^*) \quad \blacksquare
 \end{aligned}$$

Es folgt die dritte Methode zur Ermittlung einer Dichte zu einem bedingten Bernoulli-Modell.

**Methode 3** (nach [5] S. 458)

Es sei  $\Omega = \{1, \dots, N\}$ . Aus der Gleichung (5.16) folgt, dass diejenige diskrete Dichte über  $D^n$  mit der maximalen Entropie proportional zu  $\prod_{i \in \Omega} w_i^{x_i}$  sein muss, wobei  $w_i$  für  $i \in \Omega$  wie in Satz 5.5 gewählt sei. Man setze für alle  $x \in D^n$  mit  $n \in \Omega$

$$p_X^{(3)}(x) := c \prod_{i \in \Omega} w_i^{x_i}. \quad (5.20)$$

Mit (5.20) und da  $\sum_{x \in D^n} p_X^{(3)}(x) = 1$  gelten muss, damit die Forderung (5.4) erfüllt ist, folgt nun

$$\begin{aligned}
 &\sum_{x \in D^n} p_X^{(3)}(x) = 1 \\
 \stackrel{(5.20)}{\Rightarrow} &\sum_{x \in D^n} \left( c \prod_{i \in \Omega} w_i^{x_i} \right) = 1 \\
 \Rightarrow &c \sum_{x \in D^n} \left( \prod_{i \in \Omega} w_i^{x_i} \right) = 1 \\
 \Rightarrow &c = \frac{1}{\sum_{x \in D^n} \prod_{i \in \Omega} w_i^{x_i}}.
 \end{aligned}$$

Somit gilt für alle  $x \in D^n$

$$p_X^{(3)}(x) = \frac{\prod_{i \in \Omega} w_i^{x_i}}{\sum_{y \in D^n} \prod_{i \in \Omega} w_i^{y_i}}. \quad (5.21)$$

Es folgt nun

$$\pi_i \stackrel{(5.15)}{=} \sum_{x \in D^n} x_i p_X^{(3)}(x) = \sum_{x \in D^n} x_i \frac{\prod_{j \in \Omega} w_j^{x_j}}{\sum_{y \in D^n} \prod_{j \in \Omega} w_j^{y_j}}. \quad (5.22)$$

Die Gewichte  $w_1, \dots, w_N$  sind damit durch  $\pi_1, \dots, \pi_N$  mittels (5.15) eindeutig bestimmt. Mit Hilfe von (2.22) lässt sich (5.22) auch durch R-Funktionen beschreiben. Sei  $D^n$  wie in (5.2) mit  $n \in \Omega$ . Es lässt sich (2.22) dann für einen beliebigen Vektor  $w = (w_1, \dots, w_N)$  zu

$$R(n, \Omega) \stackrel{(2.22)}{=} \sum_{\substack{B \subset \Omega \\ |B|=n}} \left( \prod_{i \in B} w_i \right) = \sum_{x \in D^n} \prod_{i \in \Omega} w_i^{x_i}. \quad (5.23)$$

umformen. Mit dieser Schreibweise gilt

$$\begin{aligned} \pi_i &\stackrel{(5.22)}{=} \sum_{x \in D^n} x_i \frac{\prod_{j \in \Omega} w_j^{x_j}}{\sum_{y \in D^n} \prod_{j \in \Omega} w_j^{y_j}} \\ &\stackrel{(5.23)}{=} \frac{\sum_{x \in D^n} x_i \prod_{j \in \Omega} w_j^{x_j}}{R(n, \Omega)} \\ &= \frac{\sum_{x \in D^n} x_i w_i^{x_i} \prod_{j \in \Omega \setminus \{i\}} w_j^{x_j}}{R(n, \Omega)}. \end{aligned}$$

Nun können sämtliche Summanden aussortiert werden, in denen  $x_i = 0$  ist, denn diese sind gerade gleich Null. Damit gilt dann

$$\begin{aligned} \pi_i &= \frac{\sum_{\substack{x \in D^n \\ x_i=1}} x_i w_i^{x_i} \prod_{j \in \Omega \setminus \{i\}} w_j^{x_j}}{R(n, \Omega)} \\ &= \frac{w_i \sum_{x \in D^{n-1}} \prod_{j \in \Omega \setminus \{i\}} w_j^{x_j}}{R(n, \Omega)} \\ &\stackrel{(5.23)}{=} \frac{w_i R(n-1, \Omega \setminus \{i\})}{R(n, \Omega)}. \end{aligned} \quad (5.24)$$

Es ist weiterhin, Forderung (5.14) erfüllt. Dies folgt direkt mit der Eigenschaft (2.23).

Es ist dann nämlich für  $\Omega = \{1, \dots, N\}$  und  $n \in \Omega$

$$\sum_{i \in \Omega} w_i R(n-1, \Omega \setminus \{i\}) = nR(n, \Omega) \quad (5.25)$$

und somit

$$\begin{aligned} \sum_{i \in \Omega} \pi_i &\stackrel{(5.24)}{=} \sum_{i \in \Omega} \frac{w_i R(n-1, \Omega \setminus \{i\})}{R(n, \Omega)} \\ &\stackrel{(5.25)}{=} \frac{nR(n, \Omega)}{R(n, \Omega)} = n. \end{aligned}$$

Es bleibt zu zeigen, dass durch  $p_X^{(3)}(\cdot)$  tatsächlich eine diskrete Dichte für ein bedingtes Bernoulli-Modell gegeben ist.

Da die Werte der Exponentialfunktion echt positiv sind, ist auch  $w_i > 0$  für alle  $i \in \Omega$  im Sinne von Satz 5.5. Demnach gilt auch

$$p_X^{(3)}(x) = \frac{\prod_{i \in \Omega} w_i^{x_i}}{\sum_{y \in D^n} \prod_{i \in \Omega} w_i^{y_i}} > 0$$

für alle  $x \in D^n$  und Forderung (5.3) ist erfüllt. Weiter gilt

$$\sum_{x \in D^n} p_X^{(3)}(x) = \frac{\sum_{x \in D^n} \left( \prod_{i \in \Omega} w_i^{x_i} \right)}{\sum_{y \in D^n} \left( \prod_{i \in \Omega} w_i^{y_i} \right)} = 1,$$

und somit ist auch Forderung (5.4) erfüllt.  $\square$

**Folgerung 5.6** (nach [5] S. 458)

Man beachte, dass Methode 1 und Methode 2 genau dann dieselbe Dichte liefern, wenn die Gewichte  $w_i$  für  $i \in \Omega = \{1, \dots, N\}$  in Methode 1 als

$$w_i := \frac{p_i}{1 - p_i} \quad \text{für } i \in \Omega \quad (5.26)$$

gewählt werden.

Methode 1 und Methode 3 liefern bei gleicher Wahl der Gewichte  $w_i$  dieselbe diskrete Dichte.  $\square$

BEWEIS

Der Beweis ergibt sich durch einfaches Umformen.

Gilt (5.26), so ist für  $x \in D^n$

$$\begin{aligned}
 p_X^{(1)}(x) &\stackrel{(5.6)}{=} \frac{\prod_{i \in \Omega} w_i^{x_i}}{\sum_{y \in D^n} \left( \prod_{i \in \Omega} w_i^{y_i} \right)} \\
 &= \frac{\prod_{i \in \Omega} \left( \frac{p_i}{1-p_i} \right)^{x_i}}{\sum_{y \in D^n} \left( \prod_{i \in \Omega} \left( \frac{p_i}{1-p_i} \right)^{y_i} \right)} \\
 &\stackrel{(5.8)}{=} p_X^{(2)}(x).
 \end{aligned}$$

Bei gleicher Wahl von  $w_1, \dots, w_N$  ist für  $x \in D^n$

$$p_X^{(1)}(x) \stackrel{(5.6)}{=} \frac{\prod_{i \in \Omega} w_i^{x_i}}{\sum_{y \in D^n} \left( \prod_{i \in \Omega} w_i^{y_i} \right)} \stackrel{(5.21)}{=} p_X^{(3)}(x) \quad \blacksquare$$

Aus diesem Grund kann man alle drei Methoden als das maximale-Entropie-Modell für bedingte Bernoulli-Verteilungen mit  $w_i = \frac{p_i}{(1-p_i)}$  für  $i \in \Omega$  bezeichnen.

## 5.2 Legitimation des bedingten Bernoulli-Modells bei der Erzeugung von Binärmatrizen mit festen Zeilen- und Spaltensummen

Im folgenden Satz wird gezeigt, warum beim sequentiellen Erzeugen von Binärmatrizen mit festen Zeilen- und Spaltensummen mittels Importance Sampling die bedingte Bernoulli-Verteilung eine geeignete Probeverteilung darstellt. Dazu sei zunächst angemerkt, dass eine  $m \times n$  Matrix  $A \in \Sigma_{rc}$  als eine Realisierung eines diskreten Zufallsvektors  $X = (X_1, \dots, X_n)$  aufgefasst werden kann, wobei die  $i$ -te Spalte von  $A$  eine Realisierung von  $X_i$  ist. Dementsprechend ist die Verteilung der  $i$ -ten Spalte von  $A$  gerade die Randverteilung von  $X_i$ .

**Satz 5.7** (nach [7] S. 112)

Sei  $\Sigma_{rc_1}$  die Menge aller  $m \times n$  Binärmatrizen mit dem ZSV  $r = (r_1, \dots, r_m)$  und der ersten Spaltensumme  $c_1$  und  $p$  die Dichte der Gleichverteilung über  $\Sigma_{rc_1}$ . Dann ist die Randverteilung der ersten Spalte gerade die bedingte Verteilung  $g$  von  $m$  unabhängigen Bernoulli-verteilten Zufallsvariablen  $Z_1, \dots, Z_m$  mit den Erfolgswahrscheinlichkeiten  $p_i = \frac{r_i}{n}$ , ( $1 \leq i \leq m$ ) unter der Voraussetzung, dass  $\sum_{i=1}^m Z_i = c_1$  ist.

BEWEIS (nach [7] S. 119)

Ein Algorithmus, der gleichverteilte  $m \times n$  Binärmatrizen mit den Zeilensummen  $r_1, \dots, r_m$  und der ersten Spaltensumme  $c_1$  generiert, könnte zum Beispiel so aussehen:

**Algorithmus**

Man starte mit einer leeren  $m \times n$  Binärmatrix  $M$ .

1. Suche für  $1 \leq i \leq m$  zufällig  $r_i$  Positionen der  $i$ -ten Zeile von  $M$  aus und belege diese mit einer Eins. Jede Zeile wird dabei unabhängig von den anderen Zeilen belegt.
2. Man wähle nun diejenigen in 1. generierten Matrizen aus, deren erste Spaltensumme gleich  $c_1$  ist.

Dadurch, dass sämtliche Einträge zufällig und unabhängig getätigt wurden, ist die Wahrscheinlichkeit für jede einzelne erzeugte Matrix gleich. Man erhält so gleichverteilte  $m \times n$  Binärmatrizen mit den geforderten Zeilensummen und der geforderten ersten Spaltensumme.

Die Randverteilung der ersten Spalte einer solchen Matrix ergibt sich dann aus Schritt 1. des obigen Algorithmus. Die Wahrscheinlichkeit, dass der erste Eintrag der  $i$ -ten Zeile gleich Eins ist, ist  $\frac{r_i}{n}$ , also die Anzahl der zu verteilenden Einsen durch die Anzahl an möglichen Positionen. Man kann somit nun die erste Spalte als  $m$  unabhängige Bernoulli-Experimente  $Z_1, \dots, Z_m$  mit Erfolgswahrscheinlichkeiten  $p_i = \frac{r_i}{n}$ , ( $1 \leq i \leq m$ ) auffassen. In Schritt 2. des Algorithmus werden diejenigen Matrizen ausgewählt, deren erste Spaltensumme gerade  $c_1$  ist. Die Randverteilung ist folglich die bedingte Verteilung der  $Z_1, \dots, Z_m$  unter der Voraussetzung, dass  $\sum_{i=1}^m Z_i = c_1$  ist. ■

**Bemerkung 5.8**

Sei die  $m \times n$  Matrix  $A \in \Sigma_{rc}$  Realisierung eines diskreten,  $n$ -dimensionalen Zufallsvektors  $X = (X_1, \dots, X_n)$  und  $a_1$  die erste Spalte von  $A$ . Sei  $g$  die in Satz 5.7 erläuterte Dichte der bedingten Bernoulli-Verteilung von  $X$  und  $P$  die Gleichverteilung auf  $\Sigma_{rc}$ . Es ist dann die Wahrscheinlichkeit der ersten Spalte durch  $g(a_1) = P(X_1 = a_1 \mid r_1, \dots, r_m, c_1)$  gegeben. Die tatsächlich angestrebte Randverteilung der ersten Spalte ist jedoch durch  $P(X_1 = a_1 \mid r_1, \dots, r_m, c_1, \dots, c_n)$  gegeben. □

Die bedingte Bernoulli-Verteilung kann also die Gleichverteilung auf  $\Sigma_{rc}$  nicht exakt wiedergeben. Es wird sich jedoch in Kapitel 7 zeigen, dass für viele auf realen ökologischen Daten basierenden Binärmatrizen durch die bedingte Bernoulli-Verteilung

bereits eine gute Probeverteilung für das Importance-Sampling aufgestellt wird, deren Chi-Quadrat-Abstand (vergleiche Definition 4.6) relativ gering ausfällt. Man wird trotzdem geneigt sein, die Probeverteilung noch weiter an die Zielverteilung anzupassen. Möglichkeiten dazu werden in Abschnitt 6.2 vorgestellt.

**Bemerkung 5.9**

Nach Satz 5.7 ist die Randverteilung der ersten Spalte einer gemäß der Gleichverteilung über  $\Sigma_{r_{c_1}}$  aus Satz 5.7 verteilten  $m \times n$  Matrix  $A$  die bedingte Bernoulli-Verteilung mit den Erfolgswahrscheinlichkeiten  $p_i = \frac{r_i}{n}$ , wobei  $i \in \{1, \dots, m\}$  und  $r_i$  die  $i$ -te Zeilensumme von  $A$  ist. Zur Bestimmung der Dichte dieses bedingten Bernoulli-Modells müssen nur noch die entsprechenden Gewichte für (5.21) bestimmt werden. Es gilt dabei

$$w_i \stackrel{(5.26)}{=} \frac{p_i}{(1 - p_i)} = \frac{\frac{r_i}{n}}{(1 - \frac{r_i}{n})} = \frac{r_i}{n} \frac{1}{(1 - \frac{r_i}{n})} = \frac{r_i}{(n - r_i)}. \quad (5.27)$$

□

### 5.3 Stichprobensystem zur bedingten Bernoulli-Verteilung

Mit Hilfe der zuvor vorgestellten Methoden ist es nun möglich ein Schema anzugeben, um eine ungeordnete Stichprobe von  $n$  Einheiten aus der Menge  $\Omega = \{1, \dots, N\}$  zu wählen.

**Satz 5.10** (nach [5] S. 461)

Sei  $\Omega = \{1, \dots, N\}$  und bezeichne  $A_k$  für  $k \in \Omega \cup \{0\}$  die Menge der nach  $k$  Schritten in eine Stichprobe aufgenommenen Einheiten, wobei  $A_0 = \emptyset$  und  $A_n$  die komplette Stichprobe von  $n$  Einheiten aus  $\Omega$  sei. Es werde dann in Schritt  $k$  eine Einheit  $j \in A_{k-1}^c$  mit der Wahrscheinlichkeit

$$p_{bed}(j, A_{k-1}^c) := \frac{w_j R(n - k, A_{k-1}^c \setminus \{j\})}{(n - k + 1)R(n - k + 1, A_{k-1}^c)} \quad (5.28)$$

in die Stichprobe aufgenommen. Man vergleiche  $p_{bed}$  mit (5.24). Es gilt dann für eine Stichprobe  $s = (s_1, \dots, s_n) \in \Omega^n$  beziehungsweise deren Indikatorstichprobe  $s^I \in D^n$ , die Realisierung des diskreten,  $N$ -dimensionalen Zufallsvektors  $X = (X_1, \dots, X_N)$  ist,

$$\prod_{k=1}^n p_{bed}(s_k, A_{k-1}^c) = \frac{1}{n!} p_X^{(3)}(s^I), \quad (5.29)$$

wobei  $p_X^{(3)}$  die Dichte aus (5.21) bezeichne.  $\Omega^n$  und  $D^n$  seien dabei wie gewohnt definiert.

BEWEIS (nach [5] S. 462)

Sei  $P$  die Verteilung von  $X$ . Für die Wahrscheinlichkeit die Stichprobe  $s$ , mittels der Wahrscheinlichkeiten aus (5.28) in der Reihenfolge  $s_1, s_2, \dots, s_n$  zu ziehen, gilt mit Satz 2.4

$$\begin{aligned}
 P(X_1 = s_1, \dots, X_n = s_n) &= P(X_1 = s_1)P(X_2 = s_2 \mid X_1 = s_1) \\
 &\quad \dots P(X_n = s_n, \mid X_1 = s_1, \dots, X_{n-1} = s_{n-1}) \\
 &\stackrel{(5.28)}{=} \prod_{k=1}^n p_{bed}(s_k, A_{k-1}^c) \\
 &= \prod_{k=1}^n \frac{w_{s_k} R(n-k, A_k^c)}{(n-k+1)R(n-k+1, A_{k-1}^c)} \\
 &= \frac{w_{s_1}}{n} \frac{R(n-1, A_1^c)}{R(n, A_0^c)} \cdot \frac{w_{s_2}}{n-1} \frac{R(n-2, A_2^c)}{R(n-1, A_1^c)} \cdot \dots \\
 &\quad \cdot \frac{w_{s_{n-1}}}{2} \frac{R(1, A_{n-1}^c)}{R(2, A_{n-2}^c)} \cdot \frac{w_{s_n}}{1} \frac{R(0, A_n^c)}{R(1, A_{n-1}^c)}.
 \end{aligned}$$

Durch Kürzen und Umformen ergibt sich

$$\begin{aligned}
 P(X_1 = s_1, \dots, X_n = s_n) &= \frac{1}{n!} \left( \prod_{k=1}^n w_{s_k} \right) \frac{R(0, A_n^c)}{R(n, A_0^c)} \\
 &= \frac{1}{n!} \left( \prod_{k=1}^n w_{s_k} \right) \frac{R(0, A_n^c)}{R(n, \Omega)} \\
 &= \frac{1}{n!} \frac{\prod_{k=1}^n w_{s_k}}{R(n, \Omega)} \\
 &\stackrel{(5.23)}{=} \frac{1}{n!} \frac{\prod_{k=1}^n w_{s_k}}{\sum_{x \in D^n} \prod_{i \in \Omega} w_i^{x_i}} \\
 &\stackrel{(5.21)}{=} \frac{1}{n!} p_X^{(3)}(s^I),
 \end{aligned}$$

wobei  $s^I \in D^n$  mit  $s_i^I = 1$  für  $i \in A_n$  ist. ■

Der Faktor  $n!$  entsteht dadurch, dass durch die in Satz 5.10 beschriebene Ziehung die Stichprobe in einer festen Reihenfolge vorliegt. Da es  $n!$  Möglichkeiten gibt, die Elemente der Stichprobe anzuordnen, ergeben sich die entsprechenden Wahrscheinlichkeiten.

# 6 Ein sequentielles Importance Sampling Verfahren zur Erzeugung von Binärmatrizen mit festen Zeilen- und Spaltensummen

In diesem Kapitel wird nun, basierend auf den in den vorherigen Kapiteln aufgeführten Grundlagen, ein sequentieller Importance Sampling (kurz SIS) Algorithmus zur Generierung von unabhängigen, annähernd gleichverteilten Binärmatrizen mit festen Zeilen- und Spaltensummen vorgestellt.

Dieser wird zunächst in Form eines Schemas dargestellt. Danach werden noch einige Anpassungsmöglichkeiten des Algorithmus an unterschiedliche Daten angegeben. Im Anhang ist dieser Algorithmus, in die Programmiersprache  $C$  implementiert, angegeben. Mit Hilfe der mittels des SIS Algorithmus generierten Matrizen wird es möglich sein, statistische Tests anzuwenden. Damit werden dann in Kapitel 7 Aussagen über die Zufälligkeit der Verteilung der in der zugrunde liegenden Datenmatrix wiedergegebenen Arten auf die ebenfalls in dieser Matrix wiedergegebenen Lebensräume getroffen.

## 6.1 Der Algorithmus

Angenommen es ist  $D \in \Sigma_{rc}$  eine  $m \times n$  Datenmatrix beruhend auf ökologischen Daten, wobei  $r = (r_1, \dots, r_m)$  und  $c = (c_1, \dots, c_n)$  gilt. Dann können zur Optimierung der Rechenzeit nicht in die Berechnung eingehende Zeilen und Spalten aus  $D$  entfernt werden. So können alle *vollen* Zeilen und Spalten, also Zeilen und Spalten, in denen jeder Eintrag gleich Eins ist, und alle *leeren* Zeilen und Spalten, also diejenigen, in denen jeder Eintrag gleich Null ist, aus  $D$  entfernt werden. Diese Zeilen und Spalten sind für die späteren Berechnungen irrelevant. Es müssen jedoch entsprechend entfernte Zeilen und Spalten später an die generierte Matrix wieder angefügt werden, da eventuell auf die Matrix angewandte Statistiken sonst verfälscht werden können.

Die Zeilen von  $D$  werden der Zeilensumme nach beginnend mit der Größten sortiert. Dies ist für die Anwendung von Satz 3.16 notwendig. Die Anzahl an Matrizen mit den gleichen Zeilen- und Spaltensummen wie  $D$  hängt nicht von der Reihenfolge der Zeilen- und Spaltensummen ab, somit ist dies ohne Weiteres möglich.

Es könnte allerdings sein, dass es für eine Statistik, die auf die so erzeugten Matrizen angewandt werden soll, von Bedeutung ist, in welcher Reihenfolge die Zeilen und Spalten der Matrizen vorliegen. Eine solche Statistik macht allerdings bei der Betrachtung von Matrizen, die das Zusammenleben verschiedener Arten in verschiedenen Lebensräumen wiedergeben, keinen Sinn, da es für das Zusammenleben der Arten irrelevant ist, in welcher Reihenfolge sie in einer Datenmatrix erfasst werden. In Abschnitt 7.2.1 wird noch auf diese Problematik eingegangen.

Es sei im Folgenden stets  $D \in \Sigma_{rc}$  die betrachtete  $m \times n$  Datenmatrix.  $D$  soll insbesondere keine *vollen* und *leeren* Zeilen und Spalten enthalten.

Es sei  $X = (X_1, \dots, X_n)$  ein diskreter Zufallsvektor mit Werten in  $\Sigma_{rc}$ , wobei  $X_i$  für  $i \in \{1, \dots, n\}$  die Ziehung der Einsen in der  $i$ -ten Spalte  $t_{\cdot,i}$  einer Matrix  $T \in \Sigma_{rc}$  repräsentiere.  $g$  sei die Verteilung von  $X$ . Nach Bemerkung 4.12 ist es dann möglich, mit geeigneten Verteilungen  $g_1, \dots, g_n$  von  $X_1, \dots, X_n$ , für die

$$\begin{aligned} g_1(X_1 = t_{\cdot,1}) &= g(X_1 = t_{\cdot,1}), \\ g_2(X_2 = t_{\cdot,2}) &= g(X_2 = t_{\cdot,2} \mid X_1 = t_{\cdot,1}), \\ &\vdots \\ g_n(X_n = t_{\cdot,n}) &= g(X_n = t_{\cdot,n} \mid X_1 = t_{\cdot,1}, \dots, X_{n-1} = t_{\cdot,(n-1)}) \end{aligned}$$

gilt,  $g(X = T)$  durch

$$g(X = T) = \prod_{i=1}^n g_i(X_i = t_{\cdot,i}) \tag{6.1}$$

zu berechnen. Diese Verteilungen  $g_1, \dots, g_n$  werden mittels des maximalen Entropie-Modells für bedingte Bernoulli-Verteilungen ermittelt.

Der SIS Algorithmus vollzieht sich in folgenden Schritten (man vergleiche dazu [7] S. 114).

1. Man erzeuge eine  $m \times n$  Matrix  $T = (t_{i,j})_{i=1, \dots, m; j=1, \dots, n}$  mit  $t_{i,j} = 0$  für alle  $i \in \{1, \dots, m\}$  und  $j \in \{1, \dots, n\}$ . Diese wird im Verlauf des Algorithmus mit  $\sum_{i=1}^n c_i$  Einsen gefüllt.
2. Man betrachte zunächst nur die erste Spalte von  $T$ . Punkt 3 bis Punkt 8 wird dann für jede weitere Spalte von  $T$  wiederholt, bis schließlich nach  $n$  Durchläufen  $T \in \Sigma_{rc}$  gilt.

3. Man bestimme wie in Abschnitt 3.2.3 den  $t$ -Tupel  $K = (k_1, \dots, k_t)$  der Knoten und den  $t$ -Tupel  $V = (v_1, \dots, v_t)$  der Anzahlen an mindestens bis zu den jeweiligen Knoten einzusetzenden Einsen.
4. Es müssen dann in die ersten  $k_1$  Zeilen mindestens  $v_1$  Einsen eingetragen werden. Es können aber nur maximal  $\min(k_1, c_1)$  Einsen bis zum Knoten  $k_1$  eingetragen werden. Es bezeichne  $a_1$  die Anzahl der in die ersten  $k_1$  Zeilen zu setzenden Einsen. Die Anzahl  $a_1$  muss nun mit gleicher Wahrscheinlichkeit einen Wert  $i$  aus dem Intervall  $[v_1, \min(k_1, c_1)]$  annehmen. Man ermittle  $a_1$  also aus folgender Gleichverteilung

$$P(a_1 = i) = \frac{1}{\min(k_1, c_1) - v_1 + 1} \quad \text{für } v_1 \leq i \leq \min(k_1, c_1).$$

5. Man ermittle nun mittels der bedingten Bernoulli-Verteilung mit den Gewichten  $w_i = \frac{r_i}{n-r_i}$  (vergleiche (5.27)) eine Stichprobe  $s = (s_1, \dots, s_{a_1})$  mit  $a_1$  Elementen der Menge  $\{1, \dots, k_1\}$  und trage in die Positionen  $s_1, \dots, s_{a_1}$  der ersten Spalte von  $T$  Einsen ein. Für die Wahrscheinlichkeit, gerade diese Stichprobe zu ziehen, gilt dann

$$q_1(s^I) \stackrel{(5.29)}{=} a_1! \prod_{i=1}^{a_1} \frac{w_{s_i} R(a_1 - i, A_{i-1}^c \setminus s_i)}{(a_1 - i + 1) R(a_1 - i + 1, A_{i-1}^c)},$$

wobei

$$s^I \in \left\{ x = (x_1, \dots, x_{k_1}) \mid x_i \in \{0, 1\} \text{ und } \sum_{i=1}^{k_1} x_i = a_1 \right\}$$

die Indikatorstichprobe zu  $s$  ist. Außerdem sei  $A_i = \{s_1, \dots, s_i\}$  beziehungsweise  $A_i^c = \{1, \dots, k_1\} \setminus A_i$  für  $i \in \{1, \dots, a_1\}$ .

6. Für  $l \in \{2, \dots, t\}$  betrachte man nun den Knoten  $k_l$ . Es müssen nun mindestens  $\max(0, v_l - a_{l-1})$  und höchstens  $\min(c_1 - \sum_{h=1}^{l-1} a_h, k_l - k_{l-1})$  Einsen von der  $(k_{l-1} + 1)$ -ten bis einschließlich zur  $k_l$ -ten Zeile verteilt werden. Jeder Wert  $j \in [\max(0, v_l - a_{l-1}), \min(c_1 - \sum_{h=1}^{l-1} a_h, k_l - k_{l-1})]$  soll dabei gleich wahrscheinlich sein, sodass die Anzahl  $a_l$  der zu verteilenden Einsen gemäß der Verteilung

$$P(a_l = j) = \frac{1}{\min(c_1 - \sum_{h=1}^{l-1} a_h, k_l - k_{l-1}) - \max(0, v_l - a_{l-1}) + 1}$$

ermittelt werden kann. Man wähle dann wiederum mit Hilfe der bedingten Bernoulli-Verteilung eine Stichprobe  $s = (s_1, \dots, s_{a_l})$  mit  $a_l$  Elementen aus der

Menge  $\{k_{l-1}+1, \dots, k_l\}$ , sodass die Wahrscheinlichkeit, genau diese Stichprobe zu erlangen, gerade

$$q_l(s^I) \stackrel{(5.29)}{=} a_l! \prod_{i=1}^{a_l} \frac{w_{s_i} R(a_l - i, A_{i-1}^c \setminus s_i)}{(a_l - i + 1) R(a_l - i + 1, A_{i-1}^c)}$$

ist. Dabei sei

$$s^I \in \left\{ x = (x_{k_{l-1}+1}, \dots, x_{k_l}) \mid x_i \in \{0, 1\} \text{ und } \sum_{i=k_{l-1}+1}^{k_l} x_i = a_l \right\}$$

die Indikatorstichprobe zu  $s$  und außerdem  $A_i = \{s_1, \dots, s_i\}$  beziehungsweise  $A_i^c = \{k_{l-1}+1, \dots, k_l\} \setminus A_i$  für  $i \in \{1, \dots, a_l\}$ . In die Positionen  $s_1, \dots, s_{a_l}$  der ersten Spalte von  $T$  werden nun Einsen eingetragen.

7. Wurde Punkt 6 für alle Knoten der ersten Spalte durchgeführt, so ergibt sich die Wahrscheinlichkeit für die erste Spalte  $t_{\cdot,1}$  von  $T$  gerade die ermittelte Gestalt zu haben mit Bemerkung 4.12

$$g_1(t_{\cdot,1}) = \prod_{i=1}^t q_i((t_{k_{i-1}+1,1}, \dots, t_{k_i,1})), \quad (6.2)$$

wobei  $k_0 = 0$  sei.

8. Nun werden die Zeilensummen abhängig von den in der ersten Spalte verteilten Einsen aktualisiert und die Zeilen erneut der Zeilensummengröße nach beginnend mit der Größten sortiert. Damit wird die Anwendbarkeit von Satz 3.16 gewährleistet. Bei der Generierung der nächsten Spalte müssen die Gewichte für die bedingten Bernoulli-Verteilungen natürlich entsprechend angepasst werden. Angenommen es wurden bereits  $h-1$  Spalten generiert und es soll im nächsten Durchgang von Punkt 3 bis 7 die  $h$ -te Spalte generiert werden. Dann ergeben sich als Gewichte für die bedingte Bernoulli-Verteilung  $w_i = \frac{r_i^{(h)}}{n - (h-1) - r_i^{(h)}}$ , wobei  $r_i^{(h)}$  die nach dem  $(h-1)$ -ten Schritt aktualisierte  $i$ -te Zeilensumme ist.

Wurden alle  $n$  Spalten erfolgreich generiert, so lässt sich die Wahrscheinlichkeit  $g(X = T)$  durch (6.1) ermitteln, wobei sich die einzelnen Faktoren dieses Produkts aus (6.2) ergeben. Es ist sichergestellt, dass  $T_g = T_\pi$  gilt (vergleiche (4.1)).

Die bei diesem Verfahren entstandene Binärmatrix  $T$  ist dann aus  $\Sigma_{rc}$ . Wird dieses Verfahren mehrfach angewendet, erhält man viele unabhängig gemäß  $g$  verteilte Realisierungen. Diese kann man dann mit der Datenmatrix  $D$  vergleichen.

## 6.2 Verbesserung der Probeverteilung

Die bedingte Bernoulli-Verteilung gibt nach Bemerkung 5.8 die tatsächliche Verteilung der jeweiligen Spalten meist nicht exakt wieder. Somit ist es unter Umständen erforderlich, die Probeverteilung  $g$ , mittels der die Spalten generiert werden, anzupassen, sodass  $g$  der Gleichverteilung auf  $\Sigma_{rc}$  ähnlicher wird.

### 6.2.1 Anpassung der Gewichte

Eine Möglichkeit, die Probeverteilung  $g$  zu verbessern, wird in [7] vorgestellt. Es werden dabei die Gewichte  $w_i$  für die bedingte Bernoulli-Verteilung durch  $w_i^u$  mit einem geeigneten, konstanten Gewichtsexponenten  $u \in \mathbb{R}$  ein wenig abgeändert. Die Wahl von  $u$  hängt dabei allerdings von den jeweils vorliegenden Daten ab, sodass es kein ideales  $u$  für alle Situationen gibt.

Es muss also erst der beste Gewichtsexponent  $u$  für eine konkrete Datenmatrix  $D$  vor einer Datenauswertung mittels des SIS-Verfahrens durch einen separaten Durchgang ermittelt werden. Wie sich in Kapitel 7 zeigen wird, liegt der ideale Wert für  $u$  zu den in dieser Arbeit behandelten Daten in einem relativ kleinen Bereich um 1.

### 6.2.2 Anpassung der Datenmatrix

Eine andere Möglichkeit, die Effizienz des SIS-Verfahrens zu verbessern, besteht darin, die beobachteten Daten optimal auf das SIS-Verfahren vorzubereiten. Unterschiedliche Anordnungen der Spalten können einen Effekt auf die resultierende Probeverteilung  $g$  haben. Sind die Spalten der Spaltensummengröße nach, beginnend mit der Größten, sortiert, so werden zunächst Spalten mit vielen Einsen generiert und die Wahl der möglichen Positionen, in welche diese Einsen eingetragen werden können, ist relativ gering. Dadurch wird erreicht, dass die Verteilungen der ersten Spalten  $g(X_l = t_{.,l} \mid X_1 = t_{.,1}, \dots, X_{l-1} = t_{.,l-1})$  der Matrix  $T$  nur wenig von der erwünschten Verteilung abweichen können. Auch wird dadurch erreicht, dass sich die aktualisierten Zeilensummen schnell verkleinern und sich somit die Effizienz des SIS-Verfahrens weiter verbessert. Auf die Auswertung der generierten Matrizen mit den in dieser Arbeit verwendeten Teststatistiken hat die Umsortierung von Spalten keinen Einfluss, wie in Abschnitt 7.2.1 gezeigt wird.

Die Zeilen müssen für die Anwendbarkeit des Satzes 3.16 ohnehin der Zeilensummengröße nach sortiert werden, sodass eine Umsortierung dieser nicht zur Effizienzsteigerung des SIS-Verfahrens beitragen kann.

So bleibt nun nur noch die Möglichkeit, die Ausrichtung des SIS-Verfahrens zu verändern also zeilenweise statt spaltenweise Matrizen zu generieren. Alle weiteren Ma-

trixoperationen verändern auch die resultierenden Ergebnisse.

Es wird sich in Kapitel 7 zeigen, dass die Effizienz des SIS-Verfahrens bei den in dieser Arbeit behandelten Matrizen nicht zwangsweise größer ist, wenn bei größerer Zeilen- statt Spaltenanzahl der Datenmatrix  $D$  zeilenweise generiert wird. Zwar wird dann deutlich weniger Rechenzeit beansprucht, allerdings vergrößert sich unter Umständen der Chi-Quadrat-Abstand. Somit muss auch zur Ermittlung der optimalen Ausrichtung des SIS-Verfahrens zunächst ein Durchgang des Verfahrens durchgeführt werden, um dann diejenige Ausrichtung zu wählen, bei der der Chi-Quadrat-Abstand am geringsten ist.

# 7 Anwendung des SIS-Verfahrens auf ökologische Daten

In Abschnitt 2.4 wurden Tests zum Niveau  $\alpha$  vorgestellt. Mit Hilfe solcher Tests ist man nun auch in der Lage, mit Hilfe geeigneter Teststatistiken einen Konkurrenzzusammenhang zwischen bestimmten Arten in bestimmten Lebensräumen statistisch aufzudecken. Wie am Anfang von Kapitel 3 erwähnt wurde, wird die Verteilung der Arten auf die Lebensräume mittels Binärmatrizen wiedergegeben.

Für die Durchführung eines Tests zum Niveau  $\alpha$  benötigt man nun die exakte Gleichverteilung aller Binärmatrizen mit gleichen Zeilen- und Spaltensummen, wie die auf beobachteten Daten beruhende Binärmatrix. Da diese Verteilung allerdings nur in den seltensten Fällen explizit bestimmt werden kann, ist eine direkte Durchführung eines Tests zum Niveau  $\alpha$  nur selten möglich. Allerdings wurde in Kapitel 4 ein Verfahren vorgestellt, mit dessen Hilfe man bei unbekannter Verteilung  $\pi$  mehrere unabhängige, identisch verteilte Realisierungen einer diskreten,  $\mathcal{X}$ -wertigen Zufallsvariablen  $X$  mit einer Verteilung  $g$  generieren kann, die möglichst nahe an  $\pi$  liegt. Mit Hilfe dieser Stichprobe ist es dann möglich, den p-Wert eines Tests zum Niveau  $\alpha$  zu schätzen.

In Beispiel 4.4 wurde für den Fall der Binärmatrizen zu einer beliebigen Abbildung  $h : \mathcal{X} \rightarrow \mathbb{R}$  eine Schätzung für den Erwartungswert dieser Abbildung über  $\pi$  angegeben. Die Abbildung  $h$  kann dabei natürlich auch eine Entscheidungsregel  $\varphi$  sein.

## Beispiel 7.1

Sei  $y \in \Sigma_{rc}$ , wobei die Zeilen von  $y$  verschiedene Arten und die Spalten von  $y$  verschiedene Lebensräume repräsentieren. Die Hypothese „interspezifische Konkurrenz spielte keine Rolle bei der Besiedlung der Lebensräume“ gegen die Alternative „interspezifische Konkurrenz hat die Arten bei ihrer Lebensraumwahl beeinflusst“ würde bedeuten, dass unter der Nullhypothese jegliche Konfiguration von  $y$  die gleiche Wahrscheinlichkeit besitzt und somit die Verteilung unter der Nullhypothese die diskrete Gleichverteilung auf  $\Sigma_{rc}$  ist. Es ist also  $\Theta_0 = \{|\Sigma_{rc}|\}$  beziehungsweise  $\pi = \frac{1}{|\Sigma_{rc}|}$  die Verteilung unter der Nullhypothese.

Mit Hilfe des in Kapitel 6 vorgestellten Algorithmus ist es nun möglich, unabhängige

Realisierungen  $x_1, \dots, x_n \in \Sigma_{rc}$  einer diskreten Zufallsvariablen  $X$ , die mittels bedingter Bernoulli-Modelle beschrieben wird, zu ermitteln. Mit Hilfe einer geeigneten Teststatistik  $T$  kann die Hypothese zum Beispiel zu  $H_0 : T(X) \leq T(y)$  umformuliert werden. Es bezeichne  $t := T(y)$ . Ein Test zum Niveau  $\alpha$  wird dann für alle  $x \in \Sigma_{rc}$  durch die Entscheidungsregel

$$\varphi(x) = 1_{\{T(x) > t\}}(x) \quad (7.1)$$

ausgedrückt, wobei  $E_\pi(\varphi) \leq \alpha$  gelten soll. Nach Beispiel 4.4 ist

$$\widehat{E_\pi(\varphi)} = \frac{\sum_{i=1}^n \frac{1}{g(X=x_i)} \varphi(x_i)}{\sum_{i=1}^n \frac{1}{g(X=x_i)}} \quad (7.2)$$

eine Schätzung für  $E_\pi(\varphi)$ . Es ist

$$E_\pi(\varphi) = \pi(\varphi(X) = 1) = \pi(T(X) > t).$$

Da  $\Theta_0 = \{|\Sigma_{rc}|\}$  ist, ist somit durch (7.2) in diesem Fall eine Schätzung für den p-Wert. Gilt also  $\widehat{E_\pi(\varphi)} \leq \alpha$ , so wird  $H_0$  abgelehnt und es gilt dann als wahrscheinlich, dass interspezifische Konkurrenz einen Einfluss auf die Verteilung der beobachteten Arten hat. □

## 7.1 Diamond's Versammlungs-Regeln

Um aus einer Binärmatrix aus ökologischen Daten Aussagen darüber zu treffen, ob zwischen den verschiedenen Arten ein Konkurrenzzusammenhang besteht, müssen zunächst Regeln angeführt werden, wodurch sich ein Konkurrenzzusammenhang auszeichnet. 1975 formulierte der Biologe Jared Diamond einige Regeln nach denen sich Gemeinschaften im Tierreich zusammenfinden und diese auch beibehalten. Diese richten sich nach dem Grundsatz der interspezifischen Konkurrenz, also der Konkurrenz verschiedener Arten um Ressourcen wie Nahrung oder Lebensraum.

**Diamond's Versammlungs-Regeln** (nach [10] S. 344)

1. Berücksichtigt man alle möglichen Kombinationen von ähnlichen Arten, so sind nur einige dieser Kombinationen in der Natur zu finden.
2. Zulässige Kombinationen widerstehen Eindringlingen, die dazu führen würden, dass eine unzulässige Kombination entstehen würde.

3. Eine Kombination, die in einem großen oder artenreichen Lebensraum stabil ist, kann in einem kleinen oder artenarmen Lebensraum instabil sein.
4. In einem kleinen oder artenarmen Lebensraum kann eine Kombination einem Eindringling widerstehen, der in einem großen oder artenreichen Lebensraum aufgenommen werden würde.
5. Einige Arten-Paare koexistieren nie im selben Lebensraum, weder alleine noch in einer größeren Kombination.
6. Einige Arten-Paare, die alleine eine instabile Kombination bilden, können Teil einer größeren stabilen Kombination sein.
7. Einige Kombinationen, die nur aus stabilen Unterkombinationen bestehen, sind instabil.

Mit Hilfe dieser Regeln lassen sich Statistiken ableiten, die Rückschlüsse auf die Konkurrenz verschiedener beobachteter Arten in unterschiedlichen Lebensräumen zulassen.

## 7.2 Teststatistiken zum Testen auf Konkurrenz

In diesem Abschnitt werden vier unterschiedliche Teststatistiken mit entsprechenden Beispielen angegeben, die üblicherweise zum Testen auf einen Konkurrenzzusammenhang zwischen den durch eine Binärmatrix wiedergegebenen Arten verwendet werden. Die unterschiedlichen Teststatistiken beruhen auf verschiedenen Aspekten der interspezifischen Konkurrenz. Daher sollte, wenn auf einen bestimmten Aspekt der interspezifischen Konkurrenz getestet werden soll, die entsprechende Teststatistik gewählt werden.

Es bezeichne wie gehabt  $\{0, 1\}^{m \times n}$  die Menge aller  $m \times n$  Matrizen mit Einträgen aus  $\{0, 1\}$  und  $a_{i,j}$  für  $i \in \{1, \dots, m\}$  und  $j \in \{1, \dots, n\}$  den  $i, j$ -Eintrag einer Matrix  $A \in \{0, 1\}^{m \times n}$ .

**Definition 7.2 (COMBO)** (nach [25] S. 206)

Sei  $A \in \{0, 1\}^{m \times n}$ . Es bezeichne  $a_{\cdot,i}$  für  $i \in \{1, \dots, n\}$  die  $i$ -te Spalte von  $A$ . Des Weiteren sei  $a_{\cdot,i} \doteq a_{\cdot,j}$  für  $i, j \in \{1, \dots, n\}$  genau dann, wenn  $a_{k,i} = a_{k,j}$  für alle  $k \in \{1, \dots, m\}$  ist. Dabei ist durch  $\doteq$  trivialerweise eine Äquivalenzrelation gegeben und

$$[i] := \{a_{\cdot,j} \mid a_{\cdot,j} \doteq a_{\cdot,i}, j \in \{1, \dots, n\}\} \quad (7.3)$$

ist dann die  $i$ -te Äquivalenzklasse.

Es bezeichne

$$COMBO(A) := |\{[i] \mid i \in \{1, \dots, n\}\}| \quad (7.4)$$

die Anzahl aller verschiedenen Spalten-Konfigurationen von  $A$ .

Nach Diamond's erster und zweiter Regel ist die Anzahl an Arten-Kombinationen geringer, wenn ein Konkurrenzzusammenhang besteht. In einer Binärmatrix bedeuten unterschiedliche Arten-Kombinationen unterschiedliche Spalten-Konfigurationen. Man kann nun also mittels der Teststatistik  $COMBO$  die Anzahl von verschiedenen Spalten-Konfigurationen einer Binärmatrix, die aus beobachteten Daten besteht, mit der Anzahl von verschiedenen Spalten-Konfigurationen zufällig erzeugter Binärmatrizen vergleichen. Ist die Anzahl bei der Matrix, die aus beobachteten Daten besteht, niedriger als bei den zufällig erzeugten, ist ein Konkurrenzzusammenhang wahrscheinlich.

### Beispiel 7.3

Sei

$$A = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \in \{0, 1\}^{6 \times 11}.$$

Es ist leicht zu sehen, dass die vierte und die achte Spalte von  $A$  identisch sind. Genauso sind die fünfte und sechste Spalte von  $A$  sowie die siebte, neunte und zehnte Spalte von  $A$  identisch. Es gibt also sieben unterschiedliche Spaltenkonfigurationen. Demnach ist  $COMBO(A) = 7$ . □

### Definition 7.4 (CHECKER) (nach [15] S. 2607)

Sei  $A \in \{0, 1\}^{m \times n}$ , dann bezeichnet

$$CHECKER(A) := |\{(i, j) \mid i, j \in \{1, \dots, m\}, i < j, \text{ für die } a_{i,k} = 1 \Rightarrow a_{j,k} = 0 \text{ für alle } k \in \{1, \dots, n\} \text{ gilt}\}|$$

die Anzahl an Zeilenpaaren von  $A$ , in denen nie gleichzeitig in einer Spalte eine Eins steht.

Die Teststatistik  $CHECKER$  beruht auf Diamond's fünfter Regel. Die Anzahl an Zeilenpaaren in einer Binärmatrix, in der nie gleichzeitig in einer Spalte je eine Eins steht, ist größer in einer Binärmatrix, die auf Arten beruht, bei denen ein Konkurrenzzusammenhang besteht, als bei einer Binärmatrix, die auf Arten ohne einen

Konkurrenzzusammenhang beruht. Man kann wieder eine Binärmatrix aus beobachteten Daten mit zufällig generierten Binärmatrizen vergleichen. Es ist ein Konkurrenzzusammenhang wahrscheinlich, wenn die Anzahl an derartigen Zeilenpaaren in einer Matrix aus beobachteten Daten größer ist als bei den zufällig generierten Matrizen.

**Beispiel 7.5**

Sei

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} \in \{0, 1\}^{4 \times 5}$$

dieselbe Binärmatrix wie in Beispiel 3.19. Die dritte und vierte Zeile von  $A$  bilden das einzige Zeilenpaar in  $A$ , in denen nie in einer Spalte gleichzeitig eine Eins steht. Daher ist  $CHECKER(A) = 1$ . □

**Definition 7.6 (Checkerboard unit)**

Sei  $A \in \{0, 1\}^{m \times n}$ . Ein Paar  $((i, j), (k, l))$  zweier Indexpaare mit  $i, j \in \{1, \dots, m\}$ , wobei  $i \neq j$  ist, und  $k, l \in \{1, \dots, n\}$ , wobei  $k \neq l$  ist, bildet eine Checkerboard unit genau dann, wenn

$$a_{i,k} = a_{j,l} \neq a_{j,k} = a_{i,l} \tag{7.5}$$

gilt.

Eine Checkerboard unit liegt in einer Binärmatrix basierend auf ökologischen Daten demnach vor, wenn sich zwei Arten in zwei Lebensräumen gegenseitig ausschließen. In einer solchen Binärmatrix wird dies durch eine  $2 \times 2$  Untermatrix der Form

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \text{ oder } \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

ausgedrückt.

Eine weitere Teststatistik, die auf Diamonds fünfter Regel basiert und durch Checkerboard units beschrieben wird, ist der von Lewi Stone und Alan Roberts vorgeschlagene Checkerboard Score ( $C - score$ ).

**Definition 7.7 ( $C - score$ )** (nach [31] S. 75)

Sei zunächst  $A = (a_{i,j})_{i=1,\dots,m;j=1,\dots,n} \in \{0, 1\}^{m \times n}$  und sei  $s_{p,q} = |\{j \mid a_{p,j} = a_{q,j} = 1\}|$  für  $p, q \in \{1, \dots, m\}$  die Anzahl an Spalten, in denen sowohl in der  $p$ -ten, als auch in der  $q$ -ten Zeile eine Eins steht. Es gilt dann  $s_{p,q} = \sum_{j=1}^n a_{p,j} a_{q,j}$ .

Aufgrund dieser Schreibweise ist leicht zu sehen, dass

$$S := (s_{p,q})_{p,q=1,\dots,m} = AA^T \quad (7.6)$$

gilt. Sei des Weiteren

$$C_{p,q} := (r_p - s_{p,q})(r_q - s_{p,q}) \quad (7.7)$$

die Anzahl an Checkerboard units, die aus der  $p$ -ten und  $q$ -ten Zeile von  $A$  gebildet werden können, wobei  $r_p$  die  $p$ -te und  $r_q$  die  $q$ -te Zeilensumme von  $A$  ist.

Da es  $m$  Zeilen in  $A$  gibt, existieren  $\frac{m(m-1)}{2}$  verschiedene Zeilen-Paare in  $A$ . Es sei dann mit (7.6) und (7.7)

$$C - score(A) := \frac{2 \sum_{p=1}^{m-1} \sum_{q=p+1}^m C_{p,q}}{m(m-1)} \quad (7.8)$$

der Checkerboard Score ( $C - score$ ) von  $A$ .

Der  $C - score$  setzt die Anzahl an Checkerboard units einer Binärmatrix in Bezug zur Anzahl an Zeilen-Paaren. Genau wie bei *CHECKER* gilt auch hier, dass ein Konkurrenzzusammenhang wahrscheinlicher ist, je größer  $C - score$  ausfällt.

### Beispiel 7.8

Sei  $A$  dieselbe Binärmatrix wie in Beispiel 7.5. Es ist

$$S = AA^T = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 4 & 3 & 2 & 1 \\ 3 & 4 & 2 & 1 \\ 2 & 2 & 2 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix}. \quad (7.9)$$

Weiter gilt

$$\begin{aligned} C_{1,2} &= (4 - 3)(4 - 3) = 1, \\ C_{1,3} &= (4 - 2)(2 - 2) = 0, \\ C_{1,4} &= (4 - 1)(1 - 1) = 0, \\ C_{2,3} &= (4 - 2)(2 - 2) = 0, \\ C_{2,4} &= (3 - 1)(1 - 1) = 0, \\ C_{3,4} &= (2 - 0)(1 - 0) = 2. \end{aligned}$$

Es ist somit

$$C - score(A) = \frac{2 \sum_{p=1}^3 \sum_{q=p+1}^4 C_{p,q}}{4(4-1)} = \frac{2 \cdot 3}{12} = \frac{1}{2}. \quad \square$$

Eine weitere Teststatistik, die auf  $S$  aus (7.6) beruht, liefern Alan Roberts und Lewi Stone in [27]. Sie wird durch das arithmetische Mittel der quadrierten Nicht-Diagonaleinträge von  $S$  berechnet. Da  $s_{p,q}$  die Anzahl an Lebensräumen beschreibt, die sich die  $p$ -te und  $q$ -te Art teilen, sind die Diagonaleinträge für die weitere Betrachtung irrelevant.

**Definition 7.9 ( $\bar{S}^2$ )** (nach [27] S. 561)

Sei  $A \in \{0,1\}^{m \times n}$  und  $S = AA^T$  wie in (7.6). Es gibt genau  $m(m-1)$  Nicht-Diagonaleinträge in  $S$ . Daraus ergibt sich für das arithmetische Mittel der quadrierten Nicht-Diagonaleinträge von  $S$

$$\bar{S}^2 := \frac{1}{m(m-1)} \sum_{\substack{p,q=1,\dots,m \\ p \neq q}} s_{p,q}^2. \quad (7.10)$$

Auch diese Teststatistik basiert auf Diamonds fünfter Regel und es gilt wieder die Tatsache, dass ein Konkurrenzzusammenhang wahrscheinlicher ist, je größer die Statistik ausfällt.

**Beispiel 7.10**

Sei  $A$  wieder dieselbe Binärmatrix wie in Beispiel 7.5. Dann ist  $S$  wie in (7.9) und es ergibt sich

$$\begin{aligned} \bar{S}^2 &= \frac{\sum_{\substack{p,q=1,\dots,4 \\ p \neq q}} s_{p,q}^2}{4(4-1)} \\ &= \frac{9 + 4 + 1 + 9 + 4 + 1 + 4 + 4 + 0 + 1 + 1 + 0}{12} = \frac{19}{6}. \quad \square \end{aligned}$$

## 7.2.1 Eigenschaften der Statistiken

Stone und Roberts geben in [31] die so genannte Relabel-Äquivalenz als eine wichtige Eigenschaft einer Teststatistik zur Bestimmung eines Konkurrenzzusammenhanges aus einer Binärmatrix an.

**Definition 7.11 (Relabel-Äquivalenz)** (nach [31] S. 75)

Seien  $A, B \in \{0,1\}^{m \times n}$ , dann sind  $A$  und  $B$  genau dann relabel-äquivalent, wenn sie durch Vertauschen von Zeilen und/oder Spalten ineinander umwandelbar sind.

**Beispiel 7.12** (nach [31] S. 76)

Es seien

$$O := \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \quad P := \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \quad Q := \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} \quad R := \begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix}.$$

jeweils aus  $\{0, 1\}^{4 \times 4}$ . Die Matrizen  $O, Q$  und  $R$  sind paarweise relabel-äquivalent. So ist beispielsweise  $O$  durch Tauschen der zweiten und vierten Spalte in  $R$  umwandelbar oder  $R$  durch Tauschen der zweiten und dritten Zeile und der dritten und vierten Spalte in  $Q$  umwandelbar.  $P$  ist jedoch nicht zu  $O, Q$  oder  $R$  relabel-äquivalent.  $\square$

Es handelt sich bei der Relabel-Äquivalenz offensichtlich um eine Äquivalenz-Relation. Demnach kann man sämtliche gleichdimensionalen Binärmatrizen in Relabel-Äquivalenz-Klassen einteilen. Da es für einen Konkurrenzzusammenhang unerheblich ist, in welcher Reihenfolge Arten und Lebensräume in eine Binärmatrix eingetragen sind, sollte eine Teststatistik, die einen solchen Zusammenhang ausdrücken soll, invariant gegenüber Zeilen- oder Spaltenvertauschungen sein. Sie sollte also Matrizen der gleichen Relabel-Äquivalenz-Klasse den gleichen Wert zuweisen.

**Folgerung 7.13**

$COMBO$ ,  $CHECKER$ ,  $C$  – score und  $\bar{S}^2$  sind invariant bezüglich Zeilen- oder Spaltenvertauschungen.  $\square$

BEWEIS

Sei im Folgenden stets  $A \in \{0, 1\}^{m \times n}$ .

1.  $COMBO$

$COMBO(A)$  ist die Anzahl der verschiedenen Spalten-Konfigurationen von  $A$ . Diese ändert sich natürlich nicht durch Vertauschen von Spalten.

Durch Vertauschen von Zeilen von  $A$  wird zwar bewirkt, dass sich die Spaltenkonfigurationen ändern, da aber in jeder Spalte dieselben Einträge vertauscht werden, ändert sich auch durch Zeilenvertauschungen nichts an der Anzahl der verschiedenen Spalten-Konfigurationen.

2.  $CHECKER$

Sei  $A' \in \{0, 1\}^{m \times n}$  diejenige Matrix, die aus  $A$  durch Vertauschen der  $f$ -ten und  $g$ -ten Zeile von  $A$  entsteht. Da  $CHECKER(A)$  sämtliche Zeilenpaare von  $A$  berücksichtigt, gilt natürlich  $CHECKER(A) = CHECKER(A')$ .

Sei  $\tilde{A}$  diejenige Matrix, die aus  $A$  durch Vertauschen der  $f$ -ten und  $g$ -ten Spalte

von  $A$  entsteht. Gilt für das Paar  $(i_1, j_1)$  mit  $i_1, j_1 \in \{1, \dots, m\}$

$$(i_1, j_1) \in \{(i, j) \mid i, j \in \{1, \dots, m\}, i < j, \text{ für die} \\ a_{i,k} = 1 \Rightarrow a_{j,k} = 0 \text{ für alle } k \in \{1, \dots, n\} \text{ gilt}\}$$

so gilt auch

$$(i_1, j_1) \in \{(i, j) \mid i, j \in \{1, \dots, m\}, i < j, \text{ für die} \\ \tilde{a}_{i,k} = 1 \Rightarrow \tilde{a}_{j,k} = 0 \text{ für alle } k \in \{1, \dots, n\} \text{ gilt}\},$$

denn in beiden Mengen wird sowohl  $k = f$  als auch  $k = g$  betrachtet. Es folgt somit  $CHECKER(A) = CHECKER(\tilde{A})$ .

3.  $C - score$  (nach [31] S. 76)

Es ist  $C - score(A) \stackrel{(7.8)}{=} \frac{2 \sum_{p=1}^{m-1} \sum_{q=p+1}^m C_{p,q}}{m(m-1)}$ . Angenommen es ist  $A'$  diejenige Matrix, die aus  $A$  durch Vertauschen der  $f$ -ten und  $g$ -ten Zeile entsteht.

Es bezeichne nach (7.7)  $C_{p,q}$  die Anzahl an Checkerboard units von  $A$  und  $C'_{p,q}$  die Anzahl an Checkerboard units von  $A'$ , die jeweils aus der  $p$ -ten und  $q$ -ten Zeile gebildet werden können. Es sei dabei ohne Einschränkung der Allgemeinheit stets  $p < q$ .

Es gilt dann für  $p < f$  und  $q < g$

$$C'_{p,q} = C_{p,q}$$

und für alle  $q \in \{1, \dots, m\}$

$$C'_{f,q} = C_{g,q} \text{ und } C'_{g,q} = C_{f,q}.$$

Andere  $C_{p,q}$  sind für die weitere Betrachtung irrelevant, da sie in der Summe nicht betrachtet werden. Es folgt

$$C - score(A') \stackrel{(7.8)}{=} \frac{2 \sum_{p=1}^{m-1} \sum_{q=p+1}^m C'_{p,q}}{m(m-1)} = \frac{2 \sum_{p=1}^{m-1} \sum_{q=p+1}^m C_{p,q}}{m(m-1)} = C - score(A).$$

Auf analoge Weise folgt die Invarianz von  $C - score$  gegenüber Spaltenvertauschungen.

4.  $\bar{S}^2$

Sei  $S = AA^T$  und sei  $A'$  diejenige Matrix, die aus  $A$  durch Vertauschen der  $f$ -ten und  $g$ -ten Spalte entsteht. Dann sei  $S' = A'A'^T$ . Die Einträge von  $S'$

lassen sich dann wie folgt berechnen

$$\begin{aligned}
 s'_{i,j} &= s_{i,j} && \text{für } i, j \in \{1, \dots, m\} \setminus \{f, g\} \\
 s'_{f,j} &= s_{g,j} && \text{für } j \in \{1, \dots, m\} \setminus \{g\} \\
 s'_{f,g} &= s_{g,f} \\
 s'_{g,j} &= s_{f,j} && \text{für } j \in \{1, \dots, m\} \setminus \{f\} \\
 s'_{g,f} &= s_{f,g} \\
 s'_{i,f} &= s_{i,g} && \text{für } i \in \{1, \dots, m\} \setminus \{g\} \\
 s'_{i,g} &= s_{i,f} && \text{für } i \in \{1, \dots, m\} \setminus \{f\} \\
 s'_{f,f} &= s_{g,g} \\
 s'_{g,g} &= s_{f,f}.
 \end{aligned}$$

Damit gilt sofort

$$\bar{S}'^2 = \frac{1}{m(m-1)} \sum_{\substack{p,q=1,\dots,m \\ p \neq q}} s'_{p,q} = \frac{1}{m(m-1)} \sum_{\substack{p,q=1,\dots,m \\ p \neq q}} s_{p,q} = \bar{S}^2. \quad (7.11)$$

Analog folgt die Invarianz von  $\bar{S}^2$  gegenüber Zeilenvertauschungen. ■

Es stellt sich weiter die Frage, welche der verschiedenen Teststatistiken nun die genauesten Ergebnisse liefert. Diese Frage lässt sich nicht explizit beantworten. Keine der Teststatistiken kann eine Konkurrenzsituation exakt wiedergeben, da jeweils nur einige Aspekte der interspezifischen Konkurrenz betrachtet werden. Einige von Diamond's Regeln werden von keiner der vorgestellten Teststatistiken wiedergegeben. Durch die Konstruktionsweise des SIS-Algorithmus werden jedoch auch Aspekte der interspezifischen Konkurrenz aufgefasst. Es werden nur Matrizen mit festen Zeilen- und Spaltensummen generiert. Am Anfang von Kapitel 3 wurden die Gründe dafür erläutert. Es trägt den Unterschieden der Lebensräume und Lebewesen Rechnung. Bei der Erfassung von biologischen Daten ist es oftmals möglich, dass Fehlmessungen in den Daten auftauchen. Da lebendige, sich entwickelnde Lebensräume beobachtet werden, ist eine Beobachtung stets nur eine Momentaufnahme.

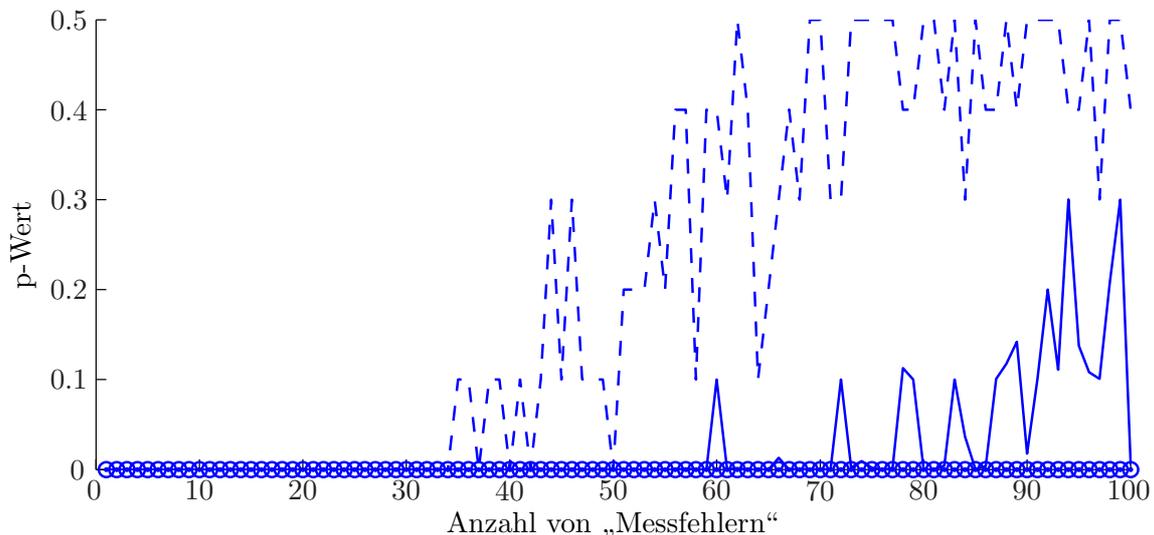
Aus diesem Grund sollte eine Teststatistik nicht zu stark durch auftretende Messfehler beeinflusst werden. Um zu ermitteln, in welchem Maße die vier vorgestellten Statistiken durch Messfehler beeinflusst werden, wird nun zunächst eine sehr strukturierte  $20 \times 20$  Binärmatrix

$$M = \begin{pmatrix} 0_{10 \times 10} & 1_{10 \times 10} \\ 1_{10 \times 10} & 0_{10 \times 10} \end{pmatrix} \quad (7.12)$$

aus [15], wobei  $0_{10 \times 10}$  eine  $10 \times 10$  Nullmatrix und  $1_{10 \times 10}$  eine  $10 \times 10$  Einsmatrix sei, mittels der vier vorgestellten Teststatistiken auf einen Konkurrenzzusammenhang getestet. Bei dieser Matrix sollte jeder Test die Hypothese ablehnen, die einzelnen Arten hätten sich zufällig in den einzelnen Lebensräumen angesiedelt.

Zum Testen wurden mittels des SIS-Verfahrens aus Kapitel 6 jeweils 1000 Matrizen generiert und der p-Wert nach Beispiel 7.1 ermittelt. Dabei wurde der Gewichtsexponent  $u = 0.95$  (vergleiche Abschnitt 6.2.1) als optimal ermittelt und die Zeilen und Spalten der Zeilen- beziehungsweise Spaltensummengröße nach sortiert. Wie  $u$  optimal ermittelt wird, wird in Abschnitt 7.3 erläutert.

Die Beobachtungsmatrix  $M$  wurde dann zufällig an einigen Stellen verändert, das heisst, an Stelle einer 0 eine 1 gesetzt beziehungsweise an Stelle einer 1 eine 0. Dann wurden die Tests erneut durchgeführt und die Veränderungen der p-Werte beobachtet. Das Ergebnis ist in Abbildung 7.1 zu sehen.



**Abbildung 7.1:** Auswirkungen von kleinen Änderungen der Matrix  $M$  aus (7.12) auf den p-Wert bei Tests mit den Teststatistiken *CHECKER* (durchgezogene Linie), *COMBO* (gestrichelte Linie), *C – score* (geschlossene Punkte),  $\bar{S}^2$  (offene Punkte).

Dabei fällt auf, dass  $\bar{S}^2$  und *C – score* robust gegenüber Messfehlern sind, wohingegen *COMBO* am ehesten durch Messfehler verfälscht werden kann und auch *CHECKER* durch kleine Änderungen an der Datenmatrix beeinflusst wird.

Wenn ein Test nicht ausdrücklich mittels *CHECKER* oder *COMBO* durchgeführt werden muss, da gerade die Aspekte, die diese Teststatistiken insbesondere wiedergeben, getestet werden sollen, so ist  $\bar{S}^2$  oder *C – score* vorzuziehen.

### 7.3 Auswertung ökologischer Daten

Mit Hilfe des in Kapitel 6 vorgestellten Algorithmus ist es nun möglich, zu einer Binärmatrix aus beobachteten Daten unabhängige, identisch verteilte Stichproben von Binärmatrizen zu erzeugen, die annähernd gleichverteilt auf der Menge der Binärmatrizen mit festen Zeilen- und Spaltensummen sind.

In diesem Abschnitt werden nun die erwähnten Verfahren auf einige reale ökologische Matrizen angewendet, um den effektiven Nutzen der erläuterten Methoden zu ermitteln. Dazu wurde der in Abschnitt 6.1 vorgestellte Algorithmus auf verschiedene Daten angewendet. Sämtliche Abbildungen wurden aufgrund der ermittelten Daten mit MATLAB<sup>®</sup> R2006a erzeugt.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	r
a	0	0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	14
b	1	1	1	1	1	1	1	1	1	1	0	1	0	1	1	0	0	13
c	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	0	14
d	0	0	1	1	1	0	0	1	0	1	0	1	1	0	1	1	1	10
e	1	1	1	0	1	1	1	1	1	1	0	1	0	1	1	0	0	12
f	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	2
g	0	0	1	1	1	1	1	1	1	0	0	1	0	1	1	0	0	10
h	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1
i	0	0	1	1	1	1	1	1	1	1	0	1	0	0	1	0	0	10
j	0	0	1	1	1	1	1	1	1	1	0	1	0	1	1	0	0	11
k	0	0	1	1	1	0	1	1	0	1	0	0	0	0	0	0	0	6
l	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2
m	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	17
c	4	4	11	10	10	8	9	10	8	9	3	10	4	7	9	3	3	122

**Abbildung 7.2:** Verteilung der Darwin Finken. Jede Zeile repräsentiert eine Finken-Art, jede Spalte eine Galapagos-Insel. Es bezeichnet  $A \hat{=}$  ‘Seymour’,  $B \hat{=}$  ‘Baltra’,  $C \hat{=}$  ‘Isabella’,  $D \hat{=}$  ‘Fernandina’,  $E \hat{=}$  ‘Santiago’,  $F \hat{=}$  ‘Rábida’,  $G \hat{=}$  ‘Pinzón’,  $H \hat{=}$  ‘Santa Cruz’,  $I \hat{=}$  ‘Santa Fe’,  $J \hat{=}$  ‘San Cristóbal’,  $K \hat{=}$  ‘Española’,  $L \hat{=}$  ‘Floreana’,  $M \hat{=}$  ‘Genovesa’,  $N \hat{=}$  ‘Marchena’,  $O \hat{=}$  ‘Pinta’,  $P \hat{=}$  ‘Darwin’,  $Q \hat{=}$  ‘Wolf’,  $a \hat{=}$  ‘Geospiza magnirostris’,  $b \hat{=}$  ‘Geospiza fortis’,  $c \hat{=}$  ‘Geospiza fuliginosa’,  $d \hat{=}$  ‘Geospiza difficilis’,  $e \hat{=}$  ‘Geospiza scandens’,  $f \hat{=}$  ‘Geospiza conirostris’,  $g \hat{=}$  ‘Camarhynchus psittacula’,  $h \hat{=}$  ‘Camarhynchus pauper’,  $i \hat{=}$  ‘Camarhynchus parvulus’,  $j \hat{=}$  ‘Platyspiza crassirostris’,  $k \hat{=}$  ‘Cactospiza pallida’,  $l \hat{=}$  ‘Cactospiza heliobates’,  $m \hat{=}$  ‘Certhidea olivacea’.  $\mathbf{r}$  sind die Zeilen- und  $\mathbf{c}$  die Spaltensummen.

Als Maß der Effizienz des SIS-Verfahrens wird der Chi-Quadrat-Abstand zwischen der jeweiligen Probeverteilung des SIS-Verfahrens zu der Gleichverteilung auf der Menge aller Binärmatrizen mit demselben ZSV und SSV wie die Matrix der beobachteten Daten betrachtet. Dieser Chi-Quadrat-Abstand ist nach Folgerung 4.9 äquivalent zum quadrierten Variationskoeffizienten  $cv^2$ , wobei  $cv^2$  stets durch  $\widehat{cv}^2$  aus 4.9 geschätzt wird. Begonnen wird zunächst mit den oft in der Literatur heran-

gezogenen Darwinschen Finken. Dieses Beispiel verdeutlicht dabei die übliche Herangehensweise bei einer derartigen ökologischen Problemstellung.

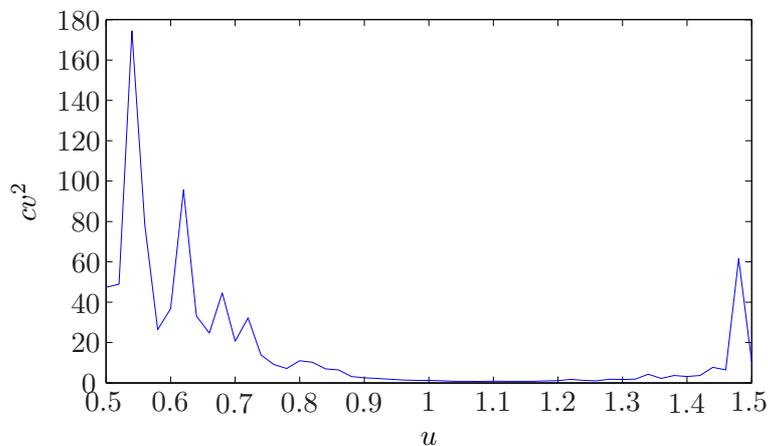
### 7.3.1 Darwins Finken

Charles Darwin beobachtete auf seiner Reise mit der *Beagle* zu den Galapagos-Inseln 13 verschiedene Finken-Arten auf 17 Inseln. Aus den Beobachtungen resultiert die Binärmatrix, die in Abbildung 7.2 wiedergegeben ist.

Mit Hilfe des in Kapitel 6 angegebenen Verfahrens wurden nun basierend auf der aus Abbildung 7.2 resultierenden Binärmatrix, die im Folgenden mit  $D$  bezeichnet wird, weitere, auf der Menge  $\Sigma_{rc}$  annähernd gleichverteilte Binärmatrizen generiert. Zunächst sind zur Optimierung des SIS-Verfahrens jeweils 1000 Matrizen zur Ermittlung des optimalen Gewichtsexponenten  $u$  und der besten Anpassung der Datenmatrix  $D$  (siehe Abschnitt 6.2) generiert worden.

Wurden die Spalten nicht sortiert und zunächst  $u = 1$  gelassen, ergab sich ein  $\widehat{cv}^2$  von 25.6258. Bei der Größe nach sortierten Spalten ergab sich für  $u = 1$  hingegen ein  $\widehat{cv}^2$  von 1.2452, woraus ersichtlich ist, dass sich dabei die Effizienz deutlich verbessert hat.

Im Folgenden seien die Spalten von  $D$  der Größe nach sortiert.

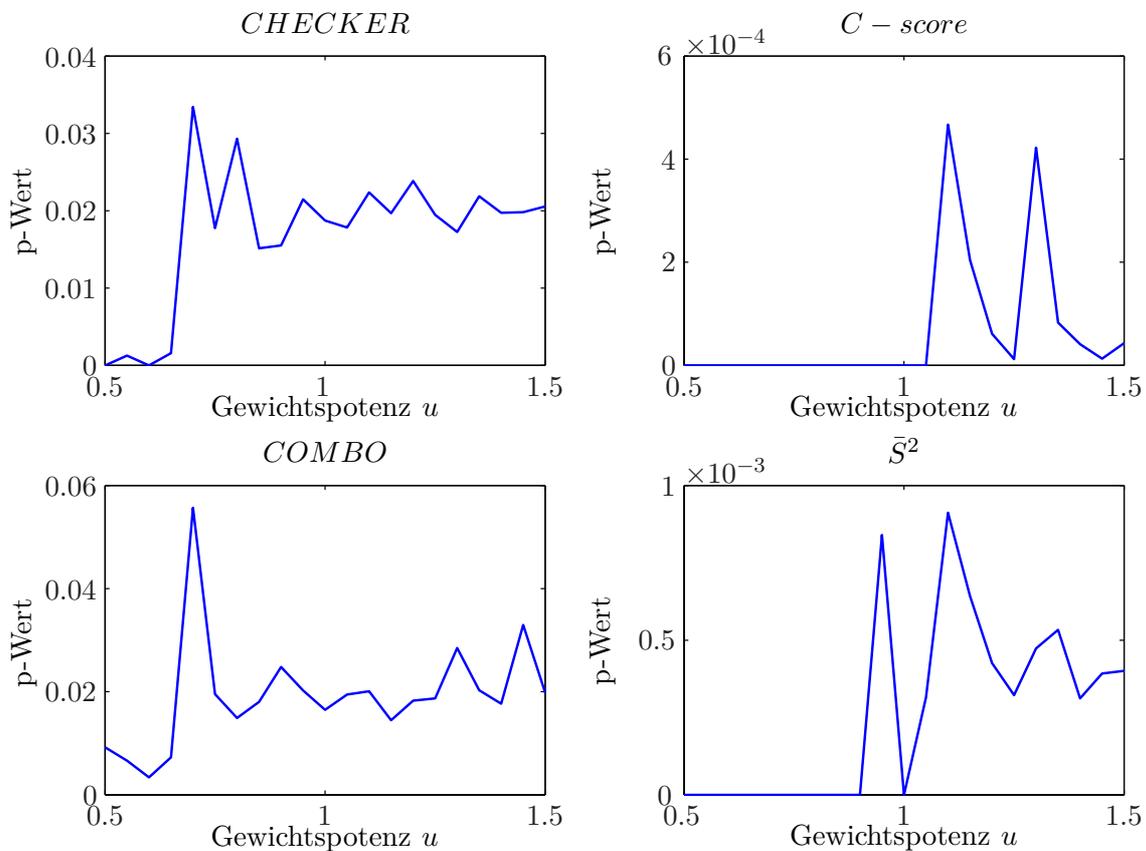


**Abbildung 7.3:** Entwicklung von  $\widehat{cv}^2$  bei verändertem Gewichtsexponenten  $u$  basierend auf jeweils 1000 mittels SIS zu den Darwin Finken generierten Matrizen.

Um  $u$  optimal zu bestimmen wurde  $u$  zwischen 0.5 und 1.5 mit einer Schrittweite von 0.05 variiert. Das Resultat ist in Abbildung 7.3 zu sehen. Dabei ist ersichtlich, dass sich  $\widehat{cv}^2$  um den optimalen Wert für  $u$  nur wenig ändert. Der optimale Wert ist dabei 1.1.

Wendet man das SIS-Verfahren statt auf  $D$  auf  $D^T$  an, um die Matrizen zeilenweise zu generieren (vergleiche Abschnitt 6.2.2), so ergibt sich minimal ein  $\widehat{cv}^2$  von

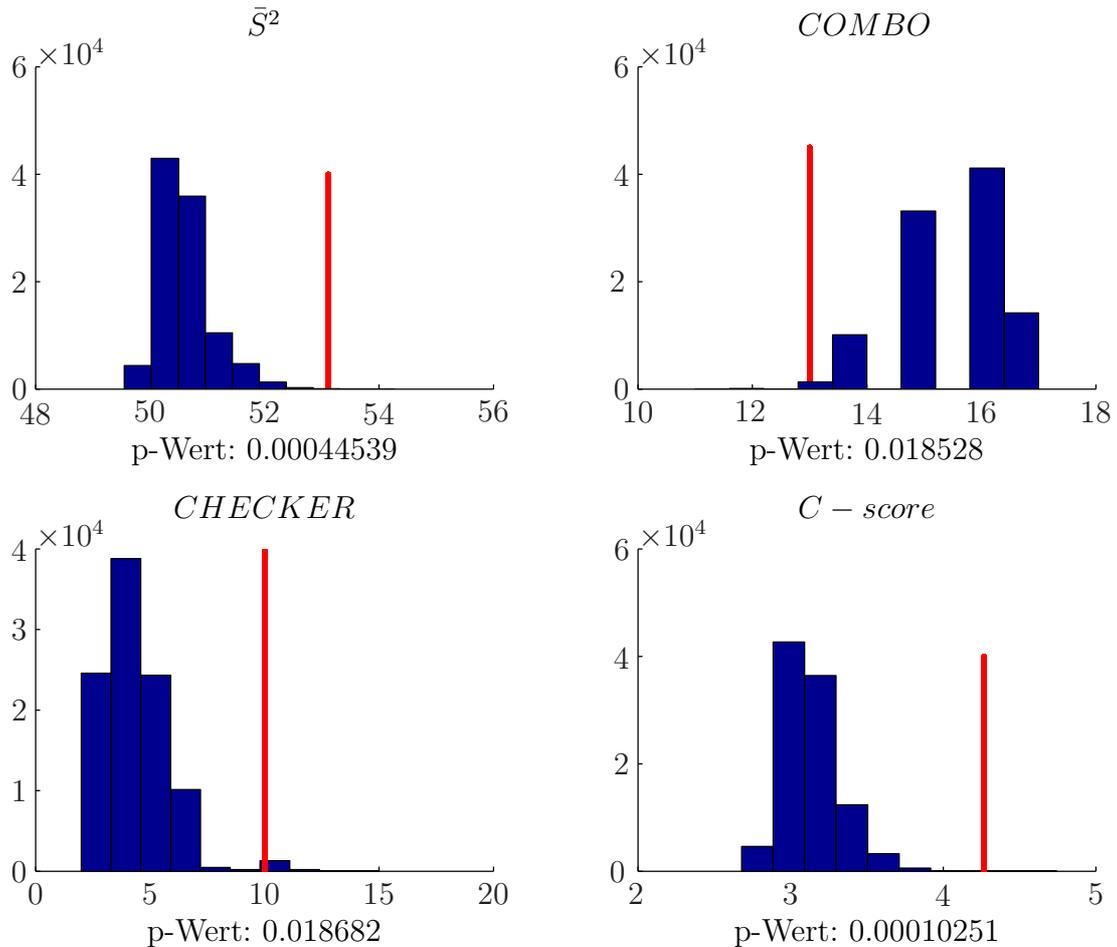
10.1363. Aus diesem Grund wird in den weiteren Betrachtungen der Darwin-Finken von einer zeilenweisen Generierung abgesehen.



**Abbildung 7.4:** Mittels der Teststatistiken  $\bar{S}^2$ , COMBO, CHECKER und  $C - score$  ermittelte p-Werte basierend auf jeweils 10000 Matrizen bei unterschiedlicher Wahl des Gewichtsexponenten  $u$ .

In Abbildung 7.4 ist ersichtlich, dass es notwendig ist,  $u$  zu optimieren, denn die mit verschiedenen Teststatistiken ermittelten p-Werte variieren bei unterschiedlicher Wahl von  $u$  stark, sodass die Wahl von  $u$  Auswirkungen auf die Entscheidung bei einem statistischen Test haben kann.

Zur Auswertung der Matrix  $D$  wurden nun weitere 100000 Binärmatrizen mittels des SIS-Verfahrens generiert, wobei die Spalten von  $D$  der Größe nach sortiert wurden und als Gewichtsexponent  $u = 1.1$  angenommen wurde. Es ergab sich ein  $\widehat{cv}^2$  von 0.6895 und  $6.684120 \cdot 10^{16}$  als Schätzung für  $|\Sigma_{rc}|$ . Mittels der in Abschnitt 7.2 vorgestellten Teststatistiken wurde die so erlangte Stichprobe ausgewertet. Wie aus Abbildung 7.5 hervorgeht, würde die Nullhypothese der zufälligen Verteilung der Finken bei einem Signifikanzniveau von  $\alpha = 0.05$  stets abgelehnt werden. Ein Einfluss von interspezifischer Konkurrenz bei der Lebensraumwahl dieser Finken ist dann also sehr wahrscheinlich.



**Abbildung 7.5:** Verteilungen der Teststatistiken  $\bar{S}^2$ , *COMBO*, *CHECKER* und *C - score* basierend auf 100000 Matrizen. Die vertikale Linie beschreibt jeweils den entsprechenden Wert der Original-Datenmatrix *D*.

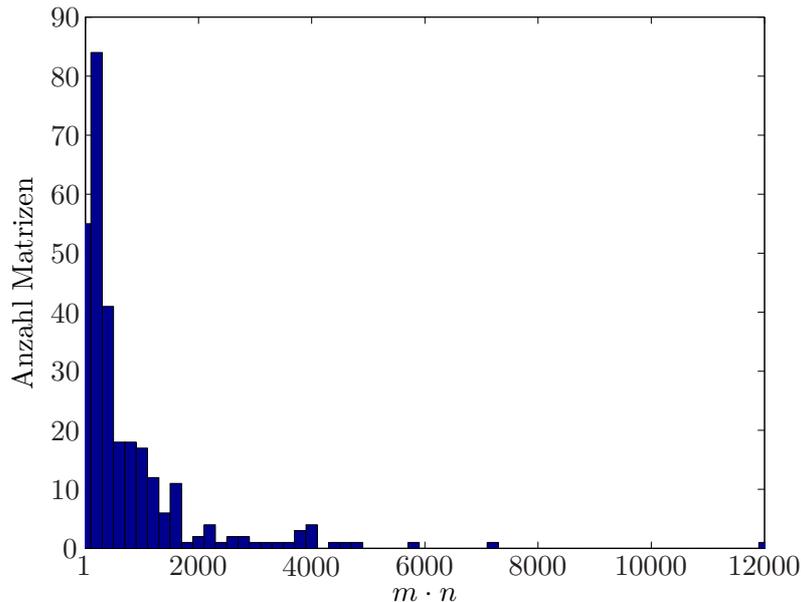
Es fällt jedoch auf, dass bei den unterschiedlichen Teststatistiken sehr unterschiedliche p-Werte ermittelt wurden. Sowohl mit *COMBO* als auch mit *CHECKER* würde bei einem durchaus üblichen Signifikanzniveau von  $\alpha = 0.01$  die Nullhypothese nicht abgelehnt werden. Allerdings wurde bereits in Kapitel 7 erwähnt, dass diese beiden Teststatistiken nicht sehr robust gegenüber kleinen Änderungen in der Datenmatrix sind.

Es werden außerdem durch die unterschiedlichen Teststatistiken jeweils nur einige wenige Parameter des Zusammenlebens erfasst, sodass immer eine Restunsicherheit darüber bleibt, ob tatsächlich ein Konkurrenzzusammenhang besteht oder nicht.

In [7] wird der tatsächliche Wert  $|\Sigma_{rc}|$  als 67149106137567626 angegeben. Der exakte Wert wurde also recht gut geschätzt.

### 7.3.2 Patterson Atmar Datensatz

In dem in [1] veröffentlichten Programm sind 291 Beispiele von Binärmatrizen ökologischer Daten enthalten, die aus vielen verschiedenen Veröffentlichungen zusammengetragen wurden. Es sind sowohl Daten über verschiedene Insekten als auch über Vögel und Säugetiere enthalten. Das Spektrum der Dimensionen der Matrizen reicht von sehr klein ( $3 \times 1$ ) bis sehr groß ( $20 \times 597$ ).



**Abbildung 7.6:** Histogramm der Größen der 291 Matrizen des Patterson Atmar Datensatzes. Es ist dabei  $m$  die Anzahl der Zeilen einer Matrix und  $n$  die Anzahl der Spalten.

Aus Abbildung 7.6 geht hervor, welche Größen von Matrizen in dem Datensatz enthalten sind.  $m \cdot n$  gibt dabei die potentielle Anzahl möglicher Kombinationen von Arten und Lebensräumen in den jeweiligen Matrizen wieder.

Die Anzahl an Binärmatrizen, die ökologische Daten repräsentieren können, ist durchaus beschränkt. Sowohl die Anzahl interessierender Arten als auch die Anzahl an Lebensräumen, die diese Arten besiedeln können, als auch die Mittel, diese Daten zu erfassen, sind begrenzt. Dementsprechend soll der Patterson Atmar Datensatz diese ökologischen Binärmatrizen repräsentieren.

Mit diesem Datensatz wird im Folgenden die Effizienz des SIS-Verfahrens aus Kapitel 6 auf die Probe gestellt .

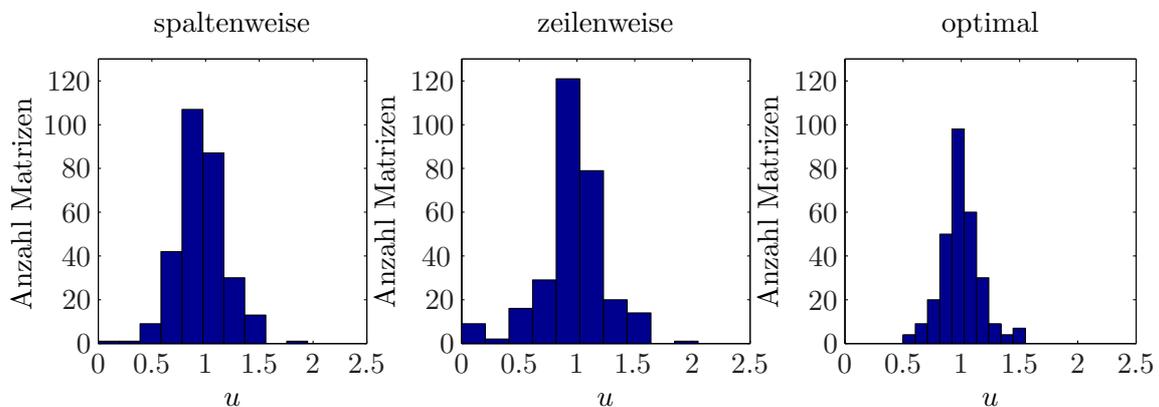
Es wurden zu jeder einzelnen Matrix des Datensatzes mittels des SIS-Verfahrens weitere Matrizen mit gleichen Zeilen- und Spaltensummen generiert. Dabei wurden zunächst die optimalen Parameter für das SIS-Verfahren bestimmt, um den Chi-Quadrat-Abstand der tatsächlichen Verteilung der generierten Matrizen zu der Gleichverteilung über alle Matrizen mit dem gleichen ZSV und SSV wie die jeweils

zugrunde liegende Datenmatrix zu minimieren. Dazu wurde ähnlich wie bei den Darwin-Finken zu jeder Matrix der Gewichtsexponent  $u$  zunächst zwischen 0.5 und 1.5 mit einer Schrittweite von 0.05 variiert.

Bei denjenigen Matrizen, bei denen  $u \in \{0.5, 1.5\}$  als optimal ermittelt wurde, wurden die Grenzen für  $u$  auf 0 und 2.5 erweitert. In diesem Bereich lag bei allen untersuchten Matrizen ein Minimum für den Chi-Quadrat-Abstand. Dieser wurde jeweils mit 100 mittels des SIS-Verfahrens generierten Matrizen durch  $\widehat{cv}^2$  im Sinne von Folgerung 4.9 geschätzt. Dies wurde sowohl zeilenweise als auch spaltenweise durchgeführt, um auch eine Verkleinerung des Chi-Quadrat-Abstandes durch die Ausrichtung des SIS-Verfahrens zu erfassen.

Es sei zunächst angemerkt, dass die benötigte Rechenzeit in der Regel kürzer ist, wenn diejenige Dimension der Datenmatrix größer ist, auf die das SIS-Verfahren angewendet wird. Ist also die Zeilenanzahl kleiner als die Spaltenanzahl, so ist das SIS-Verfahren spaltenweise angewandt schneller als zeilenweise angewandt.

Dies folgt daraus, dass je größer die zu generierenden Spalten beziehungsweise Zeilen sind, die mittels Newton's Identitäten zu berechnenden R-Funktionen (vergleiche Satz 2.42) mehr Rechenzeit beanspruchen. Eine Verbesserung der Rechenzeit ist allerdings nicht gleichbedeutend mit einer Verbesserung des SIS-Verfahrens beziehungsweise einer Verkleinerung des Chi-Quadrat-Abstandes zwischen der Probeverteilung und der Zielverteilung.

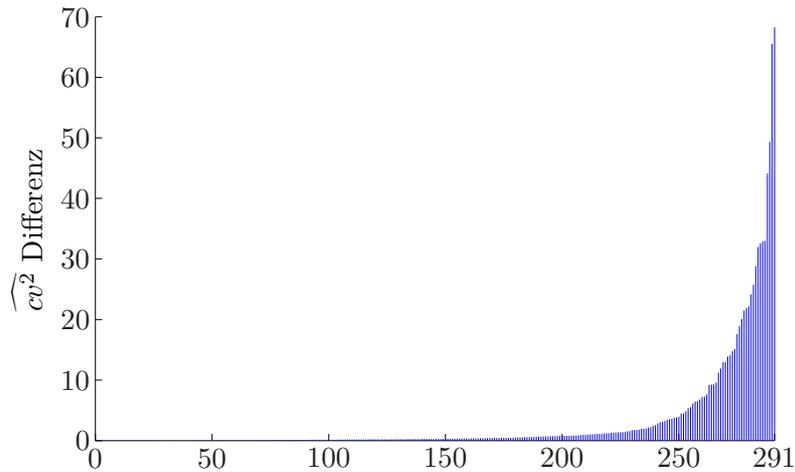


**Abbildung 7.7:** Verteilung der 291 in Bezug zum minimalen Chi-Quadrat-Abstand optimalen Gewichtsexponenten  $u$  zum Patterson Atmar Datensatz bei spaltenweiser beziehungsweise zeilenweiser Generierung und der zur Minimierung von  $\widehat{cv}^2$  optimalen Ausrichtung des SIS-Verfahrens.

Aus Abbildung 7.7 geht hervor, dass der optimale Wert für den Gewichtsexponenten  $u$  bei den meisten Matrizen sehr nahe bei 1 liegt. Dabei verstärkt sich dieser Effekt noch, wenn auch die Ausrichtung des SIS-Verfahrens optimiert wird.

Bei optimaler Ausrichtung des SIS-Verfahrens ist bei 62, also bei mehr als 20% der Matrizen,  $u = 1$  bereits optimal. Andererseits geht aber aus Abbildung 7.8 hervor,

dass bei einigen derjenigen Matrizen, bei denen  $u = 1$  nicht optimal ist, durch Anpassung von  $u$  eine erhebliche Verkleinerung von  $\widehat{cv}^2$  erreicht werden kann.



**Abbildung 7.8:** Differenz der, basierend auf jeweils 100 mittels des SIS-Verfahrens zu den 291 Matrizen des Patterson Atmar Datensatzes generierten Matrizen, ermittelten Werte  $\widehat{cv}^2$  bei  $u = 1$  beziehungsweise optimiertem Gewichtsexponenten  $u$ . Die Differenzen wurden der Größe nach sortiert.

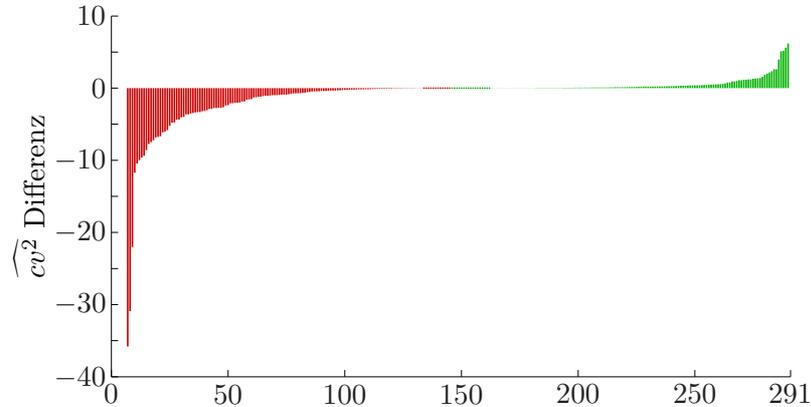
In Abbildung 7.9 ist dargestellt, wie sich  $\widehat{cv}^2$  bei jeweils optimalem Wert  $u$  ändert, wenn das SIS-Verfahren zeilenweise statt spaltenweise durchgeführt wird. Dazu wurde zunächst auch für die zeilenweise Generierung der optimale Wert für  $u$  bestimmt. Es ist ersichtlich, dass in 150 der 291 Fälle die zeilenweise Generierung der Vergleichsmatrizen eine teils enorme Verbesserung von  $\widehat{cv}^2$  zur Folge hat. Dies zeigt deutlich, dass die Bestimmung der optimalen Ausrichtung des SIS-Verfahrens dessen Effizienz steigern kann.

Das Verhältnis zwischen Zeilen- und Spaltenanzahl der Datenmatrizen spielt dabei weniger eine Rolle als die tatsächliche Gestalt der Matrizen. Bei 66 der 291 Matrizen des Datensatzes ist die Zeilenanzahl größer als die Spaltenanzahl und in 33, also exakt 50%, dieser 66 Fälle liefert die zeilenweise Generierung einen kleineren Wert für  $\widehat{cv}^2$ . Demnach ist es nicht möglich, aufgrund der Dimensionen einer Datenmatrix direkt auf die optimale Ausrichtung des SIS-Verfahrens zu schließen.

Für möglichst genaue Ergebnisse der auf die generierten Matrizen angewandten Tests ist es also erforderlich, vor der eigentlichen Auswertung der Matrizen Probelaufgänge des SIS-Verfahrens zur Bestimmung der optimalen Ausrichtung des SIS-Verfahrens und des optimalen Gewichtsexponenten  $u$  durchzuführen.

Es steht außer Frage, dass für die Aussagekraft der mit Hilfe der in dieser Arbeit verwendeten Teststatistiken ermittelten p-Werte zu der Hypothese „interspezifische Konkurrenz spielte keine Rolle für die in der behandelten Datenmatrix erfassten Arten bei der Besiedlung der erfassten Lebensräume“ gegen die Alternative „inter-

spezifische Konkurrenz hat die Arten bei ihrer Lebensraumwahl beeinflusst“ durch das SIS-Verfahren die Gleichverteilung auf der jeweiligen Menge  $\Sigma_{rc}$  möglichst genau approximiert werden muss.



**Abbildung 7.9:** Differenz der Werte  $\widehat{cv}^2$  bei spaltenweiser beziehungsweise zeilenweiser Generierung mit jeweils optimalem Gewichtsexponenten  $u$  basierend auf jeweils 100 mittels SIS generierten Matrizen zu den 291 Matrizen des Patterson Atmar Datensatzes. Bei negativer Differenz (rot) ergibt die spaltenweise Generierung einen kleineren  $\widehat{cv}^2$ , bei positiver Differenz (grün) die zeilenweise Generierung. Die Werte wurden der Größe nach sortiert.

Allerdings könnte es sein, dass eine Änderung des Gewichtsexponenten  $u$  oder der Ausrichtung des SIS-Verfahrens keinerlei Auswirkung auf die Testergebnisse bei einer konkreten Matrix hätte.

	$\alpha = 0.05$			$\alpha = 0.01$		
	original	optimiert	beide	original	optimiert	beide
<i>CHECKER</i>	36	48	24	29	32	19
<i>C - score</i>	114	120	106	95	104	90
<i>COMBO</i>	63	58	42	50	44	33
$\bar{S}^2$	98	81	72	83	61	56

**Tabelle 7.1:** Anzahl der mittels der jeweiligen Teststatistiken zu den 291 Matrizen des Patterson Atmar Datensatzes ermittelten p-Werte, die kleiner als das jeweils angegebene Niveau  $\alpha \in \{0.01, 0.05\}$  sind. Bei „original“ wurde das SIS-Verfahren nicht angepasst, bei „optimiert“ wurden  $u$  und die Ausrichtung des SIS-Verfahrens optimiert und „beide“ gibt die Schnittmenge der ersten beiden wieder. Die p-Werte wurden jeweils durch 100 mittels des SIS-Verfahrens generierten Matrizen wie in Beispiel 7.1 geschätzt.

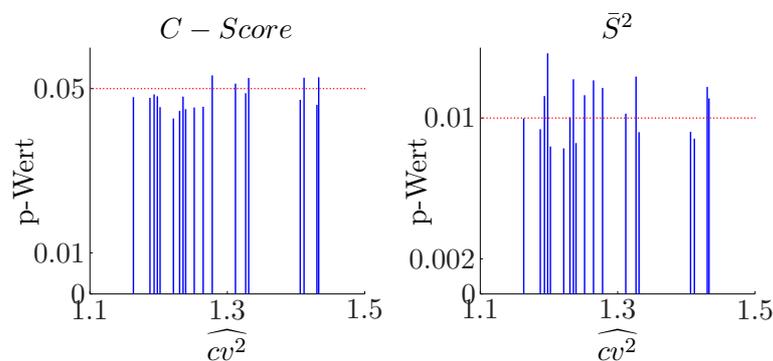
Zur näheren Betrachtung dieser Annahme wurden in Tabelle 7.1 die jeweils mittels 100 durch das SIS-Verfahren generierten Matrizen ermittelten p-Werte mit  $u = 1$  und spaltenweiser Generierung mit den jeweils entsprechenden p-Werten bei optimalem Gewichtsexponenten  $u$  und optimaler Ausrichtung des SIS-Verfahrens verglichen. Dabei ist festzustellen, dass die Ergebnisse sehr unterschiedlich ausfallen können. So würden bei einem Signifikanzniveau  $\alpha = 0.05$  durch die Teststatistik

*CHECKER* nur bei der Hälfte derjenigen Matrizen, bei denen mittels eines optimierten SIS-Verfahrens die oben genannte Hypothese abgelehnt wird, diese ebenfalls abgelehnt. Andererseits würde bei 12 anderen Matrizen die Hypothese abgelehnt werden, obwohl beim optimierten SIS-Verfahren nichts gegen diese spricht.

Diese Unterschiede sind bei sämtlichen Teststatistiken und Niveaus festzustellen. Die aus solchen verfälschten p-Werten resultierenden Fehlentscheidungen können mehr oder weniger schwerwiegende Folgen haben. Ist der Fehler zweiter Art erhöht, spricht also aufgrund eines verfälschten p-Wertes nichts gegen die Hypothese, so müssten erneute Untersuchungen angestellt werden, um einen Erkenntnisgewinn zu erzielen.

Ist hingegen der Fehler erster Art erhöht, wird also aufgrund eines verfälschten p-Wertes die Hypothese abgelehnt, so wird fälschlicherweise eine Behauptung bestätigt, auf deren Grundlage dann im schlimmsten Fall weitere Fehlentscheidungen aufbauen.

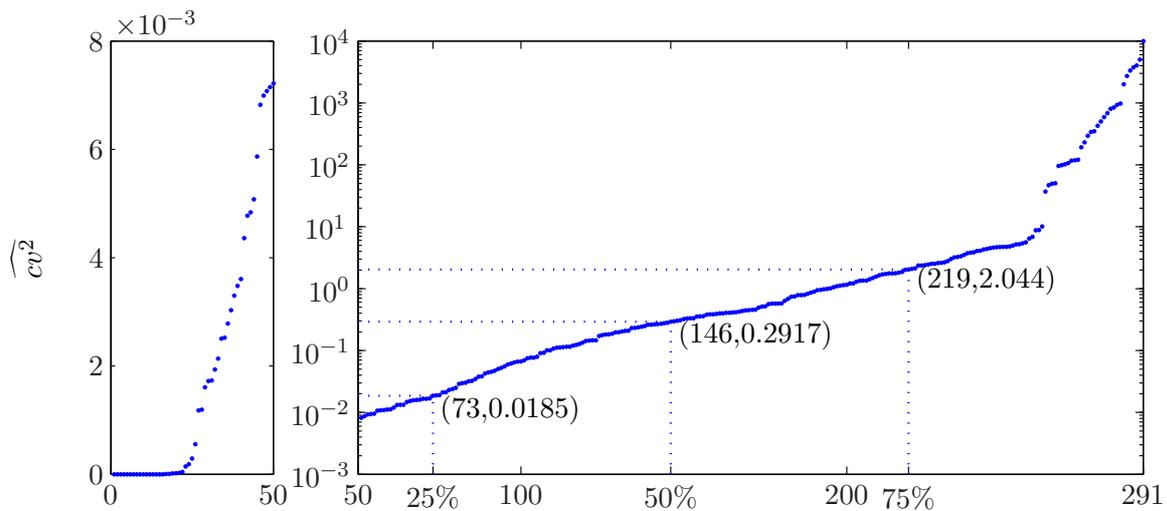
Zur weiteren Auswertung des Patterson Atmar Datensatzes wurde nun zu jeder Matrix aus dem Datensatz mit Hilfe des SIS-Verfahrens mit optimaler Ausrichtung und optimalem Gewichtsexponenten  $u$  eine weitere unabhängige Stichprobe von jeweils 10000 Binärmatrizen generiert.



**Abbildung 7.10:** Schwankungen von p-Werten, die jeweils mittels 10000 mit dem SIS-Verfahren, unter leichten Änderungen des Gewichtsexponenten  $u$  und somit auch  $\widehat{cv^2}$  generierten Matrizen ermittelt wurden. Die zugrunde liegende Datenmatrix ist Matrix Nr.40 aus dem Patterson Atmar Datensatz. Als Teststatistiken wurden  $C - score$  und  $\bar{S}^2$  angewendet.

Auch bei optimierten Gewichtsexponenten und angepasster Ausrichtung des SIS-Verfahrens können Fehlentscheidungen getroffen werden. Je größer die Schätzung für  $cv^2$  aus den generierten Matrizen ist, um so geringer ist der effektive Stichprobenumfang  $ESS$ .

In Abschnitt 7.3.1 wurde im Falle der Darwin Finken ein Wert für  $\widehat{cv^2}$  nahe 1 ermittelt und die aus den generierten Matrizen resultierenden Schätzungen für  $|\Sigma_{rc}|$  sowie die unterschiedlichen p-Werte gaben die Realität recht gut wieder. Je größer  $\widehat{cv^2}$  ist, um so kleiner ist der effektive Stichprobenumfang  $ESS$  (vergleiche Definition 4.5).



**Abbildung 7.11:** Verteilung der aufgrund jeweils 10000 mittels des SIS-Verfahrens mit optimiertem Gewichtsexponenten und optimaler Ausrichtung generierter Binärmatrizen ermittelten Schätzungen  $\widehat{cv^2}$  zu den 291 Binärmatrizen des Patterson Atmar Datensatzes. Die Werte wurden der Größe nach sortiert. Die linke Abbildung ist linear, die rechte logarithmisch skaliert.

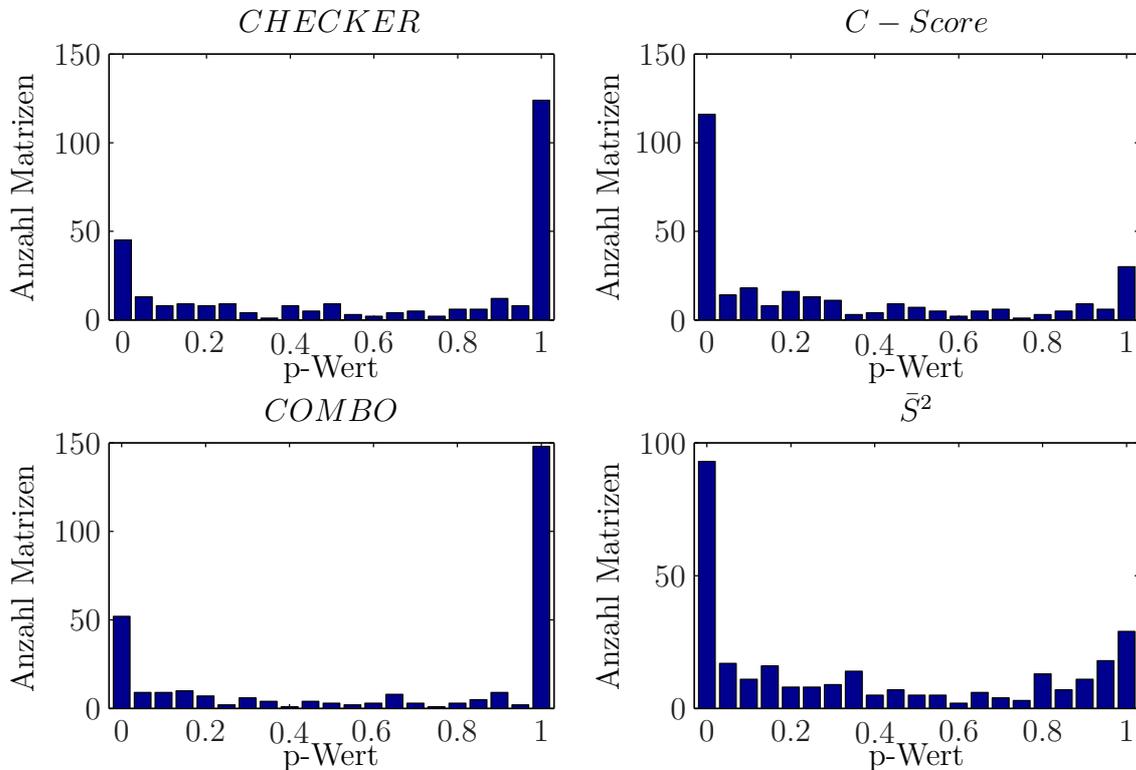
In Abbildung 7.10 ist zu sehen, dass schon kleine Änderungen des geschätzten Chi-Quadrat-Abstandes die resultierenden p-Werte und somit letztendlich die zu treffenden Entscheidungen beeinflussen können.

In Abbildung 7.11 ist die Verteilung der Schätzungen für  $cv^2$  bei den Matrizen des Patterson Atmar Datensatzes wiedergegeben. Daraus ist ersichtlich, dass bei rund 50% der 291 Matrizen  $\widehat{cv^2} \leq 0.2917$  gilt. Dies ist gleichbedeutend damit, dass in diesen Fällen die Gleichverteilung auf der Menge aller Matrizen mit den gleichen ZSV und SSV wie die jeweils zugrunde liegende Datenmatrix durch die je 10000 generierten Binärmatrizen recht gut wiedergegeben wurde. Für den effektiven Stichprobenumfang gilt dabei  $ESS \geq 7741.74$ . Bei 10 Matrizen war  $\widehat{cv^2} = 0$ , sodass in diesen Fällen die Gleichverteilung exakt wiedergegeben wurde.

Bei 75% der Matrizen ist  $\widehat{cv^2} \leq 2.044$  oder gleichbedeutend  $ESS \geq 3285.15$ . In den restlichen 25% der Fälle sind die mittels der jeweils generierten Matrizen ermittelten p-Werte zu den verschiedenen Teststatistiken kaum aussagekräftig, da dabei nicht von gleichverteilten Matrizen ausgegangen werden kann. Die Teils sehr hohen Schätzungen für  $cv^2$  resultieren auch aus der mangelnden Robustheit von  $\widehat{cv^2}$  gegenüber Ausreißern.

Liegt bei 10000 Matrizen für 9999 dieser Matrizen die Wahrscheinlichkeit des SIS-Verfahrens konstant bei  $10^{-8}$ , so genügt es für die letzte Matrix mit einer Wahrscheinlichkeit von  $10^{-11}$  generiert worden zu sein, um  $\widehat{cv^2} = 82.4944$  zu erreichen. Ist die letzte Matrix gar mit einer Wahrscheinlichkeit von  $10^{-13}$  generiert worden,

so ergibt sich bereits  $\widehat{cv^2} = 8264.4477$ . Dies entspricht einem effektiven Stichprobenumfang von  $ESS = \frac{10000}{1+8264.4477} = 1.2099$ .

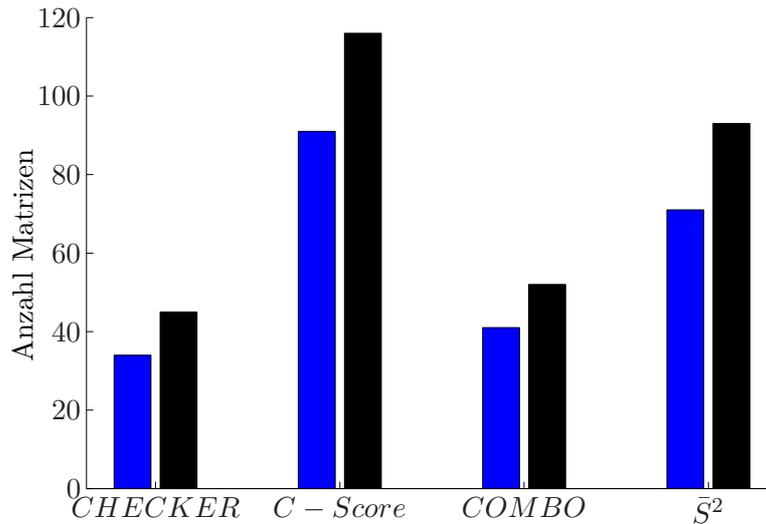


**Abbildung 7.12:** Histogramm der aufgrund jeweils 10000 mittels des SIS-Verfahrens mit optimiertem Gewichtsexponenten und optimaler Ausrichtung generierter Binärmatrizen ermittelten p-Werte zu den Binärmatrizen des Patterson Atmar Datensatzes.

Für die zu jeder Matrix des Patterson Atmar Datensatzes mittels des SIS-Verfahrens mit optimalem Gewichtsexponenten und optimaler Ausrichtung generierten 10000 Matrizen wurden nun wie in Beispiel 7.1 die p-Werte zu den Teststatistiken  $C$ -score,  $CHECKER$ ,  $COMBO$  und  $\bar{S}^2$  ermittelt.

In Abbildung 7.12 sind die ermittelten p-Werte aufgelistet. Dabei ist ersichtlich, dass sich die p-Werte unter den unterschiedlichen Teststatistiken stark unterscheiden. In Abbildung 7.13 ist die Anzahl der Matrizen aus dem Patterson Atmar Datensatz erfasst, bei denen aufgrund der ermittelten p-Werte mit den üblichen Testniveaus  $\alpha = 0.01$  und  $\alpha = 0.05$  im Falle der vier vorgestellten Teststatistiken die Nullhypothese abgelehnt wird.

Daraus geht hervor, dass durch  $CHECKER$  und  $COMBO$  deutlich seltener die Nullhypothese abgelehnt wurde, wie durch  $C$ -score oder  $\bar{S}^2$ . Wie schon am Ende von Abschnitt 7.2.1 erläutert wurde, sind  $CHECKER$  und  $COMBO$  nicht robust gegen Abweichungen der Datenmatrix. Es wurde festgestellt, dass kleine Änderungen an der Datenmatrix (7.12) die mittels dieser Statistiken ermittelten p-Werte



**Abbildung 7.13:** Anzahl derjenigen Matrizen, bei denen die mittels 10000 mit dem optimalen SIS-Verfahren ermittelten p-Werte zu den Teststatistiken *C - score*, *CHECKER*, *COMBO* und  $\bar{S}^2$  unter dem Niveau  $\alpha = 0.01$  (blau) beziehungsweise  $\alpha = 0.05$  (schwarz) lagen.

erhöhen können.

Da nicht auszuschließen ist, dass auch in den Binärmatrizen des Patterson Atmar Datensatzes Messfehler enthalten sind, könnte dies ein Grund für die unterschiedlichen p-Werte bei den unterschiedlichen Teststatistiken sein. Ein anderer Grund ist die unterschiedliche Beschaffenheit der Teststatistiken, die, wie in Abschnitt 7.2 erläutert wurde, unterschiedliche Ansätze zur Beurteilung der interspezifischen Konkurrenz als Grundlage benutzen.

## 7.4 Fazit

Wie in Abschnitt 7.3 ermittelt wurde, ist bei vielen Binärmatrizen, die auf ökologischen Daten beruhen, die Probeverteilung des SIS-Verfahrens der Gleichverteilung auf der jeweiligen Menge  $\Sigma_{rc}$  sehr ähnlich. Das heißt, dass in diesen Fällen die Gleichverteilung durch die bedingten Bernoulli-Modelle sehr gut wiedergegeben wurde. Für diese Matrizen ist das SIS-Verfahren also ein gutes Mittel, unabhängige Stichproben von Binärmatrizen zu erzeugen und mit Hilfe dieser und geeigneter Teststatistiken Tests zu der Hypothese „interspezifische Konkurrenz spielt keine Rolle für die in der behandelten Datenmatrix erfassten Arten bei der Besiedlung der erfassten Lebensräume“ gegen die Alternative „interspezifische Konkurrenz hat die Arten bei ihrer Lebensraumwahl beeinflusst“ durchzuführen.

Das SIS-Verfahren bietet gegenüber anderen Methoden zur Stichproben-Generierung, wie zum Beispiel MCMC, den Vorteil, dass die erzeugten Stichproben unabhängig

sind. Auch ist die Verteilung der erzeugten Matrizen explizit gegeben und lässt eine Schätzung für die jeweilige Größe  $|\Sigma_{rc}|$  beziehungsweise der dadurch gegebenen Gleichverteilung über sämtliche Matrizen mit dem gleichen ZSV und SSV wie der zugrunde liegenden Datenmatrix zu.

Der wohl größte Nachteil des SIS-Verfahrens besteht darin, dass die erzeugten Matrizen nur annähernd gleichverteilt sind. Je größer der Chi-Quadrat-Abstand der Probeverteilung zur jeweiligen Gleichverteilung ist, um so kleiner ist  $ESS$  und somit die Aussagekraft der jeweiligen Stichprobe.

Es ist also nicht möglich, das SIS-Verfahren auf jede Datenmatrix effektiv anzuwenden. Um zu bestimmen, ob das SIS-Verfahren bei einer konkreten Datenmatrix anwendbar ist, müssen einige Probedurchläufe durchgeführt werden. Dabei müssen zunächst die optimalen Bedingungen für das SIS-Verfahren ermittelt werden. Erst danach entscheidet sich, ob das Verfahren bei dieser Datenmatrix zuverlässige Ergebnisse erzielt.

# 8 Anhang

## 8.1 Der SIS-Algorithmus

Es folgt der in C implementierte SIS-Algorithmus.

```
/*
 * Ein Monte-Carlo Algorithmus zur Generierung von Binaermatrizen
 * mit gegebenen Zeilen- und Spaltensummen mittels 'sequential
 * Importance Sampling'. Mit Hilfe der bedingten Bernoulli-Verteilung
 * wird die Gleichverteilung angenaehert.
 */

#include <math.h>
#include <malloc.h>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

static double CB(int Anz,int Ende,int Start,int A_n[],double Gew[],
                int GewDim);
static long double P(int i, double Gew[], int Start, int Ende);
static double R(int k, double Gew[], int Start, int Ende);
static double SIS_CB(int M[],int AnzZln,int AnzSp, double GewExp,
                    int ZlnSum[],int SpSum[],int *pGExp);
static void SpKonj(int SpSum[],int AnzZln,int AnzSp, int l,
                  int SpKonj[]);
static double cv_2(double ProbVek[],int n);
static int Checker(int *M,int AnzZln,int AnzSp);
static int Combo(int *M,int AnzZln,int AnzSp);
static double CScore(int *M,int AnzZln,int AnzSp);
static double Squadrat(int *M,int AnzZln,int AnzSp);

int main()
{
double GewExp,cv2,cv2min,Prob,*ProbVek=NULL,*SquadratM=NULL,
SquadratM_0,*CScoreM=NULL,CScoreM_0,*CheckerM=NULL,CheckerM_0,
*ComboM=NULL,ComboM_0,SumProb,pCScore,pCombo,pChecker,pSquadrat,
```

---

```

        pCScoreCvMin, pComboCvMin, pCheckerCvMin, pSQuadratCvMin, SigmaRC;
int AnzStich, AnzZln, AnzSp, N, *ZlnSum=NULL, *ZlnSumOrig=NULL,
        *SpSum=NULL, *SpSumOrig=NULL, *M_0=NULL, *M=NULL, ProbExp,
        *ProbExpVek=NULL, ProbExpMin, i, j, k, l, n, Temp, iter, SigmaRCExp;
FILE *Eingabe, *Ausgabe;
char Dateiname[20];

srand(time(NULL));
/*
 * Einlesen der Matrix und Vorbereiten des SIS-Verfahrens
 */
printf("Bitte den Dateinamen eingeben!\n");
scanf("%s", Dateiname);
while((Eingabe=fopen(Dateiname, "r"))==NULL){
    printf("\nDatei wurde nicht gefunden, bitte erneut eingeben!\n");
    scanf("%s", Dateiname);
}
fscanf(Eingabe, "%d", &AnzZln);
fscanf(Eingabe, "%d", &AnzSp);
printf("\nBitte den Gewichtsexponent u eingeben!\n");
scanf("%lg", &GewExp);
printf("\nBitte den Stichprobenumfang eingeben!\n");
scanf("%d", &N);
printf("\nWieviele Stichproben sollen durchgefuehrt werden?\n");
scanf("%d", &AnzStich);
/*
 * Bereitstellen von Speicherplatz
 */
M_0=malloc((AnzZln*AnzSp)*sizeof (int));
M=malloc((AnzZln*AnzSp)*sizeof (int));
ZlnSum=malloc(AnzZln*sizeof (int));
SpSum=malloc(AnzSp*sizeof (int));
ZlnSumOrig=malloc(AnzZln*sizeof (int));
SpSumOrig=malloc(AnzSp*sizeof (int));
ProbVek=malloc(N*sizeof (double));
ProbExpVek=malloc(N*sizeof (int));
SQuadratM=malloc(N*sizeof(double));
CScoreM=malloc(N*sizeof(double));
CheckerM=malloc(N*sizeof(double));
ComboM=malloc(N*sizeof(double));
if((M_0== NULL) || (M==NULL) || (ZlnSum==NULL) || (SpSum==NULL) ||
    (ZlnSumOrig==NULL) || (SpSumOrig==NULL) || (ProbVek==NULL) ||
    (ProbExpVek==NULL) || (SQuadratM==NULL) || (CScoreM==NULL) ||
    (CheckerM==NULL) || (ComboM==NULL))
{
    printf("Fehler: malloc schlug in main fehl!\n");

```

```
    return(1);
}
/*
 * Einlesen von M_0
 */
for (i=0; i<AnzZln; i++)
{
    for (j=0; j<AnzSp; j++)
    {
        fscanf(Eingabe, "%d", &M_0[j*AnzZln+i]);
        M[j*AnzZln+i]=0;
    }
}
fclose(Eingabe);
/*
 * Bestimmen der Zeilen- und Spaltensummen ZlnSum und SpSum
 */
for (i=0; i<AnzZln; i++)
{
    ZlnSum[i]=0;
    for (j=0; j<AnzSp; j++)
    {
        ZlnSum[i]=ZlnSum[i]+M_0[j*AnzZln+i];
    }
}
for (j=0; j<AnzSp; j++)
{
    SpSum[j]=0;
    for (i=0; i<AnzZln; i++)
    {
        SpSum[j]=SpSum[j]+M_0[j*AnzZln+i];
    }
}
/*
 * Sortieren der Spalten
 */
for (i=0; i<AnzSp; i++)
{
    k=i;
    for (j=i+1; j<AnzSp; j++)
    {
        if (SpSum[j]>SpSum[k])
        {
            k=j;
        }
    }
}
```

```

Temp=SpSum[i];
SpSum[i]=SpSum[k];
SpSum[k]=Temp;
for (n=0; n<AnzZln; n++)
{
    Temp=M_0[i*AnzZln+n];
    M_0[i*AnzZln+n]=M_0[k*AnzZln+n];
    M_0[k*AnzZln+n]=Temp;
}
SpSumOrig[i]=SpSum[i];
}
/*
* Sortieren der Zeilen
*/
for (i=0; i<AnzZln; i++)
{
    k=i;
    for (j=i+1; j<AnzZln; j++)
    {
        if (ZlnSum[j]>ZlnSum[k])
        {
            k=j;
        }
    }
    Temp=ZlnSum[i];
    ZlnSum[i]=ZlnSum[k];
    ZlnSum[k]=Temp;
    for (n=0; n<AnzSp; n++)
    {
        Temp=M_0[n*AnzZln+i];
        M_0[n*AnzZln+i]=M_0[n*AnzZln+k];
        M_0[n*AnzZln+k]=Temp;
    }
    ZlnSumOrig[i]=ZlnSum[i];
}
/*
* Berechnen der Teststatistiken zu M_0
*/
ComboM_0=Combo(M_0,AnzZln,AnzSp);
CheckerM_0=Checker(M_0,AnzZln,AnzSp);
CScoreM_0=CScore(M_0,AnzZln,AnzSp);
SQuadratM_0=SQuadrat(M_0,AnzZln,AnzSp);
pCScoreCvMin=0;
pCheckerCvMin=0;
pComboCvMin=0;
pSQuadratCvMin=0;

```

```
cv2min=2*N;
for(l=0;l<AnzStich;l++)
{
    printf("\t%d\n", (l+1));
    /*
     * Initialisieren der p-Werte
     */
    pCScore=0;
    pChecker=0;
    pCombo=0;
    pSQuadrat=0;
    SumProb=0;
    ProbExpMin=0;
    for(iter=0;iter<N;iter++)
    {
        Prob=0;
        /*
         * Generierung einer Matrix M
         */
        while ((Prob<=0) || (isnan(Prob)==1) || (isinf(Prob)==1))
        {
            ProbExp=0;
            /*
             * ProbExp dient dem Kuerzen der Wahrscheinlichkeit, um in den
             * Grenzen der Rechengenauigkeit zu bleiben
             */
            Prob=SIS_CB(M,AnzZln,AnzSp,GewExp,ZlnSum,SpSum,&ProbExp);
            for(i=0;i<AnzZln;i++)
            {
                ZlnSum[i]=ZlnSumOrig[i];
            }
            for(i=0;i<AnzSp;i++)
            {
                SpSum[i]=SpSumOrig[i];
            }
        }
        ProbExpVek[iter]=ProbExp;
        /*
         * ProbExpVek sammelt die einzelnen Exponenten, um spaeter die
         * weiteren Berechnungen durchfuehren zu koennen
         */
        if((ProbExpMin==0) || (ProbExp<ProbExpMin))
        {
            ProbExpMin=ProbExp;
        }
    }
    /*
```

```
    * Berechnen der Statistiken zu M
    */
    ComboM[iter]=Combo(M,AnzZln,AnzSp);
    CheckerM[iter]=Checker(M,AnzZln,AnzSp);
    CScoreM[iter]=CScore(M,AnzZln,AnzSp);
    SquadratM[iter]=Squadrat(M,AnzZln,AnzSp);
    ProbVek[iter]=1/Prob;
    /*
    * Neuinitialisieren von M
    */
    for (i=0; i<AnzZln; i++)
    {
        for (j=0; j<AnzSp; j++)
        {
            M[(j*AnzZln)+i]=0;
        }
    }
}
for(iter=0;iter<N;iter++)
{
    if(ProbExpVek[iter]>ProbExpMin)
    {
        ProbExpVek[iter]=ProbExpVek[iter]-ProbExpMin;
        ProbVek[iter]=ProbVek[iter]*pow(10,(-ProbExpVek[iter]));
    }
    SumProb=SumProb+ProbVek[iter];
    /*
    * Berechnen der p-Werte
    */
    if (CScoreM[iter]≥CScoreM_0)
    {
        pCScore=pCScore+ProbVek[iter];
    }
    if (ComboM[iter]≤ComboM_0)
    {
        pCombo=pCombo+ProbVek[iter];
    }
    if (CheckerM[iter]≥CheckerM_0)
    {
        pChecker=pChecker+ProbVek[iter];
    }
    if (SquadratM[iter]≥SquadratM_0)
    {
        pSquadrat=pSquadrat+ProbVek[iter];
    }
}
```

---

```

pChecker=pChecker/SumProb;
pCScore=pCScore/SumProb;
pCombo=pCombo/SumProb;
pSQuadrat=pSQuadrat/SumProb;
/*
 * Berechnen des Chi-Quadrat-Abstandes cv^2
 */
cv2=cv_2(ProbVek,N);
/*
 * Von den AnzStich Durchgaengen wird diejenige Stichprobe verwendet,
 * fuer die cv^2 am geringsten ausfaellt.
 */
if (cv2<cv2min)
{
    cv2min=cv2;
    pCScoreCvMin=pCScore;
    pCheckerCvMin=pChecker;
    pComboCvMin=pCombo;
    pSQuadratCvMin=pSQuadrat;
    /*
     * Berechnen der Schaetzung fuer |Sigma_rc|
     */
    SigmaRC=SumProb/N;
    SigmaRCExp=ProbExpMin;
}
}
while(SigmaRC>10)
{
    SigmaRC=SigmaRC/10;
    SigmaRCExp=SigmaRCExp+1;
}
/*
 * Ausgabe erstellen
 */
Ausgabe=fopen("Ausgabe.txt","a");
fprintf(Ausgabe,"cv^2\t|Sigma_rc|\t|tp-Wert Checker\t|tp-Wert C-score\t");
fprintf(Ausgabe,"p-Wert COMBO\t|tp-Wert S^2\n");
fprintf(Ausgabe,"%g\t|ge+%d\t",cv2min,SigmaRC,SigmaRCExp);
fprintf(Ausgabe,"%g\t|g\t|g\t|g\n",pCheckerCvMin,pCScoreCvMin,
                                pComboCvMin,pSQuadratCvMin);

fclose(Ausgabe);
free(M_0);
free(M);
free(ZlnSum);
free(SpSum);
free(ZlnSumOrig);

```

```

free(SpSumOrig);
free(ProbVek);
free(ProbExpVek);
return(1);
}

static double SIS_CB(int M[],int AnzZln,int AnzSp, double GewExp,
                    int ZlnSum[],int SpSum[],int *pGExp){
/*
 * Generierung einer Matrix mittels SIS
 * @param M Matrix-Rohling, der mit Einsen gefuellert wird
 * @param AnzZln Anzahl der Zeilen
 * @param AnzSp Anzahl der Spalten
 * @param GewExp Vektor der Gewichte fuer die bedingte Bernoulli-
 *               Verteilung
 * @param ZlnSum zu erreichender Zeilensummenvektor
 * @param SpSum zu erreichender Spaltensummenvektor
 * @param *pGExp speichert den zu kuerzenden Exponenten, um in den
 *               Grenzen der Rechengenauigkeit zu bleiben
 */

int l,i,j,k,Count1,Count2,sum_r,sum_SpKonj,low,up,O_i,anz_einsen,
    K_start,Temp,K[AnzZln],V[AnzZln],SpKonj_l[AnzZln],A_n[AnzZln],
    K_Temp,V_Temp;
double Gew[AnzZln],Prob_CB,zufall;

/*
 * Wahrscheinlichkeit fuer Matrix initialisieren
 */
Prob_CB=1;
/*
 * spaltenweise Generierung beginnt
 */
for(l=0;l<AnzSp;l++)
{
    sum_r=ZlnSum[0];
    SpKonj(SpSum,AnzZln,AnzSp,l,SpKonj_l);
    sum_SpKonj=SpKonj_l[0];
    /*
     * Berechnen der Gewichte
     */
    for(i=0;i<AnzZln;i++)
    {
        if(ZlnSum[i]==(AnzSp-1))
        {
            Gew[i]=0;

```

```

}else
{
    double nenner=AnzSp-1-ZlnSum[i];
    double zaehler=ZlnSum[i];
    Gew[i]=pow(exp(log(1*zaehler/nenner)),GewExp);
}
}
/*
 * Suchen von Knoten
 */
Count1=0;
if(sum_r>sum_SpKonj)
{
    K[Count1]=1;
    V[Count1]=sum_r-sum_SpKonj;
    Count1=Count1+1;
}
for(i=1;i<AnzZln;i++)
{
    sum_r=sum_r+ZlnSum[i];
    sum_SpKonj=sum_SpKonj+SpKonj_l[i];
    if(sum_r>sum_SpKonj)
    {
        K[Count1]=i+1;
        V[Count1]=sum_r-sum_SpKonj;
        Count1=Count1+1;
    }
}
/*
 * Entfernen ueberfluessiger Knoten
 */
Temp=0;
if(Count1>1)
{
    for(i=0;i<(Count1-1);i++)
    {
        for(j=(i+1);j<Count1;j++)
        {
            if(V[j]≤V[i])
            {
                K[j]=0;
                Temp=1;
            }else if(V[j]-V[i]≥K[j]-K[i])
            {
                K[i]=0;
                Temp=1;
            }
        }
    }
}

```

```
    }
  }
}
if(Temp==1)
{
  Temp=0;
  Count2=0;
  for(i=0;i<Count1-1;i++)
  {
    if(K[i]==0)
    {
      for(j=(i+1);j<Count1;j++)
      {
        if((K[j]!=0)&&(Temp==0))
        {
          K_Temp=K[i];
          K[i]=K[j];
          K[j]=K_Temp;
          V_Temp=V[i];
          V[i]=V[j];
          V[j]=V_Temp;
          Count2=Count2+1;
          Temp=1;
        }
      }
    } else
    {
      Count2=Count2+1;
    }
    Temp=0;
  }
} else
{
  Count2=Count1;
}
/*
 * Fuellen der l-ten Spalte
 */
O_i=0;
for(i=0;i<Count2;i++)
{
  if(i!=0)
  {
    K_start=K[i-1];
  } else
```

```
{
  K_start=0;
}
/*
 * Bestimmen der einzusetzenden Einsen
 */
low=0;
up=0;
if((V[i]-O_i)>0)
{
  low=V[i]-O_i;
}
if((K[i]-K_start)<(SpSum[l]-O_i))
{
  up=K[i]-K_start;
} else
{
  up=SpSum[l]-O_i;
}
zufall=(double)rand()/((double) RAND_MAX+1);
anz_einsen=low+(int)((up-low+1)*zufall);
Prob_CB=Prob_CB*CB(anz_einsen,K[i],K_start,A_n,Gew,AnzZln);
if(Prob_CB<1e-10)
{
  *pGExp=*pGExp+10;
  Prob_CB=Prob_CB*1e+10;
}
for(j=0;j<anz_einsen;j++)
{
  M[(1*AnzZln)+A_n[j]]=1;
  ZlnSum[A_n[j]]=ZlnSum[A_n[j]]-1;
}
O_i=O_i+anz_einsen;
}
/*
 * Neusortieren der Zeilen der aktualisierten Zeilensummen nach
 */
for (i=0; i<AnzZln; i++)
{
  k=i;
  for (j=i+1; j<AnzZln; j++)
  {
    if (ZlnSum[j]>ZlnSum[k])
    {
      k=j;
    }
  }
}
```

```

    }
    Temp=ZlnSum[i];
    ZlnSum[i]=ZlnSum[k];
    ZlnSum[k]=Temp;
    for (j=0; j<AnzSp; j++)
    {
        Temp=M[(j*AnzZln)+i];
        M[(j*AnzZln)+i]=M[(j*AnzZln)+k];
        M[(j*AnzZln)+k]=Temp;
    }
}
}
return(Prob_CB);
}

static double CB(int Anz,int Ende,int Start,int *A_n,double *Gew,
                int GewDim){
/*
 * Generierung einer Spalte abhaengig von einer Menge Gewichten fuer
 * das SIS-Verfahren
 * @param Anz Anzahl zu setzender Einsen
 * @param Ende Zeilenindex bis zu dem Einsen gesetzt werden sollen
 * @param Start Zeilenindex ab dem Einsen gesetzt werden sollen
 * @param A_n Indikatorstichprobe
 * @param Gew Vektor der Gewichte fuer die bedingte Beroulli Verteilung
 * @param GewDim Laenge des Gewichtsvektors
 */

int i,auswahl,j;
double Prob[GewDim],Prob_neu[GewDim],GewNeu[GewDim],w_sum[GewDim],
        zufall,R_ohne_j,R_mit_j,R_Nenner,Prob_CB,Temp;

for(i=0;i<Anz;i++)
{
    A_n[i]=0; /*in die Stichprobe aufgenommene Elemente*/
}
if(Anz==0)
{
    return(1);
}
else if(Anz==(Ende-Start))
{
    for(i=0;i<Anz;i++)
    {
        A_n[i]=Start+i;
    }
    return(1);
}

```

```

}else
{
  /*
   * Initialisierung von Prob, Prob_neu
   */
  for(i=0;i<GewDim;i++)
  {
    Prob[i]=0;
    Prob_neu[i]=0;
    GewNeu[i]=0;
  }
  /*
   * bestimmen des ersten Elements in A_n
   */
  for(i=Start;i<Ende;i++)
  {
    Temp=Gew[i];
    Gew[i]=0;
    Prob[i]=Temp*R(Anz-1,Gew,Start,Ende);
    /*
     * Es ist  $Prob(i)=Gew(i)*R(n-1,A\{i\})/R(n,A)$ . Der fehlende
     * Faktor  $1/R(n,A)$  haengt nicht von i ab und kann spaeter
     * berechnet werden
     */
    Gew[i]=Temp;
    GewNeu[i]=Gew[i];
  }
  /*
   * zufaelliges Aussuchen eines Elements bedingt durch die
   * jeweiligen Gewichte
   */
  w_sum[Start]=Prob[Start];
  if((Start+1)<Ende)
  {
    for(i=(Start+1);i<Ende;i++)
    {
      w_sum[i]=w_sum[i-1]+Prob[i];
    }
  }
}else
{
  w_sum[Start+1]=w_sum[Start]+Prob[Start+1];
}
zufall=((w_sum[Ende-1])* rand()/(RAND_MAX+1.0));
auswahl=Start;
while(w_sum[auswahl]<zufall)
{

```

```
    auswahl=auswahl+1;
}
A_n[0]=auswahl;
j=0;
GewNeu[auswahl]=0;
/*
 * Bbestimmen der weiteren Stichprobe
 */
while(Anz>(j+1))
{
    /*
     * Aktualisierung der Wahrscheinlichkeiten abhaengig von A_1
     */
    for(i=Start;i<Ende;i++)
    {
        if(GewNeu[i]<1e-16)
        {
            Prob_neu[i]=0;
        }else
        {
            GewNeu[i]=0;
            R_ohne_j=R(Anz-j-2,GewNeu,Start,Ende);
            GewNeu[i]=Gew[i];
            R_mit_j=R(Anz-j-1,GewNeu,Start,Ende);
            if(R_mit_j!=0)
            {
                Prob_neu[i]=Gew[i]*R_ohne_j/((Anz-j-1)*R_mit_j);
            }else
            {
                Prob_neu[i]=1;
            }
        }
    }
    j=j+1;
    /*
     * zufaelliges Aussuchen eines Elements bedingt durch die
     * jeweiligen Gewichte
     */
    w_sum[Start]=Prob_neu[Start];
    if((Start+1)<Ende)
    {
        for(i=(Start+1);i<Ende;i++)
        {
            w_sum[i]=w_sum[i-1]+Prob_neu[i];
        }
    }else
```

```
{
    w_sum[Start+1]=w_sum[Start]+Prob_neu[Start+1];
}
zufall=((w_sum[Ende-1])* rand()/(RAND_MAX+1.0));
auswahl=Start;
while(w_sum[auswahl]<zufall)
{
    auswahl=auswahl+1;
}
A_n[j]=auswahl;
GewNeu[auswahl]=0;
for(i=Start;i<Ende;i++)
{
    Prob[i]=Prob_neu[i];
}
}
/*
 * Aktualisieren der Wahrscheinlichkeit fuer A_j
 */
Prob_CB=1;
for(i=0;i<Anz;i++)
{
    Prob_CB=Prob_CB*Gew[A_n[i]];
}
R_Nenner=R(Anz,Gew,Start,Ende);
if(R_Nenner!=0)
{
    Prob_CB=Prob_CB/R_Nenner;
}
}
return(Prob_CB);
}

static void SpKonj(int SpSum[],int AnzZln,int AnzSp, int l,
                   int SpKonj_l[]){
/*
 * Berechnung eines konjugierten Vektors
 * @param SpSum Spaltensummenvektor
 * @param AnzZln Anzahl der Zeilen
 * @param AnzSp Anzahl der Spalten
 * @param l aktuelle Spalte
 * @param SpKonj_l Vektor der Spaltenkonjugierten der aktuellen Spalte
 */

int i,j,c;
```

```

for(i=0;i<AnzZln;i++)
{
    c=0;
    for(j=(l+1);j<AnzSp;j++)
    {
        if(SpSum[j]≥(i+1))
        {
            c=c+1;
        }
    }
    SpKonj_l[i]=c;
}
}

static double R(int k, double Gew[], int Start, int Ende){
/*
* Berechnung der R-Funktion zu einem k und einer Menge von Gewichten
* @param k Parameter k der R-Funktionen
* @param Gew Vektor der Gewichte
* @param Start Zeilenindex von dem an die R-Funktion berechnet wird
* @param Ende Zeilenindex bis zu dem die R-Funktion berechnet wird
*/

if(k==0)
{
    return(1);
}else if(k>(Ende-Start))
{
    return(0);
}else if(k>15)
{
    int i,j,N,Count;
    double *r,r_Temp;
    N=(Ende-Start);
    Count=0;
    r=malloc(((N+1)*(k+1))*sizeof(double));
    r[0]=1;
    for(j=0;j≤N;j++)
    {
        r[j*(k+1)]=1;
    }
    for(j=1;j≤N;j++)
    {
        for(i=j;i≥1;i--)
        {
            if(i≤k)

```

```

    {
        if (i==j)
        {
            r[j*(k+1)+i]=(Gew[Start+j-1]*r[(k+1)*(j-1)+(i-1)]);
        } else
        {
            r[j*(k+1)+i]=r[(k+1)*(j-1)+i]+
                (Gew[Start+j-1]*r[(k+1)*(j-1)+(i-1)]);
        }
    }
}
r_Temp=r[N*(k+1)+k];
free(r);
return(r_Temp);
} else
{
    int i, j;
    double r_Temp;
    long double r[k+1],p[k+1];
    for (i=1; i≤k; i++)
    {
        p[i]=P(i, Gew, Start, Ende);
    }
    r[0]=1;
    for (j=1; j≤k; j++)
    {
        r[j]=0;
        for (i=1; i≤j; i++)
        {
            if ((i%2)==0)
            {
                r[j]=r[j]-(p[i]*r[j-i]);
            }
            else
            {
                r[j]=r[j]+(p[i]*r[j-i]);
            }
        }
        r[j]=r[j]/j;
    }
    r_Temp=r[k];
    if (r_Temp<0)r_Temp=-1*r_Temp;
    return (r_Temp);
}
}

```

```
static long double P(int k, double Gew[], int Start, int Ende){
/*
 * Berechnung der Potenzsummen fuer die Berechnung der R-Funktionen
 * @param k Parameter k der Potenzsummen
 * @param Gew Vektor der Gewichte
 * @param Start Zeilenindex von dem an die Potenzsummen berechnet
 *           werden
 * @param Ende Zeilenindex bis zu dem die Potenzsummen berechnet werden
 */
long double erg,Temp;
int j, i;
erg=0;
for (j=Start; j<Ende; j++)
{
    Temp=1;
    for (i=0; i<k; i++)
    {
        Temp=Temp*Gew[j];
    }
    erg=erg+Temp;
}
return (erg);
}

static double cv_2(double ProbVek[],int n){
/*
 * Schaetzung des Chi-Quadrat-Abstandes der Probeverteilung zur
 * Zielverteilung (Gleichverteilung) des SIS-Verfahrens
 * @param ProbVek Wahrscheinlichkeiten der Stichprobenelemente
 * @param n Stichprobenumfang
 */
int i,n_korr;
double SumProbVek,cv2,ProbVekTemp,ProbVekExpMin,ProbVekExp;
n_korr=n;
SumProbVek=0;
ProbVekExpMin=0;
ProbVekExp=0;
for(i=0;i<n;i++)
{
    ProbVekTemp=ProbVek[i];
    if(ProbVek[i]==0)
    {
        n_korr=n_korr-1;
    }else
    {
```

```
while(ProbVekTemp>10)
{
    ProbVekExp=ProbVekExp+1;
    ProbVekTemp=ProbVekTemp/10;
}
if((ProbVekExp<ProbVekExpMin) || (ProbVekExpMin==0))
{
    ProbVekExpMin=ProbVekExp;
}
}
}
for(i=0;i<n;i++)
{
    ProbVek[i]=ProbVek[i]/pow(10,ProbVekExpMin);
    SumProbVek=SumProbVek+ProbVek[i];
}
SumProbVek=SumProbVek/n_korr;
cv2=0;
for(i=0;i<n;i++)
{
    if(ProbVek[i]!=0)
    {
        cv2=cv2+((SumProbVek-ProbVek[i])*(SumProbVek-ProbVek[i]));
    }
}
cv2=cv2/(n_korr-1)/SumProbVek/SumProbVek;
return(cv2);
}

static int Checker(int *M,int AnzZln,int AnzSp)
{
    /*
    * Berechnung der CHECKER Teststatistik zu einer Binaermatrix
    * @param M Binaermatrix zu der CHECKER berechnet werden soll
    * @param AnzZln Anzahl der Zeilen von M
    * @param AnzSp Anzahl der Spalten von M
    */
    int i,j,k,l,Abbruch,check,Kontroll;
    check=0;
    Kontroll=0;
    for (i=0;i<(AnzZln-1);i++)
    {
        for (j=(i+1);j<AnzZln;j++)
        {
            Abbruch=0;
            k=0;
```

---

```

while ((Abbruch==0)&&(k<(AnzSp-1)))
{
    if (M[k*AnzZln+i]!=M[k*AnzZln+j])
    {
        l=k+1;
        while ((Abbruch==0)&&(l<AnzSp))
        {
            if ((M[k*AnzZln+i]==M[l*AnzZln+j])
                &&(M[k*AnzZln+j]==M[l*AnzZln+i]))
            {
                check=check+1;
                Abbruch=1;
                Kontroll=1;
            }
            l=l+1;
        }
        k=k+1;
    }
    if (Kontroll==1)
    {
        k=0;
        while ((k<AnzSp)&&(Kontroll==1))
        {
            if ((M[k*AnzZln+i]==1)&&(M[k*AnzZln+j]==1))
            {
                check=check-1;
                Kontroll=0;
            }
            k=k+1;
        }
        Kontroll=0;
    }
}
return(check);
}

static int Combo(int *M,int AnzZln,int AnzSp)
{
    /*
    * Berechnung der COMBO Teststatistik zu einer Binaermatrix
    * @param M Binaermatrix zu der COMBO berechnet werden soll
    * @param AnzZln Anzahl der Zeilen von M
    * @param AnzSp Anzahl der Spalten von M
    */

```

```
int i, j, k, Count, Temp, comb;
comb=0;
for (i=0; i<(AnzSp-1); i++)
{
    Temp=1;
    for (j=(i+1); j<AnzSp; j++)
    {
        Count=0;
        for (k=0; k<AnzZln; k++)
        {
            if (M[(i*AnzZln)+k]==M[(j*AnzZln)+k])
            {
                Count=Count+1;
            }
        }
        if (Count==AnzZln)
        {
            Temp=0;
        }
    }
    if (Temp==1)
    {
        comb=comb+1;
    }
}
comb=comb+1;
return(comb);
}

static double CScore(int *M, int AnzZln, int AnzSp)
{
    /*
    * Berechnung der C-score Teststatistik zu einer Binaermatrix
    * @param M Binaermatrix zu der C-score berechnet werden soll
    * @param AnzZln Anzahl der Zeilen von M
    * @param AnzSp Anzahl der Spalten von M
    */
    int i, j, k, *S2, *Q2;
    double cscore;
    S2=malloc(AnzZln*sizeof(int));
    Q2=malloc((AnzZln*AnzZln)*sizeof(int*));
    if ((Q2==NULL) || (S2==NULL))
    {
        printf("Error:malloc schlug in der Funktion CScore fehl\n");
        return(1);
    }
}
```

```
cscore=0.0;
for (i=0;i<AnzZln;i++)
{
  for (j=0;j<AnzZln;j++)
  {
    Q2[(j*AnzZln)+i]=0;
  }
}
for (i=0;i<AnzZln;i++)
{
  S2[i]=0;
  for (j=0;j<AnzSp;j++)
  {
    S2[i]=S2[i]+M[(j*AnzZln)+i];
  }
  for (j=0;j<AnzSp;j++)
  {
    if (M[(j*AnzZln)+i]==1)
    {
      for (k=0;k<AnzZln;k++)
      {
        if (M[(j*AnzZln)+k]==1)
        {
          Q2[(k*AnzZln)+i]=Q2[(k*AnzZln)+i]+1;
        }
      }
    }
  }
}
for (i=0;i<(AnzZln-1);i++)
{
  for (j=(i+1);j<AnzZln;j++)
  {
    cscore=cscore+((double)(S2[i]-Q2[(j*AnzZln)+i])
                  *(double)(S2[j]-Q2[(j*AnzZln)+i]));
  }
}
cscore=cscore/(AnzZln*(AnzZln-1)/2);
free(S2);
free(Q2);
return(cscore);
}

static double Squadrat(int *M,int AnzZln,int AnzSp)
{
  /*
```

```
* Berechnung der S^2 Teststatistik zu einer Binaermatrix
* @param M Binaermatrix zu der S^2 berechnet werden soll
* @param AnzZln Anzahl der Zeilen von M
* @param AnzSp Anzahl der Spalten von M
*/
int i, j, k;
double Sum, Temp;
Sum=0;
for (i=0; i<AnzZln; i++)
{
    for (j=0; j<AnzZln; j++)
    {
        Temp=0;
        for (k=0; k<AnzSp; k++)
        {
            Temp=Temp+(M[(k*AnzZln)+i]*M[(k*AnzZln)+j]);
        }
        if (i!=j)
        {
            Sum=Sum+(Temp*Temp);
        }
    }
}
return(Sum/(AnzZln*(AnzZln-1)));
}
```

# Literaturverzeichnis

- [1] AICS RESEARCH, Inc.: *Nestedness Temperature Calculator Program*.  
<http://aics-research.com/nestedness/tempcalc.html>
- [2] CHEN, Sean X.: Weighted polynomial models and weighted sampling schemes for finite population. In: *The Annals of Statistics* 26 (1998), Nr. 5, S. 1894–1915
- [3] CHEN, Sean X.: General Properties and Estimation of Conditional Bernoulli Models. In: *Journal of Multivariate Analysis* (2000), S. 69–87
- [4] CHEN, Sean X.; LIU, Jun S.: Statistical Applications of the Poisson-Binomial and Conditional Bernoulli Distributions. In: *Statistica Sinica* (1997), S. 875–892
- [5] CHEN, Xiang-Hui; DEMPSTER, Arthur P.; LIU, Jun S.: Weighted finite population sampling to maximize entropy. In: *Biometrika* (1994), S. 457–469
- [6] CHEN, Yuguo: Another look at rejection sampling through importance sampling. In: *Statistics & Probability Letters* 72 (2005), May, Nr. 4, S. 277–283
- [7] CHEN, Yuguo; DIACONIS, Persi; HOLMES, Susan P.; LIU, Jun S.: Sequential Monte Carlo Methods for Statistical Analysis of Tables. In: *Journal of the American Statistical Association* (2005), S. 109–120
- [8] CRAMER, Erhard; KAMPS, Udo: *Grundlagen der Wahrscheinlichkeitsrechnung und Statistik*. Springer-Verlag, 2007
- [9] DEHLING, Herold; HAUPT, Beate: *Einführung in die Wahrscheinlichkeitstheorie und Statistik*. 2. Auflage. Springer, 2004
- [10] DIAMOND, Jared: Assembly of species communities. In: CODY, Martin (Hrsg.); DIAMOND, Jared (Hrsg.): *Ecology and evolution of communities*. dritter Druck 1979. Harvard University Press, Cambridge, 1975, S. 342–444
- [11] FALK, Michael; BECKER, Rainer; MAROHN, Frank: *Angewandte Statistik. Eine Einführung mit Programmbeispielen in SAS*. Springer, 2004

- [12] FEELEY, Kenneth: Analysis of avian communities in Lake Guri, Venezuela, using multiple assembly rule models. In: *Oecologia* 137 (2003), Nr. 1, S. 104–113
- [13] GAIL, Mitchell H.; LUBIN, Jay H.; RUBINSTEIN, Lawrence V.: Likelihood Calculations for Matched Case-Control Studies and Survival Studies with Tied Death Times. In: *Biometrika* 68 (1981), Nr. 3, S. 703–707
- [14] GEORGIU, Hans-Otto: *Stochastik*. 2., bearbeitete Auflage. de Gruyter, 2004
- [15] GOTELLI, Nicholas J.: Null Model Analysis of Species Co-Occurrence Patterns. In: *Ecology* 81 (2000), Nr. 9, S. 2606–2621
- [16] JÖCKEL, Karl-Heinz: Finite Sample Properties and Asymptotic Efficiency of Monte Carlo Tests. In: *The Annals of Statistics* 14 (1986), Nr. 1, S. 336–347
- [17] KONG, A.; LIU, J.S.; WONG, W.H.: Sequential Imputations and Bayesian Missing Data Problems. In: *Journal of the American Statistical Association* 89 (1994), S. 278–288
- [18] KRENGEL, Ulrich: *Einführung in die Wahrscheinlichkeitstheorie und Statistik*. 7. Auflage. Vieweg Verlag, 2003
- [19] KULLBACK, Solomon; LEIBLER, Richard A.: On Information and Sufficiency. In: *The Annals of Mathematical Statistics* 22 (1951), Nr. 1, S. 79–86
- [20] LIU, Jun S.: *Monte Carlo Strategies in Scientific Computing*. Springer, 2001
- [21] LIU, Jun S.; CHEN, Rong: Sequential Monte Carlo Methods for Dynamic Systems. In: *Journal of the American Statistical Association* 93 (1998), S. 1032–1044
- [22] MALY, Edward J.; DOOLITTLE, William L.: Effects of Island Area and Habitat on Bahamian Land and Freshwater Snail Distribution. In: *American Midland Naturalist* 97 (1977), Nr. 1, S. 59–67
- [23] MEAD, D. G.: Newton’s Identities. In: *The American Mathematical Monthly* 99 (1992), Oktober, Nr. 8, S. 749–751
- [24] PÉREZ-SALVADOR, Blanca R.; COBOS-SILVA, Sergio de-los; GUTIÉRREZ-ÁNDRADE, Miguel A.; TORRES-CHAZARO, Adolfo: A reduced formula for the precise number of  $(0, 1)$ -matrices in  $A(R, S)$ . In: *Discrete Math.* 256 (2002), Nr. 1-2

- [25] PIELOU, D. P.; PIELOU, E. C.: Association among species of infrequent occurrence: The insect and spider fauna of *Polyporus betulinus* (Bulliard) Fries. In: *Journal of Theoretical Biology* 21 (1968), Nr. 2, S. 202–216
- [26] POKROPP: *Stichproben: Theorie und Verfahren*. Oldenbourg, 1996
- [27] ROBERTS, Alan; STONE, Lewi: Island-sharing by archipelago species. In: *Oecologia* 83 (1990), S. 560–567
- [28] RYSER, Herbert J.: *Combinatorial Mathematics*. The Mathematical Association of America, 1963
- [29] RYSER, H.J.: Combinatorial properties of matrices of zeros and ones. In: *Can. J. Math.* 9 (1957), S. 371–377
- [30] STERN, Hal; COVER, Thomas M.: Maximum Entropy and the Lottery. In: *Journal of the American Statistical Association* 84 (1989), Dezember, Nr. 408, S. 980–985
- [31] STONE, Lewi; ROBERTS, Alan: The checkerboard score and species distributions. In: *Oecologia* 85 (1990), S. 74–79
- [32] VIERTL, Reinhard: *Einführung in die Stochastik*. 3., überarbeitete und erweiterte Auflage. Springer-Verlag Wien, 2003

Hiermit versichere ich, dass ich diese Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Außerdem versichere ich, dass ich die allgemeinen Prinzipien wissenschaftlicher Arbeit und Veröffentlichung, wie sie in den Leitlinien guter wissenschaftlicher Praxis der Carl von Ossietzky Universität Oldenburg festgelegt sind, befolgt habe.