

Diplom-Arbeit

# Lernen in Bayes'schen Netzen

von: Patrick Rammelt  
Berichter: Prof. Dr. Ulrich Kockelkorn  
Prof. Dr. Gerhard Tutz  
Betreuer: Dr. Göran Kauermann

Ich versichere hiermit, daß ich die vorliegende Diplomarbeit selbstständig verfaßt und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Berlin den 25.11.1998

# Inhaltsverzeichnis

<b>1</b>	<b>Begriffe, Notationen und Rechenregeln</b>	<b>6</b>
1.1	Einflußgrößen, Merkmale, (Zufalls-)Variablen, Knoten und Dimensionen . . . . .	6
1.2	A-priori-, A-posteriori-, Gemeinsame und Bedingte Verteilungen . . . . .	6
1.3	“Bayesianer” und “Frequentisten” bzw. Bayes’sche- und Klassische Statistik . . . . .	7
1.4	Mehrdimensionale Tafeln . . . . .	7
1.4.1	Zuordnung von Merkmalen zu Dimensionen . . . . .	7
1.4.2	Multiplikation von Tafeln . . . . .	7
1.4.3	Marginalisierung . . . . .	8
1.4.4	Bedingte Tafeln / Verteilungen . . . . .	8
1.5	Unabhängigkeit und bedingte Unabhängigkeit . . . . .	9
1.6	Datensatz und Kontingenztafeln . . . . .	9
1.7	Normierung . . . . .	10
<b>2</b>	<b>Bayes’sche Netze</b>	<b>11</b>
2.1	Was ist ein Bayes’sches Netz? . . . . .	11
2.2	Bestandteile eines Bayes’schen Netzes . . . . .	11
2.3	(Un-) Abhängigkeitsstruktur und d-separation . . . . .	12
2.3.1	Der Moralische Graph . . . . .	13
2.4	Beschränkung der dargestellten Abhängigkeiten durch die Netzstruktur . . . . .	13
2.5	Nutzung Bayes’scher Netze . . . . .	13
2.6	Stärken Bayes’scher Netze . . . . .	13
<b>3</b>	<b>Inferenz - Junction-Tree</b>	<b>14</b>
3.1	Bedeutung des Problems . . . . .	14
3.2	Grundgedanke . . . . .	14
3.3	Cluster-Tree / Junction-Tree . . . . .	15
3.4	Inferenzberechnung mit dem Junction-Tree . . . . .	17
3.5	Konsequenzen . . . . .	18
<b>4</b>	<b>Grundlagen und Zielsetzung</b>	<b>19</b>
4.1	Bayes’sche Netze als Modellannahme . . . . .	19
4.2	Parameter und Dimension eines Bayes’schen Netzes . . . . .	19
4.3	Ziel . . . . .	20
4.4	Grundlage der Schätzung . . . . .	20
4.4.1	Der Datensatz . . . . .	20
4.5	Bayes’sche Statistik . . . . .	21
<b>5</b>	<b>Parameterschätzung</b>	<b>21</b>
5.1	Grundlagen der Parameterschätzung . . . . .	21
5.2	Einstweilige Vereinfachung des Problems . . . . .	22
5.3	Die Beta-Verteilung . . . . .	22
5.4	Gleichartige Betrachtung von Vorwissen und Daten . . . . .	23
5.4.1	Die $\Gamma$ -Funktion . . . . .	24
5.5	Die Normierungskonstante der Beta-Verteilung . . . . .	24
5.6	Konjugierte Verteilung . . . . .	25
5.7	Bestimmung der <i>marginalen Likelihood</i> . . . . .	25
5.8	Schätzung der Parameter . . . . .	25
5.8.1	Vergleich des Bayes’schen und des Klassischen Ansatzes . . . . .	26
5.8.2	Allgemeine Form der Bayes’schen Parameterschätzung . . . . .	26
5.8.3	Schätzung unter der Annahme einer Beta-Verteilung . . . . .	26
5.8.4	Erkenntnisse . . . . .	26
5.8.5	Parameterschätzung in Bayes’schen Netzen (volle Modellannahme) . . . . .	27

<b>6</b>	<b>Netzstrukturbewertung</b>	<b>27</b>
6.1	Die Dirichlet-Verteilung (multinomiale Variablen)	27
6.2	Die Normierungskonstante der Dirichlet-Verteilung	28
6.3	Der Multivariate Fall (volle Modellannahme)	28
6.4	Allgemeine Form der Netzbewertung	29
6.5	Herleitung der Netzbewertungsformel	29
6.6	Bewertung der Netzstruktur	30
6.7	Bewertung von Abhängigkeiten	30
6.8	Rücksicht auf die Parameterschätzung Und Vermeidung von <i>overfitting</i>	33
6.9	Gewinnung der $\alpha$ - (Vorwissen-) Tafeln	33
<b>7</b>	<b>Netzstruktursuche</b>	<b>35</b>
7.1	Schrittweise Verbesserung der Netzstruktur ( "Greedy-Search" )	35
7.2	Effiziente Berechnung	37
7.3	Veränderte Kontingenz- und Vorwissen-Tafeln	38
7.3.1	Entfernen einer Kante	40
7.3.2	Hinzufügen einer Kante	40
7.3.3	Vereinfachtes Hinzufügen einer Kante	42
7.4	Fortführung des Verfahrens über mehrere Veränderungsschritte	43
7.4.1	Umdrehen von Kanten	43
7.4.2	Lokale Maximierung	44
<b>8</b>	<b>Umgang mit fehlendem Vorwissen</b>	<b>44</b>
8.1	Fehlendes Vorwissen über die Netzstruktur	44
8.2	Vorwissen durch "Bayes'sches Unwissen" ersetzen	44
8.2.1	Wahl der <i>UserSampleSize</i>	44
8.2.2	Beispielrechnung zur Wahl der <i>UserSampleSize</i>	45
8.2.3	Mathematische Erklärung des beobachteten Effekts	45
8.2.4	Die " $\alpha + 1$ "-Methode	47
8.3	Gewinnung des Vorwissens aus den Daten	49
8.3.1	Vorwissen als Korrektiv zu den Daten	49
8.3.2	"Doppelte" Verwendung der Daten	49
8.3.3	Vermeidung von Nullzellen	50
8.3.4	Erstellung der "Vorwissen"-Tafeln bei Netzveränderungen	50
8.3.5	Wahl des Teilungsverhältnisses der Daten	51
8.4	(Teil-) Vorwissen über die Netzstruktur	52
8.5	Einbeziehung der Netz-a-priori-Wahrscheinlichkeit	52
<b>9</b>	<b>Nachbetrachtung</b>	<b>52</b>
9.1	Uneinheitliche <i>UserSampleSize</i>	52
<b>10</b>	<b>Simulationen und Testberechnungen</b>	<b>53</b>
10.1	Testverfahren	53
10.2	Berechnungen mit "perfekten" Daten	54
10.2.1	Gewinnung "perfekter" Daten	54
10.2.2	Versuch 1: "Vorwissen" aus Daten (relativ geringer Anteil)	55
10.2.3	Versuch 2: "Vorwissen" aus Daten (erhöhter Anteil)	57
10.2.4	Versuch 3: "Vorwissen" aus Daten (minimaler Anteil)	59
10.2.5	Versuch 4: "Vorwissen" aus Daten (Verkleinerung des Datensatzes)	59
10.2.6	Versuch 5: "Vorwissen" aus Daten (Weitere Verkleinerung des Datensatzes)	60
10.2.7	Versuch 6: Mit Bayes'schem "Unwissen" (geringes Gewicht)	61
10.2.8	Versuch 7: Mit Bayes'schem "Unwissen" (erhöhtes Gewicht)	62
10.3	Berechnungen mit dem Kredit-Datensatz	64
10.3.1	Der Kredit-Datensatz	64
10.3.2	Versuch 1: "Vorwissen" aus Daten (geringer Anteil)	64
10.3.3	Versuch 2: "Vorwissen" aus Daten (erhöhter Anteil)	66
10.3.4	Vergleich von Versuch 1 und Versuch 2	67
10.3.5	Versuch 3: Mit Bayes'schem "Unwissen" (geringes Gewicht)	69

10.3.6	Versuch 4: Mit Bayes'schem "Unwissen" (minimales Gewicht)	70
10.3.7	Vergleich von Versuch 3 und Versuch 4	71
10.3.8	Versuch 5: Mit Bayes'schem "Unwissen" (rel. hohes Gewicht)	73
10.4	Daten-Sampling	74
10.4.1	Verfahren zum Daten-Sampling	74
10.4.2	Versuch 1: $B \rightarrow A \rightarrow C$	74
10.4.3	Versuch 2: $B \rightarrow A \quad C$	76
10.4.4	Versuch 3: $B \rightarrow A \leftarrow C$	76
10.4.5	Vergleich der beiden Testvarianten	77
<b>11</b>	<b>Implementierung und Technisches</b>	<b>78</b>
11.1	Vorraussetzungen und Grundlagen	78
11.1.1	Implementierungssprache(n)	78
11.1.2	S-Plus	80
11.1.3	Festlegung der Speicherklassen in S-Plus	80
11.1.4	Aufspaltung von (gemischten) Objekten	80
11.1.5	Erstellung der Objekte in zwei Stufen	81
11.1.6	Laden von C/C++-Modulen	81
11.1.7	Nutzung der Vorteile von C++ gegenüber C	81
11.1.8	Objektübergabe / Objekterzeugung (S-Plus/C++)	82
11.1.9	Der "Zwei-Stufen-Fall"	83
11.1.10	Listen Komplexer Objekte	84
11.2	Repräsentation der Objekte	85
11.2.1	Die Wahrscheinlichkeitstablen	85
11.2.2	Der Datensatz	85
11.2.3	Der DAG	87
11.3	(Daten-) Struktur des Lernalgorithmus'	87
11.3.1	Mit Vorwissen	89
11.3.2	Gruppierte Daten	89
11.3.3	Ohne Vorwissen	89
11.3.4	Der Grundbewertungs-Vektor	90
11.3.5	Die Änderungsbewertungs-Matrix	90
11.4	Struktur des Lernalgorithmus' ( Überblick )	91
11.5	Rechnen mit logarithmierten Werten	91
<b>12</b>	<b>Anhang</b>	<b>93</b>
12.1	Mehrdimensionale Tafeln ( formale Definitionen )	93
12.1.1	Größe Mehrdimensionaler Tafeln	93
12.1.2	Vektoren "in den ( mehrdimensionalen ) Raum legen"	93
12.1.3	Größe resultierender Tafeln	93
12.1.4	Zugriff auf einzelne Zellen in mehrdimensionalen Tafeln	94
12.1.5	Rechenoperationen auf mehrdimensionalen Tafeln	94
12.1.6	Rechnen mit mehrdimensionalen Tafeln und Einzelwerten	94
12.1.7	Marginale Tafeln / Marginalisierung	94
12.1.8	Gewinnung einzelner Zellen einer Kontingenztafel	94
12.1.9	Darstellungsformen verschiedener Verteilungen	95
12.2	Detaillierte Berechnung zur Wahl der <i>UserSampleSize</i>	95
12.3	Detaillierte Berechnung zum Beispiel zur " $\alpha + 1$ -Methode"	97
12.4	Ausgabe des Simulations-Tools in Kapitel 10	98
12.4.1	DAG in Matrix-Form	98
12.4.2	Die bedingten Wahrscheinlichkeiten zum DAG	98
12.4.3	Die "Unwissen"-Tafeln	99
12.5	Konventionen	100
Literatur		101

# 1 Begriffe, Notationen und Rechenregeln

## 1.1 Einflußgrößen, Merkmale, (Zufalls-)Variablen, Knoten und Dimensionen

Die Begriffe “**Einflußgröße**”, “**Merkmal**”, “**(Zufalls-)Variable**”, “**Knoten**” und im weiteren Sinne auch “**Dimension**” sind äquivalent. Jedesmal ist ein für die Beschreibung eines fraglichen Sachverhalts relevanter Oberbegriff gemeint: z.B. “Alter”, “Körpergröße”, “Beruf”, “Vermögen” oder “Haarfarbe” zur allgemeinen Beschreibung einer Person. Solche Merkmale werden mit Dimensionen in mehrdimensionalen Tafeln korrespondieren. Bei uns wird diese Zuordnung fest sein, so daß in jeder beliebigen Tafel die  $i$ -te Dimension dem  $i$ -ten Merkmal entspricht (selbst wenn dieses Merkmal in einer speziellen Tafel nicht repräsentiert werden soll). Wird man zufällig ausgewählte Personen zu den oben genannten Punkten befragen, so wird man zufällige Ausprägungskombinationen zu den Merkmalen erhalten. Mit (möglichen) **Ausprägungen** bezeichnen wir die konkreten Werte, die ein Merkmal annehmen kann, z.B. Alter = 24 *Jahre*, Größe = 183 *cm* oder Beruf = “Informatiker”. Die Ausprägungen solcher Merkmale können **stetig** oder **diskret** sein. Dabei können Alter und Größe theoretisch beliebig genau gemessen werden - wären damit stetig. Andere Merkmale wie Beruf sind immer diskret, da sie nur mit verschiedenen **Kategorien** ( “Informatiker”, “Arzt”, “Kameltreiber”,... ) zu beschreiben sind. Potentiell stetige Merkmale können aber ebenfalls in diskreten Kategorien gemessen werden, oder in solche überführt werden, z.B. Alter = 0 – 10, 11 – 14, 15 – 18, 21 – 30,... *Jahre*. Die Wahl der Kategorien ist nicht unproblematisch. Gibt es etwa in Bezug auf die Fragestellung einen entscheidenden “Cut”, z.B. zwischen Personen unter bzw. über 25 Jahren, so könnte dieser Umstand durch die oben gewählte Einteilung (21 – 30 Jahre) verwischt werden. Die Wahl der Kategorien ist aber als allen hier vorgestellten Verfahren vorangestellte Festlegung, die i.d.R. von einem Experten getroffen wird, nicht mehr im nachhinein automatisch beeinflussbar. Wir werden hier ausschließlich mit diskreten bzw. diskretisierten Merkmalen arbeiten, deren Kategorien **umfassend** und sich gegenseitig **ausschließend** gewählt sind. D.h. jedes betrachtete Objekt bzw. Individuum fällt in **genau eine** Kategorie **jedes** Merkmals.

## 1.2 A-priori-, A-posteriori-, Gemeinsame und Bedingte Verteilungen

(Wahrscheinlichkeits-) Verteilungen **einzelner** Merkmale geben an, mit welcher Wahrscheinlichkeit beispielsweise eine zufällig ausgewählte Person die Haarfarbe blond (*Haarfarbe* = *blond*) aufweist. Eine solche Verteilung nennen wir **a-priori-Verteilung**, weil sie die Vorab-Verteilung eines einzelnen Merkmals ungeachtet aller möglichen anderen Einflußgrößen darstellt. Eine andere Verteilungsart für einzelne Merkmale ist die **a-posteriori-Verteilung**. Sie gibt z.B. an, mit welcher Wahrscheinlichkeit eine Person blond ist, nachdem die Information bekannt geworden ist, daß die Person Schwede ist, jedoch ohne diese Verteilung zusätzlich auch nach der Nation (allen Nationen) aufzuschlüsseln (vgl. bedingte Wahrscheinlichkeiten).

Neben solchen einfachen Verteilungen können auch die **gemeinsamen Verteilungen** mehrerer Merkmale betrachtet werden. Mit welcher Wahrscheinlichkeit ist etwa eine zufällig ausgewählte Person blond **und** Schwede? Die a-priori-Verteilungen einzelner Merkmale können aus der gemeinsamen Verteilung der Merkmale abgeleitet werden (siehe Abschnitt 1.4.3) der Umkehrschluß gilt dagegen im allgemeinen nicht (vgl.  $p(A, B)$  und  $p'(A, B)$  aus Abbildung 41 3) und 4) sowie Abschnitt 1.5).

Neben gemeinsamen Verteilungen können auch **bedingte Verteilungen** (bedingte Wahrscheinlichkeiten) betrachtet werden. Wenn schon bekannt ist, daß eine Person Schwede ist, mit welcher Wahrscheinlichkeit ist sie dann auch blond? Im Gegensatz zur a-posteriori-Verteilung wird hier nach allen betrachteten Nationen aufgeschlüsselt, so daß diese Tafel eher eine Betrachtung der Eventualitäten enthält (“Wenn bekannt wäre, daß ... dann ...”), wohingegen in der a-posteriori-Verteilung alle anderen Möglichkeiten bereits ausgeschlossen sind und für die weitere Betrachtung fallen gelassen werden. Hier beträgt die Wahrscheinlichkeit, daß ein **Schwede** blond oder schwarzhaarig ist oder sonsteiner Kategorie von Haarfarbe angehört 1, denn es wird nur noch die Gruppe der Schweden betrachtet und jeder Schwede gehört (!) einer Kategorie des (umfassenden) Merkmals *Haarfarbe* an.

$$\sum_{\text{Haarfarbe} \in \{\text{blond, schwarz, ...}\}} p(\text{Haarfarbe} | \text{Nationalität} = \text{Schwede}) = 1,$$

Demgegenüber ist die Summe der Wahrscheinlichkeiten aller Haarfarben aus der gemeinsamen Wahrscheinlichkeitstafel, dafür daß **jemand** Schwede ist:

$$\sum_{\text{Haarfarbe} \in \{\text{blond, schwarz, ...}\}} p(\text{Haarfarbe}, \text{Nationalität} = \text{Schwede}) = ?$$

noch davon abhängig welcher Anteil der Weltbevölkerung überhaupt **a-priori** Schweden sind:

$$p(\text{Nationalität} = \text{Schwede}) = \sum_{\text{Haarfarbe} \in \{\text{blond}, \text{schwarz}, \dots\}} p(\text{Haarfarbe}, \text{Nationalität} = \text{Schwede})$$

Auch bedingte Verteilungen können aus gemeinsamen Wahrscheinlichkeitsverteilungen abgeleitet werden, wobei Information verloren geht (siehe Abschnitt 1.4.4).

### 1.3 “Bayesianer” und “Frequentisten” bzw. Bayes’sche- und Klassische Statistik

Eine Aufgabe und auch ein Problem der Statistik ist es Verteilungen aus der Beobachtung (möglichst vieler) Personen bzw. dem jeweiligen Gegenstand der Untersuchung zu schätzen.

Bei der Interpretation der geschätzten Wahrscheinlichkeiten und den daraus weiter ableitbaren Verwendungsmöglichkeiten bestehen unterschiedliche Auffassungen zwischen den Anhängern der klassischen Statistik (“Frequentisten”) und den sogenannten “Bayesianern” (Anhänger der Bayes’schen Statistik). Wagt es der Frequentist nur Aussagen über die Verteilungen in potentiell unendlichen Mengen zu treffen, so ist es für den Bayesianer möglich aus einer Wahrscheinlichkeitsverteilung auch Entscheidungen in einem konkreten Fall abzuleiten. Liegt etwa das Regenrisiko laut Wetterbericht an einem bestimmten Tag bei 80 Prozent, so wird der Bayesianer wohl seinen Schirm mitnehmen, wohingegen der Frequentist sagt “Soetwas wie ein Regenrisiko an einem bestimmten Tag kann es nicht geben, denn entweder es regnet an diesem Tag oder es regnet nicht”. Möglich ist aus Sicht des Frequentisten nur die Angabe einer Regenwahrscheinlichkeit für (alle) Tage, mit den gleichen Ausgangsparametern (Wind, Luftdruck, etc.). Der Bayes’sche Ansatz hingegen läßt sich oft mit der Frage charakterisieren: “Wäre ich (in diesem konkreten Fall) bereit eine Wette einzugehen und welchen Einsatz wäre es mir wert?” Spätestens unter diesem Gesichtspunkt wird wohl jedem, der sich nicht ausdrücklich den Frequentisten zuordnet der Sinn des Bayes’schen Wahrscheinlichkeitsverständnisses klar. Wir werden uns durchgehend der Bayes’schen Statistik verschreiben, schon deshalb weil es uns sonst unmöglich wäre aus unseren Ergebnissen sinnvolle Konsequenzen zu ziehen.

### 1.4 Mehrdimensionale Tafeln

Da wir mit diskreten Merkmalen arbeiten, lassen sich Informationen über die Verteilungen (gleich welcher Art) durch **mehrdimensionale Tafeln** darstellen. Dabei entspricht jede Dimension **jeder** Tafel einem festzugeordneten Merkmal. Jede Zelle einer Tafel entspricht dabei einer Ausprägungskombination der enthaltenen Merkmale. Der Umgang mit mehrdimensionalen Tafeln wird im Folgenden anhand von einfachen Beispielen gezeigt.

#### 1.4.1 Zuordnung von Merkmalen zu Dimensionen

Wir werden von einer festen Zuordnung von Dimensionen und Merkmalen ausgehen. D.h. das  $i$ -te Merkmal wird immer der  $i$ -ten Dimension in jeder beliebigen Tafel entsprechen.

$p(A)$			$p(B)$		$p(A, B)$			
$A$					$A$			
$a_1$	$a_2$	$a_3$	$B$		$a_1$	$a_2$	$a_3$	
0.3	0.6	0.1	$b_1$	0.4	$b_1$	0.120	0.240	0.040
			$b_2$	0.1	$b_2$	0.030	0.060	0.001
			$b_3$	0.5	$b_3$	0.150	0.300	0.050

Die Tafel  $p(A)$  enthält Information über das Merkmal  $A$ , das in die Kategorien  $a_1$ ,  $a_2$  und  $a_3$  eingeteilt ist. Die Tafel  $p(B)$  enthält Information über das Merkmal  $B$ , das seinerseits in die Kategorien  $b_1$ ,  $b_2$  und  $b_3$  unterteilt ist. Die Dimensionen sind fest bestimmten Merkmalen zugeordnet, so hat die Tafel  $p(A)$  in Richtung der Dimension von  $B$  (und eventueller weiterer Dimensionen) nur die Länge 1, hier entsprechend einer Zeile. Entsprechendes gilt für  $p(B)$  und die Dimension zu  $A$  (Spalten). Die Tafel  $p(A, B)$  enthält Information über  $A$  und  $B$ .

#### 1.4.2 Multiplikation von Tafeln

Die Tafel  $p(A, B)$  mit den konkreten Werten könnte errechnet worden sein durch  $p(A) \cdot p(B)$  (im allgemeinen gilt aber  $p(A, B) \neq p(A) \cdot p(B)$  - siehe Abschnitt 1.5). Dabei wird ein Wert einer Tafel, die eine bestimmte

Dimension nicht darstellt, für die Multiplikation mit den Werten einer zweiten Tafel in der betreffenden Dimension entsprechend der Anzahl der Werte wiederholt, wenn die zweite Tafel diese Dimension enthält. So entsteht die erste Zeile von  $p(A, B)$  durch Multiplikation der Werte 0.3, 0.6 und 0.1 aus  $p(A)$  jeweils mit demselben Wert 0.4 aus  $p(B)$ . Entsprechend wurde die linke Spalte von  $p(A, B)$  durch Multiplikation der Werte 0.4, 0.1 und 0.5 aus  $p(B)$  mit dem Wert 0.3 aus  $p(A)$  erhalten. Damit entsteht durch Multiplikation eine Tafel, die jede in den Ausgangstafeln enthaltene Dimension selbst auch enthält.

### 1.4.3 Marginalisierung

Umgekehrt kann in Tafeln, die Information über mehrere Merkmale enthalten, auch die Information bestimmter Merkmale **marginalisiert** werden, d.h. die Information der marginalisierten Merkmale wird vernachlässigt und geht damit in der marginalen Tafel verloren. Die Werte einer zu marginalisierenden Dimension werden durch Addition zu einem einzigen Wert zusammengefaßt. So kann im konkreten Fall  $p(A)$  bzw.  $p(B)$  aus  $p(A, B)$  durch Marginalisierung über  $B$  bzw.  $A$  gewonnen werden. Wir schreiben:

$$\begin{aligned} p(A) &= \sum_B p(A, B) \\ p(B) &= \sum_A p(A, B) \end{aligned}$$

Z.B. entsteht der Wert 0.3 von  $a_1$  in  $p(A)$  durch  $0.120 + 0.030 + 0.150$ . Enthält eine Tafel eine zu marginalisierende Dimension von vornherein nicht, so ändert sich durch die Marginalisierung nichts. Es gilt z.B.:

$$\begin{aligned} p(A) &= \sum_B p(A) \\ p(B) &= \sum_A p(B) \end{aligned}$$

Wird über mehrere Merkmale marginalisiert, so ist die Reihenfolge egal. Es gilt z.B.:

$$\sum_A \left( \sum_B p(A, B) \right) = \sum_B \left( \sum_A p(A, B) \right) = \sum_{A, B} p(A, B) = 1$$

### 1.4.4 Bedingte Tafeln / Verteilungen

Äquivalent zur Multiplikation mehrdimensionaler Tafeln ist auch die Division erklärt. Die Division wird insbesondere bei der Gewinnung bedingter Wahrscheinlichkeitstafeln aus gemeinsamen Wahrscheinlichkeitstafeln benötigt. Hat man eine Tafel, in der die gemeinsame Wahrscheinlichkeitsverteilung der Merkmale  $A, B$  abgelegt ist ( $p(A, B)$ ), so kann man z.B. die bedingte Tafel für das gegebene Merkmale  $B$  erhalten durch:

$$p(A|B) = \frac{p(A, B)}{\sum_A p(A, B)} = \frac{p(A, B)}{p(B)}$$



In unserem Beispiel erhielten wir<sup>1</sup>:

$p(A, B)$		$A$		
		$a_1$	$a_2$	$a_3$
$B$	$b_1$	0.3	0.6	0.1
	$b_2$	0.3	0.6	0.1
	$b_3$	0.3	0.6	0.1

umgekehrt gilt:

$$p(A, B) = p(A|B) \cdot \sum_A p(A, B) = p(A|B) \cdot p(B)$$

## 1.5 Unabhängigkeit und bedingte Unabhängigkeit

Eine entscheidende Frage bei der Betrachtung mehrerer Merkmale ist, ob diese Merkmale abhängig oder unabhängig voneinander sind. Es seien die a-priori-Wahrscheinlichkeiten eines Merkmals *Beruf* mit  $p(\text{Beruf})$  und entsprechend mit  $p(\text{Haarfarbe})$  die des Merkmals *Haarfarbe* bezeichnet. **Unabhängigkeit** zwischen diesen beiden Merkmalen besteht dann, wenn sich aus Wissen über die Ausprägung eines der Merkmale keine Veränderung in der a-posteriori-Verteilung des anderen Merkmals ergibt, bzw. aus der gemeinsamen Betrachtung keine zusätzliche Information über die Merkmale entsteht. Sind z.B. 15% der betrachteten Gesamtbevölkerung *blond* und auch unter der Gruppe der *Informatiker* 15% *blond*, so ergibt das Wissen darüber, daß jemand *blond* ist keine veränderte Erwartungslage in Bezug darauf, ob diese Person außerdem *Informatiker* ist. Gilt dieses auch für **alle anderen** Kombinationen der Ausprägungen von *Haarfarbe* und *Beruf*, so sind die Merkmale **unabhängig**. In Bezug auf eine gemeinsame Wahrscheinlichkeitsverteilung (i.d. Form einer mehrdimensionalen Tafel) schreiben wir:

$$p(\text{Beruf}) \cdot p(\text{Haarfarbe}) = p(\text{Beruf}, \text{Haarfarbe})$$

$\Rightarrow A \text{ und } B \text{ sind unabhängig}$

Besteht zwischen mehreren Merkmalen  $A, B, C$  und  $D$  **totale Unabhängigkeit** so gilt auch

$$p(A) \cdot p(B) \cdot p(C) \cdot p(D) = p(A, B, C, D)$$

$\Rightarrow A, B, C \text{ und } D \text{ sind total unabhängig}$

Aus der paarweisen Unabhängigkeit folgt noch nicht die totale Unabhängigkeit!

Ein für die Definition Bayes'scher Netze besonders interessantes Kriterium ist die **bedingte Unabhängigkeit** von Merkmalen. Übt ein Merkmal  $A$  **Einfluß** auf ein anderes Merkmal  $B$  aus, das seinerseits das Merkmal  $C$  beeinflusst, so werden i.d.R. auch  $A$  und  $C$  abhängig sein. Ist der Einfluß von  $A$  auf  $C$  aber vollständig durch den indirekten Einfluß von  $A$  über  $B$  auf  $C$  zu erklären, so ist  $C$  bedingt unabhängig von  $A$  bei gegebenem  $B$ . Formal schreiben wir:

$$p(A) \cdot p(C|B) = p(A, C|B)$$

$\Rightarrow A \text{ und } C \text{ sind bedingt unabhängig gegeben } B$

## 1.6 Datensatz und Kontingenztafeln

Aussagen über Verteilungen können insbesondere aus der Beobachtung realer Fälle abgeleitet werden. Man spricht hierbei von "Schätzung", da die Beobachtungen immer auch vom Zufall (bzw. nichtmeßbaren Störungen) beeinflusst werden. Dieser Einfluß nimmt mit der Menge der Beobachtungen ab (Schwaches Gesetz der Großen Zahlen), so wird z.B. beim Würfeln auch mit einem idealen Würfel bei 6 Würfeln kaum (mit geringer Wahrscheinlichkeit) jede Zahl genau einmal geworfen werden. Je mehr Würfe man aber durchführt desto näher wird man sich dem Wert  $1/6$  für den Anteil jeder Augenzahl an den Gesamtwürfeln nähern. Mit unendlich vielen Würfeln würde man sogar - mit Wahrscheinlichkeit 1 - die wahre Verteilung von exakt  $1/6$  für jede Augenzahl erreichen (Starkes Gesetz der Großen Zahlen)<sup>2</sup>. In einem Datensatz sind i.d.R. die Ausprägungskombinationen einiger Beobachtungen (Fälle) für mehrere Merkmale enthalten.

**Bsp.:**

<sup>1</sup>Die Werte in den einzelnen Spalten sind identisch, da in diesem speziellen Fall gilt:  $p(A, B) = p(A) \cdot p(B)$ , was aber nur bei Unabhängigkeit von  $A$  und  $B$  gilt (siehe Abschnitt 1.5)

<sup>2</sup>der Zusatz "mit Wahrscheinlichkeit 1" ist eine für den Beweis der Aussage unumgänglicher Zusatz

	Alter	Größe	Beruf	Vermögen	Haarfarbe
1. Fall	31 Jahre	191 cm	Kameltreiber	200 Kamele	schwarz
2. Fall	24 Jahre	183 cm	Informatiker	0 Kamele	blond
...	...	...	...	...	...

Aus einem solchen Datensatz lassen sich mehrdimensionale Tafeln - genannt Kontingenztafeln - gewinnen. Diese Tafeln enthalten die Merkmale in der schon geschilderten Art, wobei jedes Merkmal einer Dimension der Tafel entspricht. In dem Beispieldatensatz sind die Merkmale Alter, Größe und Vermögen noch nicht (ausreichend) diskretisiert<sup>3</sup> und müssen erst in **geeignete(re)** Kategorien eingeteilt werden. Jede Zelle entspricht dann einer Ausprägungskombination der Merkmale und enthält die Anzahl der Beobachtungen im Datensatz, die diese Ausprägungskombination aufweist.

Eine Tafel  $K'$ , die ein Merkmal  $A$  nicht enthalten soll, entspricht der marginalen Tafel  $\sum_A K$ .

## 1.7 Normierung

Aus einer Kontingenztafel  $K$  kann eine gemeinsame Wahrscheinlichkeitstafel  $T$  der enthaltenen Merkmale gewonnen werden durch **Normierung** von  $K$ :

$$T := \frac{K}{\sum K}$$

dabei steht $\sum K$	für die Gesamtsumme aller Zellen in $K$ , was der Anzahl der Beobachtungen aus denen $K$ gewonnen wurde entspricht
----------------------	--------------------------------------------------------------------------------------------------------------------------

In entsprechender Weise werden auch andere Tafeln normiert. Für eine normierte Tafel  $T$  gilt:

$$\sum T = 1$$

---

<sup>3</sup>Die Daten im Beispieldatensatz wie z.B. 31 Jahre oder 191 cm sind sicher nicht wirklich als stetige Größen exakt gemessen worden. Jedoch machen sehr fein unterteilte Kategorien (wie ..., 30 Jahre, 31 Jahre, 32 Jahre, ...) für die i.d.R. nur sehr wenige Beobachtungen (evtl. nur eine) vorliegen für die Schätzung von Wahrscheinlichkeitsverteilungen und Abhängigkeiten keinen Sinn, da sich daraus nur unhaltbare Schlußfolgerungen ziehen ließen, wie z.B.: "Jeder 31 Jährige ist 191 cm groß !?"

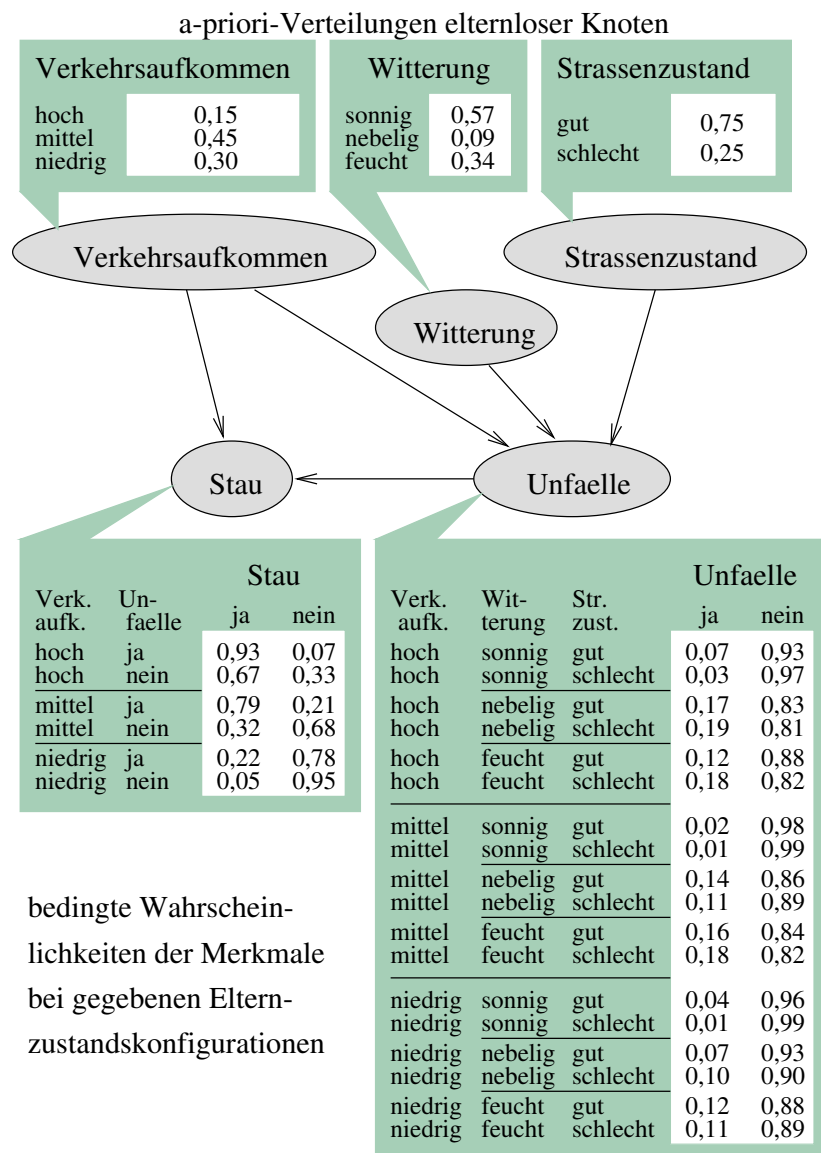


Abbildung 1: **Bayes'sches Netz** bestehend aus: dem **DAG**, sowie den **bedingten Wahrscheinlichkeiten** hier in Tabellenform (Dimensionszuordnung nicht erkennbar)

## 2 Bayes'sche Netze

### 2.1 Was ist ein Bayes'sches Netz?

Ein **Bayes'sches Netz** (teilweise auch unter dem Begriff **Kausales Netz** bekannt) ist ein statistisches Modell, das in der Lage ist in anschaulicher, grafischer Art komplexe statistische Abhängigkeitsstrukturen zwischen für den Gültigkeitsbereich des Modells relevanten Merkmalen darzustellen.

### 2.2 Bestandteile eines Bayes'schen Netzes

Für ein Bayes'sches Netz ist zunächst eine Auswahl der für die Beschreibung des fraglichen Zusammenhangs **relevanter Merkmale** von Bedeutung. Diese ausgewählten Merkmale können von verschiedener Art sein. Zum einen können grundsätzlich diskrete oder auch stetige Merkmale und zum anderen auch beobachtbare und unbeobachtbare (nicht direkt meßbare / gemessene) Merkmale dem Benutzer als relevant erscheinen. Die Mischung diskreter und stetiger Merkmale ist mit einer Einschränkung<sup>4</sup> möglich, wird aber in dieser Arbeit nicht behandelt. Vielmehr werden wir uns auf **diskrete** Merkmale beschränken. Dazu können auch stetige

<sup>4</sup>Es dürfen keine Kanten von einem stetigen Merkmal zu einem diskreten Merkmal enthalten sein

Merkmale durch die Einteilung in geeignete Kategorien in diskrete Merkmale überführt werden. Die Frage der (un-) beobachtbaren Merkmale spielt erst bei dem eigentlichen Ziel der Arbeit, Netzstrukturen aus gegebenen Daten abzuleiten, eine Rolle und wird daher erst genauer beleuchtet, wenn wir an diesem Punkt angelangt sind (Abschnitt 4.1).

Damit kommen wir zum zweiten wichtigen Teil der Bestimmung eines Bayes'schen Netzes, der **Netzstruktur**, die über die Merkmale gelegt wird. Die Struktur wird in der Form eines **DAG's** (**D**irected **A**cyclic<sup>5</sup> **G**raph) gegeben. Eine Kante von einem Merkmal (Knoten)  $A$  zu einem Knoten  $B$  drückt dabei einen direkten Einfluß von  $A$  auf  $B$  aus. Dieser Einfluß wird verschieden interpretiert. Es gibt Ansätze die eine Kante als streng **kausale** Abhängigkeit bewerten. Andere Betrachtungen sind weniger streng und sehen in der Richtung einer Kante lediglich eine für das Modell günstig zu treffende Wahl. Wir werden letzteren Ansatz verfolgen, da durch automatisierte Verfahren echte Kausalitäten kaum nachzuvollziehen sind. Auch bei nichtautomatisierten Verfahren ein Bayes'sches Netz durch Experten aufstellen zu lassen, kann die Richtung einer Kante durchaus in vielen Fällen von der Betrachtung des Sachverhalts abhängig gemacht werden.

**Bsp.:** Unterschiedliche "Richtungen" der Betrachtung

1. Betrachtungsweise:	Wenn es regnet wird die Straße naß	$Regen \rightarrow Nässe$
2. Betrachtungsweise:	Ob die Straße naß ist oder nicht läßt Rückschlüsse darüber zu, ob es geregnet hat oder nicht.	$Nässe \rightarrow Regen$

Dabei ist nicht nur die "Richtung der Betrachtung" im schlußendlich aufgestellten Netz von Bedeutung, sondern vielmehr auch in welcher "Richtung" das Wissen über die Wahrscheinlichkeitsverteilungen der Merkmale gegeben werden kann, insbesondere wenn dies durch einen Experten geschieht und sich nicht auf Daten gründet. Hiermit sind wir bei dem dritten Bestandteil eines Bayes'schen Netzes angelangt. Korrespondierend zu der Struktur der Merkmale müssen Wahrscheinlichkeiten in Form **bedingter Wahrscheinlichkeiten** für die einzelnen Ausprägungen jedes (diskreten) Merkmals, gegeben die Zustände aller Elternknoten formuliert werden. Eine einzelne Angabe kann z.B. die Form haben:

"**Wenn** jemand zwischen 180 cm und 190 cm groß ist, **dann** ist diese Person mit 83 prozentiger Wahrscheinlichkeit männlich." ( $\Rightarrow p(Geschlecht = männlich | Größe = 180 - 190cm) = 0,83$ )

Diese Form die Angabe der Wahrscheinlichkeiten auf Vorbedingungen zu stützen ("Wenn ... dann ...") macht es Experten oft leichter ihr Wissen anzugeben. Wie wir bereits angedeutet haben kann sich die Struktur eines Netzes auch danach richten, wie die Experten solche Angaben zu machen imstande sind. Wenn ein Knoten allerdings keine Elternknoten besitzt, was bei mind. einem Knoten in einem DAG der Fall sein muß, so müssen die **a-priori-Wahrscheinlichkeiten** der Ausprägungen des Knotens angegeben werden<sup>6</sup>. A-priori-Wahrscheinlichkeiten sind Wahrscheinlichkeiten, die ohne sich auf gegebene Zustände anderer Merkmale zu stützen den Anteil der einzelnen Ausprägungen an der potentiell unendlichen Gesamtmasse der durch das Merkmal beschriebenen Objekte oder Individuen angeben. Eine solche Angabe kann z.B. in der Form gegeben werden:

"12 Prozent der Bevölkerung sind zwischen 180 cm und 190 cm groß." ( $\Rightarrow p(Größe = 180 - 190cm) = 0,12$ )  
Solche Wahrscheinlichkeiten sind durch Experten vielfach nur schwer zu formulieren und können oft besser auf Daten und Messungen gestützt werden, wenn die Möglichkeit dazu besteht.

**Def.:** Ein Bayes'sches Netz besteht aus:

1. Einer Auswahl von für den Sachverhalt relevanten, (hier) diskreten Merkmalen
2. Einer Netzstruktur in der Form eines DAG's für diese Merkmale
3. Den bedingten Wahrscheinlichkeiten jedes Merkmals gegeben seine Eltern im DAG

## 2.3 (Un-) Abhängigkeitsstruktur und d-separation

Die Abhängigkeitsstruktur der Merkmale, wie sie durch ein DAG repräsentiert wird, ist intuitiv recht gut erfaßbar. Innerhalb dieser komplexen Struktur direkter und indirekter Abhängigkeiten verbergen sich vor allem auch bedingte Unabhängigkeiten.

*Es gilt: Ein Knoten  $A$  gegeben alle seine Eltern  $\mathbf{Pa}_A$  im DAG ist (bedingt) unabhängig von allen sonstigen Vorfahren.*

<sup>5</sup>Gerichtete Zirkel, d.h. Kanten, die die Möglichkeit eröffnen sich im DAG unter Beachtung der Kantenrichtungen im Kreis bewegen zu können sind verboten

<sup>6</sup>ein Punkt der vielfach für einen entscheidende Schwäche Bayes'scher Netze gehalten wird, uns hier aber nicht weiter beschäftigen wird

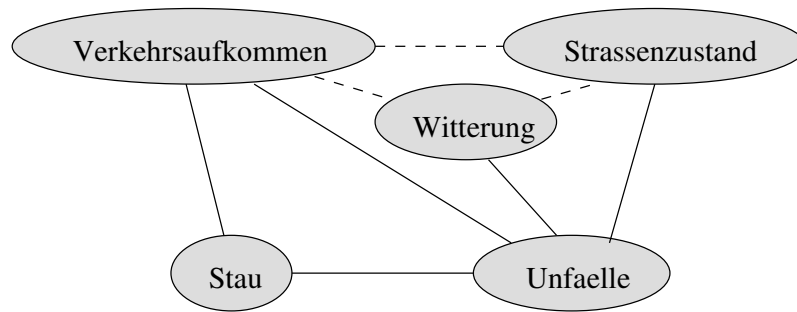


Abbildung 2: **Moralischer Graph** zum Bsp. aus Abbildung 1

Leichter zu durchschauen wird diese Unabhängigkeitsstruktur anhand des Konzepts der **d-separation**:

*Zwei Knoten A und B sind bedingt unabhängig gegeben eine Menge (anderer) Knoten S wenn es im moralischen Graph (s.u.) zum DAG keine Verbindung zwischen A und B gibt, die nicht über Knoten aus S führt. (siehe [Jensen (1996)])*

### 2.3.1 Der Moralische Graph

**Def.:** Der **moralische Graph** entsteht aus dem DAG durch die paarweise Verbindung (“Verheiratung”) aller Eltern eines jeden Knotens durch (ungerichtete) Kanten und die Überführung auch der restlichen Kanten in ungerichtete Kanten (durch Vernachlässigung der Richtung).

Im dargestellten Beispiel sind *Stau* und *Straßenzustand* gegeben *Verkehrsaufkommen* und *Unfälle* bedingt unabhängig. Ist nur *Unfälle* gegeben, so gibt es noch eine Verbindung im moralischen Graph über *Verkehrsaufkommen* (und *Witterung*) zwischen *Stau* und *Straßenzustand*, die damit nicht unabhängig sind.

## 2.4 Beschränkung der dargestellten Abhängigkeiten durch die Netzstruktur

Ein Bayessches Netz stellt durch seine Wahrscheinlichkeitsangaben einen Ausschnitt der Information dar, die in einer vollständigen Wahrscheinlichkeitstafel für alle betrachteten Merkmale darstellbar wäre. Diese von der Kantenanzahl abhängige Beschränkung auf einen Ausschnitt ist in zweierlei Hinsicht von Bedeutung: Zum einen muß es bei der Schätzung der Wahrscheinlichkeiten aus Daten<sup>7,8</sup>, im Sinne einer nicht zu starken Anpassung an diese Daten (*overfitting*), vermieden werden alle scheinbar im Datensatz enthaltenen Abhängigkeiten tatsächlich auch für real zu halten und in Kanten umzusetzen. Zum anderen kann diese mehr oder weniger starke Beschränkung in eine entsprechend effiziente Inferenz-Berechnung<sup>9</sup> des Netzes umgesetzt werden.

## 2.5 Nutzung Bayes’scher Netze

Ist ein Bayes’sches Netz erst einmal aufgestellt kann es variabel genutzt werden. Die normale Nutzung besteht darin, daß in einem konkret zu bewertenden Fall die Ausprägungen einiger Merkmale bekannt sind oder wenigstens einige Ausprägungen ausgeschlossen werden können. Solche Informationen können in das Netz eingebracht werden und die Auswirkungen auf die Wahrscheinlichkeitsverteilungen der übrigen Merkmale berechnet werden, die dann in der Form von **a-posteriori-Wahrscheinlichkeiten** ausgegeben werden, um eine Prognose über die Ausprägungen dieser noch unbekannten Merkmale in dem betrachteten Fall zu erhalten.

## 2.6 Stärken Bayes’scher Netze

Vorteile Bayes’scher Netze liegen in ihrer **anschaulichen**, d.h. von Experten vorgebbaren und. bewertbaren Art auch komplexe Abhängigkeitsstrukturen abzubilden. Auch die Verwendung der **bedingten Wahrscheinlichkeitsangaben** - außer bei elternlosen Knoten - macht es in vielen Fällen für den menschlichen Experten relativ einfach selbst Vorgaben zu machen, bzw. vorgegebene Wahrscheinlichkeiten zu verifizieren.

<sup>7</sup>Daten, Datensatz : Aufzeichnung der konkreten Ausprägungen als relevant erachteter Merkmale in einer Menge von beobachteten Fällen

<sup>8</sup>Daten werden u.a. eine wichtige Grundlage zur Erstellung Bayes’scher Netze darstellen

<sup>9</sup>Nutzung des Netzes zur Prognose

Ein weiterer Vorteil liegt generell in dem **konsequenten Rechnen auf der Basis von Wahrscheinlichkeiten**, wodurch immer gewährleistet ist, daß mit vergleichbaren Größen operiert wird - im Gegensatz zu anderen Expertensystemen, die z.B. mit Bewertungen auf Punkteskalen arbeiten wodurch die Gewichtung verschiedener Merkmale gegeneinander oft schwer nachzuvollziehen ist und keiner in sich konsistenten Logik folgt. Zuguter Letzt besteht ein großer Vorteil darin, daß auch mit **unvollständigen Angaben** gerechnet werden kann, bzw. daß durch das sukzessive Einfließen lassen einzelner Informationen (über einzelne Merkmale) deren Einfluß auf die interessierende(n) Größe(n) erkennbar wird. Dieses kann auch umgekehrt zur Ermittlung der den stärksten Ausschlag gebenden Merkmale zu einem bestimmten Zeitpunkt (Wissensstand) genutzt werden (eine Information über welches Merkmal ergäbe die stärkste Veränderung der interessierenden(n) Größe(n)?). Ein Vorteil den Bayes'sche Netze insbesondere auch **Neuronalen Netzen** voraus haben. Dabei können die Merkmale zu denen Information gegeben werden kann auch von Fall zu Fall wechseln. Auch die wechselnde Nutzung eines Merkmals als zu prognostizierend in dem einen, bzw. als bekannt in einem anderen Fall ist ohneweiteres möglich.

### 3 Inferenz - Junction-Tree

Wir haben bisher gesehen was wir im Sinne dieser Arbeit als ein Bayes'sches Netz betrachten werden. Außerdem haben wir uns zuvor schon damit beschäftigt einige Operationen auf mehrdimensionalen Tafeln, wie sie für die Wahrscheinlichkeits-Spezifikationen eines Bayes'schen Netzes hier Verwendung finden, zu definieren. Beides werden wir nun in einem ersten Schritt zu einem sinnvollen Verfahren zur Inferenz-Berechnung in Bayes'schen Netzen zusammenführen. Dabei geht es um die Frage wie ein fertig aufgestelltes Netz zur Propagation der Wahrscheinlichkeitsverteilungen bestimmter Merkmale genutzt werden kann. Dabei wird aus Effizienzgründen besonderer Wert auf die Ausnutzung der in der Netzstruktur codierten bedingten Unabhängigkeiten gelegt werden.

#### 3.1 Bedeutung des Problems

Die Berechnungen, die für die Nutzung Bayes'scher Netze (Inferenzberechnung) relevant sind scheinen auf den ersten Blick wenig mit den Berechnungen zu tun zu haben, mit denen die Struktur eines Bayes'schen Netzes ermittelt werden kann. Dennoch besteht ein Zusammenhang. Daher werden wir uns zunächst der Frage widmen, wie in einem Bayes'schen Netz gerechnet wird, bevor wir uns damit beschäftigen ein solches Netz aufzustellen, d.h. "die" (eine vernünftige) Netzstruktur zu ermitteln. Berechnet werden müssen dabei neben den a-priori-Wahrscheinlichkeiten der einzelnen Ausprägungen eines Merkmals (Knotens), um für eine dem Benutzer verständliche Darstellung des enthaltenen Wissens zu sorgen, vor allem das Einfließen lassen von durch den Benutzer gegebenem Wissen über nicht mehr mögliche (mit Wahrscheinlichkeit = 0 aufgetretene) Ausprägungen einzelner Knoten, um anschließend die a-posteriori-Verteilungen der übrigen, im konkreten Fall (noch) unbeobachteten Merkmale ausgeben zu können. Auf diese wird der Benutzer dann seine Entscheidung in dem betrachteten Fall gründen.

Von Bedeutung auch für das eigentlich zu behandelnde Problem, dem Finden einer Netzstruktur, ist die Frage der Inferenz-Berechnung, weil eine Netzstruktur in der durch ihre Komplexität nicht mehr in annehmbarer Zeit gerechnet werden kann keinen Nutzen bringt und wir also neben einigen anderen Aspekten auch diesen Umstand nicht unberücksichtigt lassen können.

#### 3.2 Grundgedanke

Ein Bayes'sches Netz stellt ja, wie bereits erwähnt, eine anschauliche Form dar Wissen über die Wahrscheinlichkeitsverteilungen mehrerer Merkmale wiederzugeben und dabei Abhängigkeiten zwischen diesen Merkmalen zu berücksichtigen. Letztenendes beinhaltet ein Bayes'sches Netz immer die Information mit der eine gemeinsame Wahrscheinlichkeitstafel<sup>10</sup> aller enthaltenen Merkmale errechnet werden kann.

*Die vollständige gemeinsame Wahrscheinlichkeitstafel kann durch Ausmultiplizieren aller zu den einzelnen Knoten gegebenen bedingten Wahrscheinlichkeitstafeln gewonnen werden:*

$$p(\mathbf{X}|\mathbf{S}^h) = \prod_i p(X_i|\mathbf{Pa}_i, \mathbf{S}^h) \quad (1)$$

---

<sup>10</sup>wobei ein Bayes'sches Netz i.d.R. gerade nicht alle Information wiedergibt, die in einer vollständigen Wahrscheinlichkeitstafel darstellbar wären (aber natürlich nicht dargestellt werden müssen)

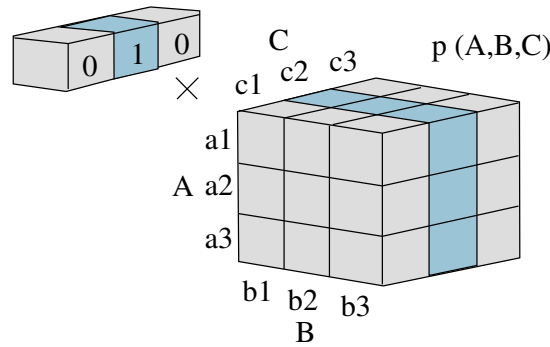


Abbildung 3: Multiplikation einer Tafel der Merkmale  $A$   $B$  und  $C$  mit einem Informationsvektor zu  $C$  d.h. “Nullsetzen” der ausgeschlossenen Zustandskombinationen

$$\left| \begin{array}{ll} \text{Es sind: } \mathbf{Pa}_i & \text{die Elternknoten von } X_i \\ S^h & \text{die (Hypothese über die) Netzstruktur} \end{array} \right|$$

Es entsteht damit bei  $n$  Merkmalen eine  $n$ -dimensionale Tafel, wie sie für drei Merkmale  $A, B$  und  $C$  in Abbildung 3 (rechts) dargestellt ist. Informationen darüber welche Zustände eines Knotens (noch) möglich sind und welche (bereits) ausgeschlossen werden können, kann in der Form eines **Informationsvektors** gegeben werden, der die Werte Null für ausgeschlossene, bzw. Eins für (noch) mögliche Ausprägungen enthält.

**Bsp.: Informationsvektor** für die Information, daß eine Person braun- oder schwarzhaarig ist:

Haarfarbe				
blond	braun	rot	schwarz	sonst.
0	1	0	1	0

Dieser Vektor (Abbildung 3 links) “liegt” für den  $i$ -ten Knoten auch in der  $i$ -ten Dimension<sup>11</sup>. Damit kann durch Multiplikation von  $p(\mathbf{X}|S^h)$  mit diesem Vektor und anschließende Normierung der neuen Tafel  $p(\mathbf{X}|S^h, \text{Information})$  das Einfließen von Information in das Netz berechnet werden, da durch die Multiplikation mit dem Informationsvektor die Wahrscheinlichkeiten aller durch den Benutzer ausgeschlossener Knoten-Zustands-Konfigurationen auf Null gesetzt werden. Mit der anschließenden Normierung der neuen Tafel wird erreicht, daß die Gesamtwahrscheinlichkeit wieder 1 beträgt ( $\sum p(\mathbf{X}|S^h, \text{Information}) = 1$ ).

Die einzelnen Wahrscheinlichkeiten der Knotenausprägungen erhält man vor der ersten sowie nach jeder Inferenzberechnung durch Marginalisierung der gemeinsamen Wahrscheinlichkeitstafel

$$(\sum_{\mathbf{X} \setminus X_i} p(\mathbf{X}|S^h) \text{ bzw. } \sum_{\mathbf{X} \setminus X_i} p(\mathbf{X}|S^h, \text{Information}) \forall X_i \in \mathbf{X}).$$

Von der eigentlichen Idee her ist dieser Ansatz einfach, wegen der mit zunehmender Anzahl der Knoten extrem wachsenden Tafel  $p(\mathbf{X}|S^h)$  ist diese Art der Berechnung allerdings i.d.R. inakzeptabel aufwendig - ein Problem, dem wir nicht zum letzten Mal begegnet sind.

### 3.3 Cluster-Tree / Junction-Tree

Ein Bayes’sches Netz stellt bedingte Abhängigkeiten und damit eben auch bedingte Unabhängigkeiten dar, es ist also, wie an der Spezifikation des Bayes’schen Netzes selbst schon zu erkennen ist, eben nicht notwendig eine gemeinsame Wahrscheinlichkeitstafel aller Knoten aufzustellen. Man ist stattdessen bestrebt gemeinsame Wahrscheinlichkeitstafeln statt für die Gesamtmenge der Merkmale nur für Teilmengen der Knoten aufzustellen. Diese Teilmengen müssen sinnvoll gewählt werden. Optimal wäre es wenn die Tafeln zu den Knoten-teilmengen nicht größer wären als die ursprünglichen bedingten Wahrscheinlichkeitstafeln zu den einzelnen Knoten, d.h. die Tafeln sollen nicht größer werden als dies zur Darstellung der im Netz enthaltenen Abhängigkeiten nötig ist. Die gesuchten Teilknotenmengen entsprechen den Cliques im moralischen Graphen<sup>12</sup> zum DAG.

**Def.:** Als **Cliques** bezeichnen wir maximal verbundene Teilgraphen, das sind Ausschnitte eines Graphen in denen jeder Knoten mit jedem anderen verbunden ist.

<sup>11</sup>wir wollen allgemein von einer festen Zuordnung eines Knotens im Netz zu einer Dimension der korrespondierenden Tafeln ausgehen

<sup>12</sup>Der **moralische Graph** wird durch die “Verheiratung” (paarweise Verbindung durch eine (ungerichtete) Kante) aller Elternknoten und durch die anschließende Überführung in einen ungerichteten Graphen gewonnen (siehe auch Abschnitt 2.3 und 2.3.1).

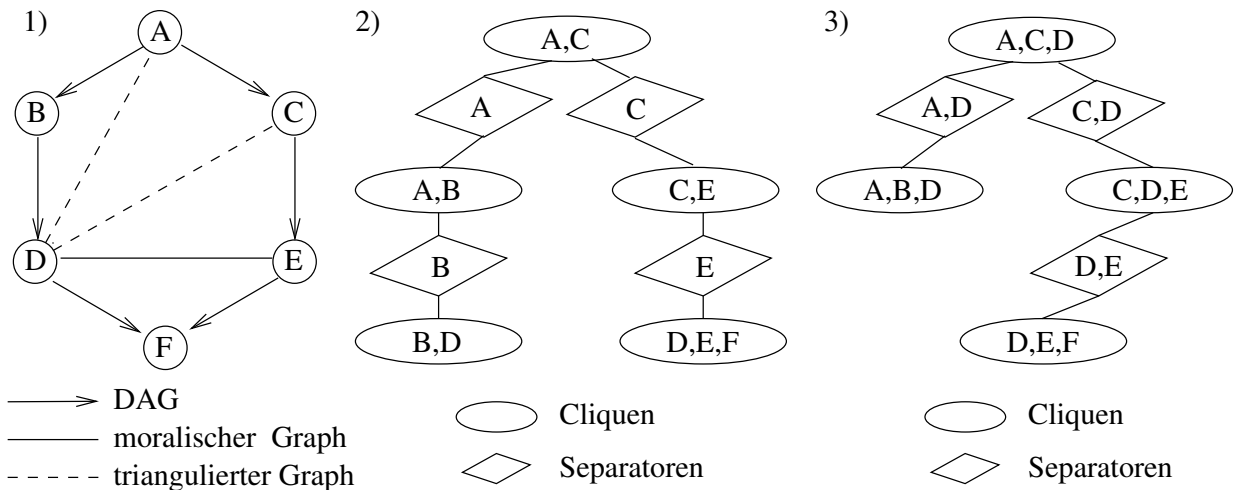


Abbildung 4:

- 1) DAG → moral. Graph → triangulierter Graph
- 2) Cluster-Tree (entsprechend moral. Graph)
- 3) Junction-Tree (entsprechend triangulierter moral. Graph)

Der Einfluß einer gegebenen Information ist dabei i.d.R. nicht auf Knoten innerhalb einer Clique beschränkt und so muß Information auch zwischen den Cliquen weitergereicht werden können.

*Information, die zwischen zwei Cliquen weitergegeben werden muß, kann nur jene Knoten betreffen, die in beiden Cliquen enthalten sind, das ist die Schnittmenge oder **Separatormenge** der betreffenden Cliquen.*

Im Sinne einer einfachen Berechnung sollen die Cliquen mit den dazugehörigen Tafeln in einem Baum organisiert werden, wobei die Separatoren, zu denen ebenfalls die gemeinsamen Wahrscheinlichkeitstafeln der enthaltenen Knoten gebildet werden, als Verbindungskanten fungieren.

**Def.:** Um den Baum aufzustellen benutzen wir folgendes Verfahren:

*Es werden zunächst die beiden Cliquen, die über die größte Separatormenge verbunden sind zu einem initialen Baum zusammengefügt. Danach werden sukzessive die jeweils größte verbliebene Separatormenge und die durch sie verbundenen Clique dem Baum angeschlossen, wenn die Baumstruktur dabei erhalten bleibt. Einen so aufgestellten Baum nennen wir **Cluster-Tree**.*

([Jensen (1996)])

Um dabei zu gewährleisten, daß verschiedene Tafeln, die dasselbe Merkmal beinhalten auch dieselbe Information zu diesem Knoten wiedergeben, muß ein Knoten der in zwei Cliquen auftaucht auch in jeder Separatormenge auf dem Weg zwischen den beiden Cliquen enthalten sein. Es zeigt sich, daß diese Bedingung mit dem bisher beschriebenen Verfahren zur Erstellung eines Cluster-Tree's noch nicht für jeden Graphen erfüllt ist (siehe Abbildung 4 2) Merkmal D). Sie ist aber in jedem Falle dann erfüllt, wenn der zugrundeliegende Graph trianguliert ist.

**Def.:** Als **trianguliert** wird ein Graph dann bezeichnet wenn er - ungeachtet eventueller Kantenrichtungen - keinen Zirkel mit mehr als drei Knoten beinhaltet, ohne daß dieser Zirkel abkürzbar wäre.

Ein nicht triangulierter Graph kann durch Hinzufügen von Kanten i.d.R. auf verschiedene Arten in einen triangulierten Graphen überführt werden. So ist z.B. ein zugehöriger vollständig verbundener Graph trianguliert. Im Sinne des Verfahrens muß aber versucht werden einen Graphen, sofern er die Bedingung noch nicht erfüllt so zu triangulieren, daß möglichst wenige Kanten eingefügt werden bzw. die Cliquen und damit die aufzustellenden Tafeln klein gehalten werden. Wir sollten dabei aber nicht vergessen, daß diese Kanten lediglich eingefügt werden um die korrekte Übermittlung des Wissens zwischen den Cliquen zu gewährleisten, wenn wir danach in



der beschriebenen Art und Weise einen Baum aufstellen. Jedoch sind diese Kanten keine informativen Kanten im Sinne der Darstellung (weiterer) bedingter Abhängigkeiten. Wir nehmen an dieser Stelle nur in Kauf, daß die Cliques-Tafeln größer werden als zur Darstellung der enthaltenen Information eigentlich nötig wäre. Es handelt sich also lediglich um andere Ausschnitte (marginale Tafeln) der großen gemeinsamen Wahrscheinlichkeitstafel  $p(\mathbf{X}|S^h)$  (gemäß Gleichung (1)), die bei allen Berechnungen unsere Referenz sein soll. Eine Methode zur Triangulation von Graphen und gleichzeitiger Identifikation der Cliques kann in [Jensen (1996)] gefunden werden. Mit einer einfachen Erweiterung, um möglichst wenige Kanten zur Triangulierung einzufügen, lautet das Verfahren:

*Wähle einen Knoten und verbinde alle Nachbarn paarweise durch weitere Kanten (fill-ins), soweit sie nicht schon verbunden sind. Wähle dabei den Knoten bei dem die wenigsten fill-ins entstehen. Der ausgewählte Knoten mit allen Nachbarn bildet eine Clique. Entferne danach den ausgewählten Knoten und alle anliegenden Kanten und führe das Verfahren mit dem Restgraphen erneut durch, bis alle Knoten einer Clique zugeordnet wurden. Der Ursprungsgraph plus alle so erstellten fill ins ist dann trianguliert. Unverbundene Teilgraphen können (nur) getrennt trianguliert werden.*

**Def.:** Zu einem triangulierten moralischen Graphen kann dann ohne weitere Probleme, wie erläutert ein Baum aufgestellt werden, der die genannten Bedingungen für die Berechnung erfüllt und dann als **Junction-Tree** bezeichnet wird.

### 3.4 Inferenzberechnung mit dem Junction-Tree

Um einen konsistenten Junction-Tree zu erhalten benötigen wir die **gemeinsamen Wahrscheinlichkeitstafeln** zu den Cliques und Separatoren.

**Def.:** Ein Junction-Tree ist konsistent wenn für alle Knoten  $X_i$  gilt, daß jede Cliques- / Separatoren-Tafel, die  $X_i$  beinhaltet auch dieselbe Information über diesen Knoten enthält. d.h.  $\sum_{\mathbf{X} \setminus X_i} T_1 = \sum_{\mathbf{X} \setminus X_i} T_2$ , wobei  $T_{1,2}$  für alle entsprechenden Paare aus der Menge der Cliques- und Separatoren-Tafeln stehen.

Die Cliques- und Separatoren-Tafeln könnten natürlich aus der gemeinsamen Wahrscheinlichkeitstafel zu allen Knoten berechnet werden:

$$p(\mathbf{X}_{cl}|S^h) = \sum_{\mathbf{X} \setminus \mathbf{X}_{cl}} \prod_i p(X_i|\mathbf{Pa}_i, S^h) \quad (2)$$

$$p(\mathbf{X}_{sep}|S^h) = \sum_{\mathbf{X} \setminus \mathbf{X}_{sep}} \prod_i p(X_i|\mathbf{Pa}_i, S^h) \quad (3)$$

dabei sind	$S^h$	die gegebene Netzstruktur
	$\mathbf{X}$	alle Knoten in $S^h$
	$\mathbf{X}_{cl}$	die Knoten in einer Clique $cl$
	$\mathbf{X}_{sep}$	die Knoten in einem Separator $sep$
	$X_i$	der i-te Knoten in $S^h$
	$\mathbf{Pa}_i$	Menge der Eltern von $X_i$

Dieser Ansatz soll aber gerade vermieden werden. Es zeigt sich, daß der korrekte konsistente Junction-Tree erhalten werden kann, indem alle Tafeln (Cliques und Separatoren) zunächst mit 1 (alle Elemente) initialisiert werden. danach werden die bedingten Wahrscheinlichkeitstafeln für alle Knoten in jeweils **eine** Home-Clique dieses Knotens einmultipliziert.

**Def.:** Eine Home Clique zu einem Knoten  $X_i$  ist eine Clique die sowohl  $X_i$  selbst als auch alle Eltern-Knoten von  $X_i$  ( $\mathbf{Pa}_i$ ) enthält.

Durch die DAG-Struktur und die Bedingungen zum Aufbau des Junction-Trees ist gewährleistet, daß mindestens eine Home-Clique zu jedem Knoten existiert.

Nachdem der Junction-Tree aufgebaut, und die Tafeln wie beschrieben initialisiert wurden muss eine initiale Propagation erfolgen, um den gesuchten konsistenten Junction-Tree zu erhalten.

Bevor wir beschreiben wie die Propagation in einem Junction-Tree durchgeführt werden kann, soll auf eine eventuelle Fehlerquelle bei der Schätzung der gemeinsamen Wahrscheinlichkeitstafeln zu Cliques- und Separatormengen sei an dieser Stelle noch gesondert hingewiesen: Die Cliques und Separatortafeln dürfen nicht

direkt aus einem zur Verfügung stehenden Datensatz abgeleitet werden, indem sie etwa als Kontingenztafeln aufgestellt und anschließend normiert werden. Sowohl die von moralischen als auch die von Triangulationskanten betroffenen Cliquentafeln (und damit evtl. auch die zugehörigen Separatorentafeln) sind rein von ihrer Größe her in der Lage Abhängigkeiten zwischen den enthaltenen Werten darzustellen, die nicht aus dem dazugehörigen DAG ableitbar sind. Solche Abhängigkeiten würden aber durch die direkte Ableitung aus Daten in den Tafeln dargestellt<sup>13</sup>. Es ist also auch bei der Stützung der Wahrscheinlichkeitsschätzung auf Daten nötig, zuerst die bedingten Wahrscheinlichkeitstafeln aus den Daten abzuleiten (Kontingenztafeln + Normierung), um dann aus ihnen die gemeinsamen Wahrscheinlichkeitstafeln zu den Cliques und Separatoren zu gewinnen. Die Propagation kann nun wie folgt durchgeführt werden: Nehmen wir an  $\mathbf{V}$  und  $\mathbf{W}$  sind benachbarte Cliques im Junction-Tree. Zu Clique  $\mathbf{V}$  (zu einem der enthaltenen Merkmale) wurde Information in der Form eines Informations-Vektors gegeben und durch Multiplikation mit der Cliquentafel  $p(\mathbf{V}|S^h)$  zu  $\mathbf{V}$  (Abbildung 3) und Normierung dieser neuen Tafel  $p'(\mathbf{V}|S^h)$  in den Junction-Tree eingebracht. Zunächst wird für die Kante also den Separator  $\mathbf{VW}_{\text{sep}}$  zwischen  $\mathbf{V}$  und  $\mathbf{W}$  eine neue Separatorentafel  $p'(\mathbf{VW}_{\text{sep}}|S^h)$  entsprechend Gleichung (3) aus  $p'(\mathbf{V}|S^h)$  (nicht etwa aus  $p(\mathbf{W}|S^h)$ ) erzeugt. Danach wird die neue Tafel zur Clique  $\mathbf{W}$  gewonnen durch (nach [Jensen (1996)]):

$$p'(\mathbf{W}|S^h) := p(\mathbf{W}|S^h) \frac{p'(\mathbf{VW}_{\text{sep}}|S^h)}{p(\mathbf{VW}_{\text{sep}}|S^h)} \quad (4)$$

**Def.:** *Damit ist eine Informationsweiterleitung von Clique  $\mathbf{V}$  nach Clique  $\mathbf{W}$  erklärt.*

Die gesamte Neuberechnung eines Junction-Tree's, wenn Information gegeben wird, kann z.B. erfolgen durch:

1. Multiplikation des Informations-Vektors mit jeder Cliquentafel, die das Merkmal enthält zu dem der Informationsvektor gegeben wurde.
2. Wahl einer beliebigen Root-Clique im Junction-Tree.
3. Informationsweiterleitung entsprechend Gleichung (4) ausgehend von der Root-Clique bis zu den Blättern des Junction-Tree's und anschließend eine weitere Informationsweiterleitung ebenfalls gemäß Gleichung (4) in umgekehrter Reihenfolge von den Blättern zur Root-Clique. Damit wird bei einem Update-Durchlauf über jede Kante im Junction-Tree zweimal (in jeder Richtung einmal) Information weitergeleitet.

Dieses Verfahren ist als "HUGIN propagation"<sup>14</sup> bekannt und u.a. auch in [Jensen et al. (1990)] und [Dawid (1992)] beschrieben. Es stellt eine Abwandlung des in [Lauritzen & Spiegelhalter (1988)] beschriebenen Verfahrens dar. Eine andere Methode zur Informationsweiterleitung kann in [Shafer & Shenoy (1990)] gefunden werden. Weitere Propagationmethoden für diskrete Variablen sind z.B. in [Shachter (1988)], [Pearl (1986)] und [D'Ambrosio (1991)] beschrieben. Für Gauss'sche Variablen (u.U. gemischt mit diskreten Variablen) sind Verfahren in [Shachter & Kenley (1989)] und [Lauritzen (1992)] zu finden.

### 3.5 Konsequenzen

Auf unsere Fragestellung bezogen, worauf im Sinne einer effizienten Inferenzberechnung bei der Erstellung einer Netzstruktur geachtet werden muß, können wir nun festhalten, daß für die Komplexität der Berechnung neben der reinen Anzahl der Merkmale insbesondere die Größe der Cliques und damit der entsprechenden Tafeln von Bedeutung ist. Zur Anzahl der Merkmale ist zu sagen, daß die interessierenden Merkmale vom Benutzer zu bestimmen sind. Sollten sich durch die Berechnungen zur Bestimmung der Netzstruktur Merkmale als nicht so bedeutend erweisen, wie vielleicht vom Benutzer zunächst erwartet, so soll es auch an ihm liegen diese Merkmale wieder aus dem Netz zu streichen. Automatisch soll darauf aber kein Einfluß genommen werden. Der andere Punkt, die Größe der zum Junction-Tree zu erzeugenden Tafeln, wird seinerseits bestimmt von der Komplexität des zugrundeliegenden DAG's, kann aber durch die u.U. notwendige Triangulierung zusätzlich in die Höhe getrieben werden. Allerdings sollte dieses Problem, abhängig von der verwendeten Triangulationsmethode, nicht allzusehr in's Gewicht fallen. An Abbildung 4 können wir erkennen, daß durch die Triangulation statt einer Drei-Knoten- und vier Zwei-Knoten-Cliques vier Drei-Knoten-Cliques entstanden sind. Auch wenn man den Zirkel aus Abbildung 4 durch weitere Knoten vergrößern würde, entstünden keine Cliques mit mehr als drei Merkmalen. Erst bei komplexeren Mehr-Zirkel-DAG's kann je nach verwendeter Methode die Größe einzelner Cliques durch Triangulation um mehr als ein Merkmal steigen. Alles in allem bleibt aber die Dimension des

<sup>13</sup> daß nicht alle aus einem Datensatz scheinbar herleitbaren Abhängigkeiten dargestellt werden sollen wird noch einen gewichtigen Punkt dieser Arbeit darstellen

<sup>14</sup> HUGIN ist ein Propagation-Tool für Bayes'sche Netze, daß in Zusammenarbeit mit der Aalborg University (Dänemark) entwickelt wurde.

Ausgangs-DAG's hauptauschlaggebend für eventuell nicht mehr in annehmbarer Zeit berechenbare Inferenz. Diese Dimension in vertretbaren Grenzen zu halten wird also ein wichtiges Anliegen bei der Gewinnung einer Netzstruktur sein; dieses jedoch auch noch aus anderen Gründen, wie wir noch sehen werden.

## 4 Grundlagen und Zielsetzung

Bisher haben wir gesehen wie ein Bayes'sches Netz spezifiziert wird und wie in einem solchen Netz - abhängig von der Komplexität des Netzes - effizient gerechnet werden kann. Nun müssen wir uns einige Gedanken darum machen was wir von einem solchen Netz erwarten, worauf wir es gründen und welche Folgen das für unser weiteres Vorgehen zur Bestimmung der Netzstruktur haben muß. Dabei wird die Komplexität der Abhängigkeiten unter den Merkmalen durch die Netzstruktur abgebildet, die damit bestimmt mit wievielen Werten (Parametern) die Abhängigkeiten letztlich zu beschreiben sind. Vor diesem Hintergrund muß die Bestimmung der Netzstruktur gesehen werden. Bei der Netzstruktursuche soll auf verschiedene Arten von Wissen zurückgegriffen werden, zum einen handelt es sich dabei um Vorwissen - i.d.R. von einem Experten vorgegeben, zum anderen um die Aufzeichnung der Merkmalsausprägungen mehrerer (möglichst vieler) konkreter Fälle (Datensatz).

### 4.1 Bayes'sche Netze als Modellannahme

Ganz allgemein ist ein Bayes'sches Netz ein Modell. In sehr vielen, ganz verschiedenen Bereichen finden unterschiedlichste Modelle Verwendung. Auch in der Statistik wird noch mit diversen anderen Modellannahmen gearbeitet, wie sie z.B. bei der (linearen) Regressionsanalyse oder auch der Diskriminanzanalyse getroffen werden. Gemein ist allen Modellen, daß sie einen, für den zu behandelnden Problembereich angemessenen Ausschnitt der Wirklichkeit repräsentieren, wobei i.d.R. neben der Abgrenzung des Modellausschnitts nach außen ein weiteres Ziel darin besteht, über den im Modell dargestellten Bereich selbst verschiedene vereinfachende Annahmen zu treffen.

Ein Bayes'sches Netz soll solche Merkmale enthalten, deren Wahrscheinlichkeitsverteilungen entweder prognostiziert werden sollen oder die zur Prognose dieser Merkmale als relevant betrachtet werden und über die später in einem konkret zu bestimmenden Fall zumindest Teilweise vom Benutzer Information gegeben werden kann - wobei in der Nutzung des Netzes die Rolle der Merkmale auch von Fall zu Fall wechseln kann, je nachdem über welche der Merkmale Informationen vorliegen. Wie auch immer, durch die Wahl der Merkmale wird also der Ausschnitt der Wirklichkeit gewählt, den wir durch das Modell spezifizieren wollen. Innerhalb dieses Bereichs ist eine Einschränkung durch das Modell selbst gegeben, durch das von vornherein festgelegt ist, daß sich das darzustellende Problem durch die Einflüsse verschiedener Merkmale aufeinanderder spezifizieren läßt und daß der Einfluß den die Merkmale untereinander entwickeln in der Form bedingter - mehr oder weniger auch als kausal zu interpretierender<sup>15</sup> - Abhängigkeiten darstellbar ist. Eine weitere entscheidende Vereinfachung gegenüber der "Wirklichkeit" wird erst bei dem Aufstellen eines konkreten Bayes'schen Netzes durch die Auswahl der tatsächlich in die Netzstruktur umgesetzten Abhängigkeiten getroffen. In "Wirklichkeit" wird es so sein, daß tatsächlich alle Merkmale direkten Einfluß aufeinander ausüben. Als Beispiel sei das Paradebeispiel der Chaos-Theoretiker angeführt, nachdem der Schlag eines Schmetterlingsflügels in China einen Tornado in Nordamerika auslösen kann. Die Frage wird aber sein welche Abhängigkeiten tatsächlich zu verifizieren sind bzw. durch die Darstellung welcher (wievieler) Abhängigkeiten man ein möglichst optimales Modell erhält. Wir wollen hier als eine weitere Einschränkung Bayes'scher Netze festlegen, daß die in ein solches Netz aufgenommenen Merkmale in jedem Fall beobachtbar sein müssen.

### 4.2 Parameter und Dimension eines Bayes'schen Netzes

Bei einem Bayes'schen Netz handelt es sich um ein sogenanntes **parametrisiertes Modell**. Wie wir bereits gesehen haben wird ein Bayes'sches Netz normalerweise durch seine Netzstruktur und den dazu korrespondierenden bedingten Wahrscheinlichkeiten gegeben. Was ist dabei nun genau der Zusammenhang zwischen Parametern und bedingten Wahrscheinlichkeiten, wie wir sie bisher verwendet haben? Die beiden Formen sind ineinander überführbar. Unter dem Begriff Parameter verstehen wir in diesem Zusammenhang die bedingten Wahrscheinlichkeiten der ersten  $r_i - 1$  Zustände eines Knotens  $X_i$  gegeben die Ausprägungen seiner Eltern  $\mathbf{Pa}_i$ . Denn als Parameter wollen wir nur solche Werte bezeichnen, die keine Redundanz aufweisen. Die bedingte Wahrscheinlichkeit des letzten ( $r_i$ -ten) Zustands eines Knotens  $X_i$  ist aber durch die ersten  $r_i - 1$

<sup>15</sup>Es existieren Ansätze in denen die Interpretation der Kantenrichtungen als kausale Richtungen sehr strikt verfolgt wird, wir wollen in diesem Punkt aber diesen hohen Anspruch nicht erheben.

Werte bereits gegeben, da sich die Summe der bedingten Wahrscheinlichkeiten bei einer jeweils gegebenen Elternzustandskonfiguration zu 1 ergeben muß. Diese vielleicht kleinlich wirkende Abgrenzung von Parametern und bisher verwendeten bedingten Wahrscheinlichkeitstafeln ist deswegen von besonderer Bedeutung, weil die Anzahl der zu schätzenden Parameter, die der **Dimension** einer Netzstruktur  $S^h$  entspricht ein exaktes Maß für die tatsächlich in einer Netzstruktur umgesetzten Abhängigkeitsverhältnisse darstellt. Die Dimension einer Netzstruktur  $S^h$  ist bestimmt durch:

$$\dim(S^h) = \sum_{i=1}^n q_i \cdot (r_i - 1) \quad (5)$$

$n$	ist die Anzahl der Knoten in $S^h$
$q_i$	ist die Anzahl der Elternzustandskonfigurationen eines Knotens $X_i$
$r_i$	ist die Anzahl der Ausprägungen von $X_i$

Die Dimension eines Bayes'schen Netzes sollte nicht mit der Dimensionalität einzelner Tafeln verwechselt werden!

### 4.3 Ziel

Die Güte mit der ein Bayes'sches Netz die Wirklichkeit abbildet und damit die Genauigkeit der Prognosen, die es liefern kann, hängt, wie bei allen parametrisierten Modellen, eben von der Wahl der Parameter - hier der bedingten Wahrscheinlichkeiten - ab. Um ein gutes Modell zu erhalten müssen demnach einerseits die Parameter - die wir im folgenden immer mit  $\theta$  bezeichnen - möglichst gut geschätzt werden. Wir werden uns mit dieser Frage beschäftigen, obwohl sie nicht das eigentliche Thema dieser Arbeit darstellt. Welche Parameter aber andererseits zu schätzen sein werden, wird durch die - Hypothese über die - Netzstruktur  $S^h$  bestimmt. Um das Finden wenn schon nicht der "wahren"<sup>16</sup>, so doch einer möglichst guten Netzstruktur soll es in dieser Arbeit letztenendes gehen. Andersherum ausgedrückt wollen wir eine Netzstruktur finden, die es erlaubt die Parameter so zu bestimmen, daß unser Netz gute Prognosen liefert. Die Suche nach der Netzstruktur ist also von der Schätzung der Parameter nicht wirklich zu trennen, zumindest muß die Frage, wie die Parameter bestimmt werden bei der Struktursuche berücksichtigt werden. Daher beginnen wir mit der Schätzung der Parameter, woraus sich dann die Berechnungsformel, mit der wir Netzstrukturen bewerten werden ableiten läßt.

### 4.4 Grundlage der Schätzung

Für beide Fragestellungen, der Suche nach den Parametern einerseits, als auch der anschließenden Suche nach der Netzstruktur andererseits benötigen wir eine Grundlage auf die wir unsere Schätzung stützen können. Diese Grundlage soll zum einen durch **Vorwissen** über die Netzstruktur und die Parameter und zum anderen durch neues Wissen gebildet werden, das uns in Form eines **Datensatzes**  $D$  zugänglich wird. Das Vorwissen kann z.B. von Experten gegeben werden, wir werden aber auch Verfahren diskutieren mit einem Minimum an, bzw. ohne Vorwissen auszukommen.

#### 4.4.1 Der Datensatz

Was den Datensatz betrifft gehen wir ja davon aus, daß die Merkmale, die wir in ein Bayes'sches Netz integrieren, **beobachtbar** sind. D.h. wir wollen auch davon ausgehen, daß der Datensatz  $D$  alle diese Merkmale enthält. Desweiteren sollen auch alle Merkmale in jedem in  $D$  enthaltenen Fall beobachtet worden sein.

**Def.:** *Einen Datensatz der diese Bedingungen erfüllt nennen wir **vollständig**.*

Es gibt Methoden um auch mit unvollständigen Daten umzugehen. Dabei können Merkmale in einzelnen oder auch in allen Fällen unbeobachtet geblieben sein. Wenn ein Merkmal in allen Fällen nicht beobachtet wurde so handelt sich um eine im obigen Sinne unbeobachtbare Variable. Bei teilweise nicht beobachteten Merkmalen muß noch unterschieden werden zwischen den Möglichkeiten, daß das Nichtbekanntsein der Ausprägung einer

---

<sup>16</sup>Da ein Bayes'sches Netz wie alle Modelle eine der Problemstellung angemessene Vereinfachung der Wirklichkeit darstellt, kann ohnehin bestenfalls eine im Sinne der Fragestellung und unter den Restriktionen des Modells optimale Lösung gefunden werden.

Variable in einzelnen Fällen abhängig<sup>17</sup> bzw. unabhängig von der tatsächlichen Ausprägung ist. Bei unbeobachtbaren Variablen liegt immer der unabhängige Fall vor; dagegen ist der abhängige Fall wesentlich schwieriger zu handhaben. Trotzdem gehören beide Fälle einem weiteren großen Feld der Statistik an und sollen hier nicht weiter behandelt werden.

## 4.5 Bayes'sche Statistik

Nicht festgelegt ist neben der Anzahl der Merkmale, die in ein Netz aufgenommen werden auch die "Größe" des Datensatzes - also die Anzahl der Fälle die er enthält.

**Def.:** *Dieses Kennzeichen bezeichnen wir als **Beobachtungsumfang** des Datensatzes.*

Bei der Verwendung der Daten soll im Sinne der **Bayes'schen Statistik** nicht außer acht gelassen werden, daß die Schätzung eines Parameters umso sicherer ist je mehr Beobachtungen der Schätzung zugrundeliegen. Dieser Ansatz ist von besonderer Bedeutung, da je größer ein Datensatz ist umso geringer sollten sich zufällige Schwankungen bemerkbar machen (Schwaches Gesetz der großen Zahlen) und umso besser sind auch - rein numerisch - gegebene Wahrscheinlichkeiten darstellbar.

Es geht dabei insbesondere darum eine zu starke Anpassung (*overfitting*) des Netzes an die i.d.R. durch zufällige Störungen verzerrten Daten zu vermeiden. Im Gegensatz z.B. zur linearen (quadratischen...) Regression, in der das Maß der möglichen Anpassung an die Daten bereits mit der Modellwahl festgelegt wird, ist diese Anpassung in einem Bayes'schen Netz eben durch die Netzstruktur weitgehend variabel, worin eine Stärke aber auch eine Gefahr liegt. Man möge beachten, daß mit  $S^h$  insbesondere auch die Anzahl der zu bestimmenden Parameter ( $= \text{Dimension}(S^h)$ ) gegeben ist. Je mehr Parameter es aber gibt umso geringer wird deren Fundierung durch Beobachtungen sein.

Wie sich zeigen wird kann dieses Prinzip der Gewichtung von Wissen durch Beobachtungsumfänge nicht nur auf den Teil der Schätzung, der sich auf die Daten  $D$  bezieht angewendet werden, sondern in äquivalenter Weise auch auf das Vorwissen über  $S^h$  und die Parameter  $\theta$ . Denn auch das Vorwissen, ob von Experten vorgegeben oder z.B. auch selbst aus zuvor erhaltenen Daten gewonnen, bezieht sich i.d.R. weitestgehend auf eine Wissensbasis aus gemachten Beobachtungen über den fraglichen Sachverhalt<sup>18</sup>. Durch diese Form der Betrachtung und der sich daraus ableitenden Vorgehensweise werden wir eine, im Sinne unserer Berechnungen, homogene Wissensbasis aus Vorwissen und Daten erhalten.

## 5 Parameterschätzung

### 5.1 Grundlagen der Parameterschätzung

Da letztendlich die Parameter (die bedingten Wahrscheinlichkeiten) ein Bayes'sches Netz bestimmen und die Struktur des Netzes die Anzahl der Parameter bestimmt, beschäftigen wir uns, wie angekündigt zunächst mit der Schätzung der Parameter, bevor wir daraus einen sinnvollen Weg zur Bestimmung der Art und Anzahl der Parameter also der Netzstruktur an sich ableiten.

Belassen wir es noch einen Moment bei einer getrennten Betrachtung von Vorwissen und Daten, so ist zunächst von Interesse wie die Parameter nur aus den Daten  $D$  bestimmt werden können - das Vorwissen liegt ja ohnehin in der Form von Parametern eines Bayes'schen Netzes, also in bedingten Wahrscheinlichkeiten vor. Um die Parameter zu bestimmen benötigen wir die Verteilung der Parameter bei gegebenen Daten<sup>19</sup>:  $p(\theta|D)$ . Für den Moment lassen wir dabei neben dem Vorwissen auch noch außer Acht daß die Parameter selbst in Bayes'schen Netzen von der Netzstruktur abhängen und betrachten den Fall zunächst möglichst allgemein.

Um zu leichter bestimmbareren Termen zu gelangen zerlegen wir  $p(\theta|D)$  entsprechend der **Bayes-Regel**<sup>20</sup>:

$$p(\theta|D) = \frac{p(\theta)p(D|\theta)}{p(D)} \quad (6)$$

<sup>17</sup>liegen z.B. über die Stärke eines Erdbebens aus der dem Epizentrum nächstgelegenen Meßstation keine Meßdaten vor, so könnte dies eine Indikation für ein besonders schweres Beben sein, das auch die seismologischen Meßinstrumente in Mitleidenschaft gezogen hat.

<sup>18</sup>wir gehen auf ein Problem in diesem Zusammenhang noch im Abschnitt 9.1 ein

<sup>19</sup>Für die Parameterschätzung gibt es auch andere Ansätze, die wir am Ende dieses Kapitels kurz erörtern.

<sup>20</sup> $p(A|B) = \frac{p(A)p(B|A)}{p(B)}$

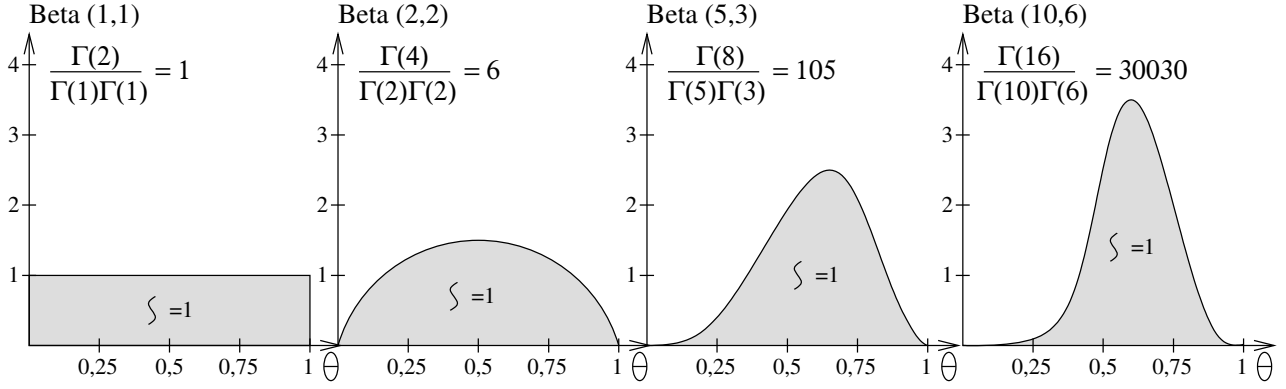


Abbildung 5: Einige beispielhafte Beta-Verteilungen

dabei sind	$p(D) = \int p(D \theta)p(\theta)d\theta$ d.h. $p(D)$ ist eine von $\theta$ unabhängige Normierungskonstante; auch <i>marginal Likelihood</i> genannt $p(\theta)$ die a-priori-Wahrscheinlichkeiten der Parameter $p(D \theta)$ die Likelihood der Daten bei gegebenen Parametern
------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Die einzelnen Teile der Gleichung werden wir nun betrachten.

## 5.2 Einstweilige Vereinfachung des Problems

Die Gleichung (6) ist sehr allgemein gehalten. Um uns unserer konkreten Fragestellung zu nähern - der Bestimmung der Parameter in einem Bayes'schen Netz - werden wir schrittweise die speziellen Bedingungen in unserer Problemstellung miteinbeziehen. Für eine erste Bestimmung der Likelihood  $p(D|\theta)$  und den a-priori-Wahrscheinlichkeiten der Parameter  $p(\theta)$  gehen wir daher zuerst von einer einzigen binomial verteilten Variable  $X$  mit den Ausprägungen  $X = \text{Kopf}$  und  $X = \text{Zahl}$  (Beispiel Münzwurf) aus, für deren Wahrscheinlichkeitsspezifikation ein einziger Parameter  $\theta$  ausreicht (die Gegenwahrscheinlichkeit ist durch  $1 - \theta$  gegeben).

Damit erhalten wir für die Likelihood:

$$p(D|\theta) = \theta^{\text{kopf}} (1 - \theta)^{\text{zahl}} \quad (7)$$

wobei	$\text{kopf} = N_{\text{kopf}}$ und $\text{zahl} = N_{\text{zahl}}$ die Anzahlen der beiden Ausprägungen $\text{Kopf}$ und $\text{Zahl}$ des Merkmals $X$ in $D$ darstellen.
-------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Indem wir die Produkte  $\theta^{\text{kopf}} = \prod_{n=1}^{\text{kopf}} \theta$  und  $(1 - \theta)^{\text{zahl}} = \prod_{n=1}^{\text{zahl}} (1 - \theta)$  bilden unterstellen wir - wie im weiteren auch immer - die **Unabhängigkeit** der einzelnen "Würfe" bzw. Beobachtungen. Auch dies stellt eine weitere Bedingung in unseren Annahmen - diesmal in Bezug auf den Datensatz  $D$  - dar. Allerdings sollte die Unabhängigkeitsannahme der einzelnen Beobachtungen für die meisten Fällen adäquat sein.

## 5.3 Die Beta-Verteilung

Die a-priori-Wahrscheinlichkeitsverteilung des Parameters  $p(\theta)$  ist, unter der für diesen Fall gebräuchlichen Annahme einer **Beta-Verteilung** gegeben durch:

$$p(\theta|\alpha_{\text{kopf}}, \alpha_{\text{zahl}}) = \text{Beta}(\theta|\alpha_{\text{kopf}}, \alpha_{\text{zahl}}) \equiv \frac{\Gamma(\alpha)}{\Gamma(\alpha_{\text{kopf}})\Gamma(\alpha_{\text{zahl}})} \theta^{\alpha_{\text{kopf}}-1} (1 - \theta)^{\alpha_{\text{zahl}}-1} \quad (8)$$

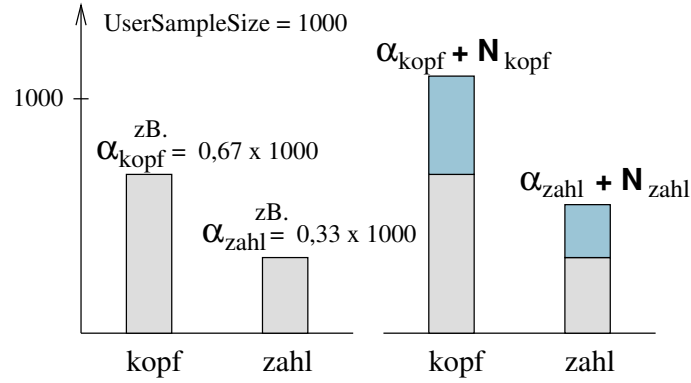


Abbildung 6: Gemeinsame gleichartige Betrachtung der Quantitäten  $N_{kopf}$ ,  $N_{zahl}$  und  $\alpha_{kopf}$ ,  $\alpha_{zahl}$ , die aus dem Vorwissen und den Daten gewonnen wurden

dabei sind	$\alpha_{kopf}$ und $\alpha_{zahl}$	die Hyperparameter der Beta-Verteilung es handelt sich dabei um Quantitäten äquivalent zu $kopf$ und $zahl$ , die aber nicht aus einem Datensatz gewonnen werden sondern das Vorwissen (i.d.R. vom Benutzer gegeben) widerspiegeln
	$\alpha = \alpha_{kopf} + \alpha_{zahl}$	der Beobachtungsumfang, aus dem das Vorwissen geschöpft wurde
	$\Gamma()$	die $\Gamma$ -Funktion siehe Abschnitt 5.4.1

Als a-priori bezeichnen wir diese Verteilung, weil wir nur von dem Vorwissen gegeben durch  $\alpha_{kopf}$  und  $\alpha_{zahl}$  ausgehen. Indem  $\alpha_{kopf}$  und  $\alpha_{zahl}$  das Vorwissen in der Form von Quantitäten äquivalent zu  $kopf$  und  $zahl$  wiedergeben sind wir bei der angekündigten äquivalenten Behandlung des Vorwissens und der Daten.

## 5.4 Gleichartige Betrachtung von Vorwissen und Daten

Wir hatten bereits gesehen, daß ein Datensatz neben Informationen über die Verteilungen der Merkmale ( $\rightarrow$  Parameter), durch den Beobachtungsumfang auf den sich ein Parameter gründet auch Aufschluß darüber gibt, mit welcher Sicherheit der geschätzte Wert des jeweiligen Paramteres angenommen werden kann. Dasselbe kann nun auch für die vom Benutzer erwarteten Wahrscheinlichkeiten für das Eintreten von  $X = Kopf$  bzw.  $X = Zahl$  realisiert werden.

**Def.:** Die Größe von  $\alpha = \alpha_{kopf} + \alpha_{zahl}$  spiegelt den Beobachtungsumfang eines (imaginären) Datensatzes  $D_{Vorwissen}$  wieder, aus dem der Benutzer sein Wissen bezogen hat. Die Werte für  $\alpha_{kopf}$  und  $\alpha_{zahl}$  können gewonnen werden indem das Vorwissen über die Wahrscheinlichkeiten  $p_{Vorwissen}(X = Kopf)$  und  $p_{Vorwissen}(X = Zahl)$  multipliziert wird mit einem Wert, der den Umfang dieses (imaginären) Datensatzes  $D_{Vorwissen}$  wiedergibt und im Folgenden als  $UserSampleSize$  bezeichnet wird.

Diese Technik ist als *equivalent sample size* ([Heckerman (March 1995)]) bekannt. Auf diese Weise erhalten wir eine Art Kontingenztafel korrespondierend zu  $D_{Vorwissen}$ .

Dabei sind die Begriffe “Kontingenztafel” und “Datensatz” nicht allzu wörtlich zu begreifen. Wie wir an Abbildung 7 erkennen können lassen wir zu, daß durch die Multiplikation  $UserSampleSize \cdot p_{Vorwissen}$  nicht unbedingt nur ganzzahlige Werte in der “ $\alpha$ -Kontingenztafel” entstehen, was wie im Beispiel der Abbildung 7 z.B. “halben Beobachtungen” in  $D_{Vorwissen}$  entspräche. Zwar wäre so etwas nicht möglich würden wir streng voraussetzen, daß sich das Vorwissen tatsächlich (ausschließlich) auf gemachte, zählbare Beobachtungen (also z.B. einzelne Erfahrungen eines Experten) gründet und die  $UserSampleSize$  auch genau diesen Beobachtungsumfang wiedergibt, jedoch wollen und müssen wir in diesem Punkt nicht derartig restriktiv verfahren. Der Experte soll sein Wissen durchaus intuitiv formulieren dürfen, in der Unterstützung genau dieser Möglichkeit durch die graphische Netzstruktur und den dazu korrespondierenden bedingten Wahrscheinlichkeiten liegt ja eine der Stärken Bayes’scher Netze, die wir uns an diesen Stellen nicht nehmen wollen, indem wir den Experten dazu zwingen sein Vorwissen über die Wahrscheinlichkeiten und den Wert für die  $UserSampleSize$

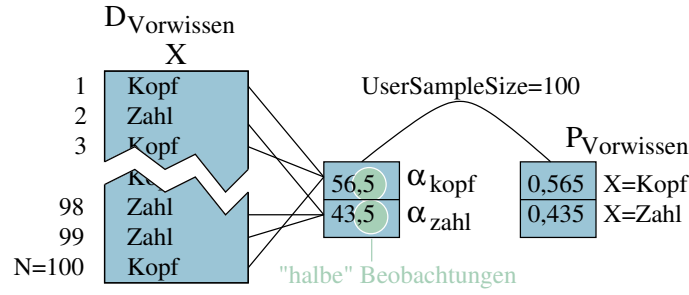


Abbildung 7: **Zusammenhang im Vorwissen** von: Wahrscheinlichkeiten, Hyperparameter  $\alpha$  und “Erfahrungsschatz”  $D_{Vorwissen}$  (über  $UserSampleSize$ )

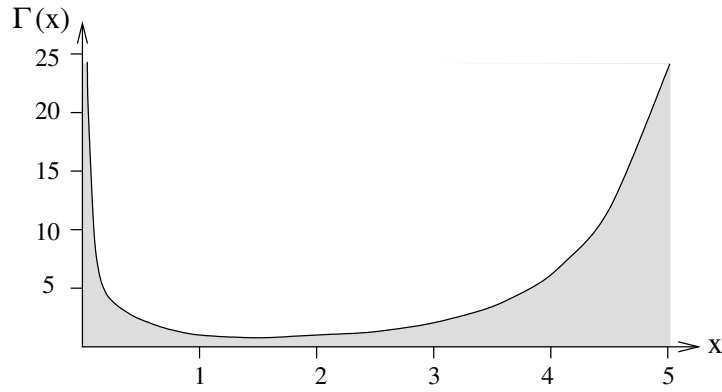


Abbildung 8:

$$\begin{aligned}\Gamma(x) &:= \lim_{n \rightarrow \infty} \frac{n! n^{x-1}}{x(x+1)(x+2)\dots(x+n-1)} & \forall x \notin \{0, -1, -2, -3, \dots\} \\ &= \int_0^\infty e^{-t} t^{x-1} dt & \forall x > 0\end{aligned}$$

exakt aufeinander abzustimmen. Man bedenke dabei, daß in einem realen Netz u.U. die Angabe sehr vieler Wahrscheinlichkeitswerte vom Benutzer verlangt wird. Aus diesem Umstand erklärt sich aber die Verwendung der  $\Gamma$ -Funktion.

#### 5.4.1 Die $\Gamma$ -Funktion

Für die  $\Gamma$ -Funktion gilt für natürliche Zahlen  $n$ :  $\Gamma(n) = (n-1)!$  jedoch ist die  $\Gamma$ -Funktion die stetige Fortsetzung von  $(n-1)!$ , also auch für alle Zwischenwerte erklärt.

### 5.5 Die Normierungskonstante der Beta-Verteilung

Betrachten wir, etwas vorgreifend die Beta-Verteilung mit Hinblick auf unser Vorhaben letztendlich keine Parameter, sondern Netzstrukturen zu bewerten etwas genauer. Dabei wollen wir gänzlich um die eigentliche Parameterschätzung umhinkommen. Es zeigt sich, daß in Gleichung (8) ein Teil separierbar ist, der  $\theta$  nicht enthält und als Normierungskonstante dient:

$$\frac{\Gamma(\alpha)}{\Gamma(\alpha_{kopf})\Gamma(\alpha_{zahl})} = c_{Beta}(\alpha_{kopf}, \alpha_{zahl}) \quad (9)$$

womit dann, wie für eine Verteilung gefordert gilt:

$$\int \frac{\Gamma(\alpha)}{\Gamma(\alpha_{kopf})\Gamma(\alpha_{zahl})} \theta^{\alpha_{kopf}-1} (1-\theta)^{\alpha_{zahl}-1} d\theta = 1$$



## 5.6 Konjugierte Verteilung

Eine **konjugierte Verteilung** liegt vor, wenn die Verteilungs-**Art** (in unserem Fall die Beta-Verteilung) erhalten bleibt, auch wenn Wissen (hier in der Form des Datensatzes  $D$ ) hinzukommt.

Unter der Annahme einer konjugierten Verteilung können wir auch den linken Teil der Gleichung (6) direkt berechnen:

$$\begin{aligned}
 p(\theta|D) &= \text{Beta}(\theta|\alpha_{kopf} + kopf, \alpha_{zahl} + zahl) \\
 &\equiv \frac{\Gamma(\alpha + N)}{\Gamma(\alpha_{kopf} + kopf)\Gamma(\alpha_{zahl} + zahl)} \theta^{\alpha_{kopf} + kopf - 1} (1 - \theta)^{\alpha_{zahl} + zahl - 1} \quad (10)
 \end{aligned}$$

|

wobei  $N$  der Umfang des Datensatzes  $D$  ist  
 $N_{kopf}$  bzw.  $N_{zahl}$  die Anzahl der Fälle darstellt, in  
denen  $X = Kopf$  bzw.  $X = Zahl$   
in  $D$  beobachtet wurde.

|

Äquivalent zu Gleichung (8) läßt sich auch in Gleichung (10) eine Normierungskonstante identifizieren:

$$\frac{\Gamma(\alpha + N)}{\Gamma(\alpha_{kopf} + kopf)\Gamma(\alpha_{zahl} + zahl)} = c_{Beta}(\alpha_{kopf} + kopf, \alpha_{zahl} + zahl) \quad (11)$$

## 5.7 Bestimmung der *marginalen Likelihood*

Tragen wir die Teile der Gleichung (6) zusammen, die wir bestimmen können:

$$p(\theta): \quad \text{Gleichung (8)} \qquad p(D|\theta): \quad \text{Gleichung (7)} \qquad p(\theta|D): \quad \text{Gleichung (10)}$$

Durch Einsetzen in Gleichung (6) und Umformen nach der *marginalen Likelihood*  $p(D)$  erhalten wir:

$$\begin{aligned}
 p(D) &= \frac{p(\theta)p(D|\theta)}{p(\theta|D)} \\
 p(D) &= \frac{\frac{\Gamma(\alpha)}{\Gamma(\alpha_{kopf})\Gamma(\alpha_{zahl})} \theta^{\alpha_{kopf} - 1} (1 - \theta)^{\alpha_{zahl} - 1} \theta^{kopf} (1 - \theta)^{zahl}}{\frac{\Gamma(\alpha + N)}{\Gamma(\alpha_{kopf} + kopf)\Gamma(\alpha_{zahl} + zahl)} \theta^{\alpha_{kopf} + kopf - 1} (1 - \theta)^{\alpha_{zahl} + zahl - 1}} \\
 p(D) &= \frac{\Gamma(\alpha)}{\Gamma(\alpha_{kopf})\Gamma(\alpha_{zahl})} \frac{\Gamma(\alpha_{kopf} + kopf) \Gamma(\alpha_{zahl} + zahl)}{\Gamma(\alpha + N)} \\
 p(D) &= \frac{\Gamma(\alpha)}{\Gamma(\alpha + N)} \frac{\Gamma(\alpha_{kopf} + kopf) \Gamma(\alpha_{zahl} + zahl)}{\Gamma(\alpha_{kopf}) \Gamma(\alpha_{zahl})} \quad (12) \\
 p(D) &= \frac{c_{Beta}(\alpha_{kopf}, \alpha_{zahl})}{c_{Beta}(\alpha_{kopf} + kopf, \alpha_{zahl} + zahl)}
 \end{aligned}$$

## 5.8 Schätzung der Parameter

Es ging uns bislang darum einen Weg der Parameterschätzung zu finden. Dazu benötigten wir zunächst die Verteilungsfunktion des (bisher einzigen) Parameters  $\theta$  bei gegebenen Daten  $p(\theta|D)$ . Unter der Annahme einer konjugierten Verteilung und der Annahme, einer Beta-Verteilung des Parameters konnten wir  $p(\theta|D)$  unter Einbeziehung auch des Vorwissens direkt berechnen (Gleichung (10)). Wir haben damit die Grundlagen für die Schätzung der Parameter zusammengetragen. Zur eigentlichen Schätzung kommen wir nun. Nach wie vor gehen wir von einer einzigen binomialverteilten Variable  $X$  aus. Jedoch wird sich zeigen, daß sich die Parameterschätzung leicht auf die weiteren verkomplizierenden Annahmen übertragen läßt, so daß wir die Frage mit diesem Kapitel abschließen können.

### 5.8.1 Vergleich des Bayes'schen und des Klassischen Ansatzes

Die Ansätze bei der Parameterschätzung in der klassischen und der Bayes'schen Statistik unterscheiden sich grundlegend - sind praktisch gegenteilig in ihren Grundideen.

In der **klassischen Statistik** wird der Parameter als gegeben festgehalten und (formal) alle Datensätze betrachtet, die durch sampling<sup>21</sup> aus diesem Parameter entstanden sein könnten; obgleich eigentlich der “richtige”, noch unbekannte Parameter zu den beobachteten und damit gegebenen Daten gesucht ist. Ein häufig verwendeter Schätzer ist der **Maximum-Likelihood-Schätzer**, der  $p(D|\theta)$  maximiert.

Dagegen werden in der **Bayes'schen Statistik** die Daten als gegeben hingenommen und die Wahrscheinlichkeitsverteilung des unbekannten Parameters betrachtet ( $p(\theta|D)$ ).

Über die Vor- und Nachteile der beiden Ansätze herrscht Uneinigkeit. Der Bayes'sche Ansatz hat zumindest den Vorteil intuitiver zu sein. Wir werden uns daher weiter an die Bayes'sche Statistik halten.

### 5.8.2 Allgemeine Form der Bayes'schen Parameterschätzung

Der nun zu schätzende Parameter  $\theta$  dient uns dazu die Wahrscheinlichkeit mit der das Ereignis  $X_{N+1} = Kopf$  eintritt abzuschätzen, nachdem wir über Vorwissen und neues Wissen in Form der Daten  $D$  verfügen. Allgemein gilt für beliebige Verteilungen:

$$\begin{aligned} p(X_{N+1} = \text{Kopf} | D) &= \int p(X_{N+1} = \text{Kopf} | \theta) p(\theta | D) d\theta \\ &= \int \theta p(\theta | D) d\theta \\ &\equiv E_{p(\theta | D)}(\theta) \end{aligned} \tag{13}$$

wobei	$X_{N+1}$	die Ausprägung von $X$ im nächsten noch nicht beobachteten Fall darstellt, nachdem uns mit $D$ $N$ Fälle bekannt sind
	$E_{p(\theta D)}(\theta)$	den Erwartungswert von $\theta$ unter Berücksichtigung der Verteilung $p(\theta D)$ wiedergibt

### 5.8.3 Schätzung unter der Annahme einer Beta-Verteilung

Da für die *Beta*-Verteilung gilt:

$$\int \theta \text{Beta}(\theta | \alpha_{kopf}, \alpha_{zahi}) d\theta = \frac{\alpha_{kopf}}{\alpha} \quad (14)$$

erhalten wir für die Gleichung (13) und damit als Schätzer des Parameters:

$$p(X_{N+1} = K \text{opf} | D) = \frac{\alpha_{k \text{opf}} + k \text{opf}}{\alpha + N} = \theta \quad (15)$$

Oder wenn wir ohne Daten beginnen und unser Wissen schrittweise durch Beobachtungen erweitern:

$$p(X_1 = K\text{opf}) = \frac{\alpha_{kopf}}{\alpha} \quad p(X_2 = K\text{opf} | X_1 = K\text{opf}) = \frac{\alpha_{kopf} + 1}{\alpha + 1} \quad (16)$$

#### 5.8.4 Erkenntnisse

Was daran gezeigt werden soll ist folgendes:

- Erstens entspricht das so gewonnene Verfahren zur Parameterbestimmung wohl der intuitiven Erwartung. Der Parameter wird “einfach” durch das Verhältnis der Beobachtungen mit  $X = Kopf$  zum Gesamtbeobachtungsumfang gewonnen.
- Zweitens erkennt man, daß auf die eigentliche Parameterschätzung der Beobachtungsumfang keinen Einfluß hat. Da ein Parameter ein einzelner Wert, der eine Wahrscheinlichkeit wiedergibt ist, sind weitere Informationen, wie eben zB. über die Sicherheit mit der dieser Parameter gegeben ist auch nicht darstellbar. Wir werden nun daher für die Bestimmung der Netzstruktur zu Berücksichtigen haben, daß die Beobachtungsumfänge (Daten und Vorwissen) nicht unter den Tisch fallen.

<sup>21</sup>künstliche Erzeugung von Daten, wie sie beobachtet werden könnten, unter der Annahme, daß die Parameter die wahre Wahrscheinlichkeitsverteilung widerspiegeln

- Zuletzt sei noch angemerkt, daß für die Verteilung des Parameters - die Beta-Verteilung - von Bedeutung ist, daß wir die Wahrscheinlichkeit für das Eintreten eines zukünftigen Ereignisses aus dem bis dahin zur Verfügung stehenden Wissen (Vorwissen und Daten) bestimmen. Die minimale Grundlage auf die sich eine Schätzung gründen kann sind aber bisher zwei Beobachtungen (je eine für  $\alpha = Kopf$  und  $\alpha = Zahl$ ). Daher die “-1” die in der *Beta*-Verteilung mehrfach auftaucht:

$$Beta(\theta|\alpha_{kopf}, \alpha_{zahl}) \equiv \frac{\Gamma(\alpha)}{\Gamma(\alpha_{kopf})\Gamma(\alpha_{zahl})} \theta^{\alpha_{kopf}-1} (1-\theta)^{\alpha_{zahl}-1}$$

wobei für natürliche Zahlen  $n$  gilt:  $\Gamma(n) = (n-1)!$

### 5.8.5 Parameterschätzung in Bayes’schen Netzen (volle Modellannahme)

Vorrausschauend in Bezug auf die Schätzung der bedingten Wahrscheinlichkeitstafeln zu den Knoten eines Bayes’schen Netzes können wir hier bereits festhalten, daß sich an der einfachen Form der Schätzung von Parametern durch die Erweiterung auf den multinomialen, multivariaten Fall nichts grundsätzlich ändern wird; einzig die benötigten Objekte (Tafeln) und damit die Berechnungen werden komplexer. Die Schätzung der bedingten Wahrscheinlichkeitstafeln und damit der Parameter erfolgt im wesentlichen in vier Schritten:

1. Ableitung der Kontingenztafeln zu jedem Knoten und seinen Eltern aus den Daten  $D$ .
2. Ableitung der Tafeln, die das Vorwissen wiedergeben, äquivalent zu der Form in der die Kontingenztafeln das Wissen aus den Daten darstellen, aus den benutzergegebenen bedingten Wahrscheinlichkeitstafeln und einem *UserSampleSize*-Wert (auf diesen Punkt wird im weiteren noch eingegangen, da diese Tafeln auch zur Netzstrukturbestimmung erstellt werden müssen, was aber nicht ganz trivial ist (siehe Abschnitt 6.8)).
3. Addition der Kontingenz- und Vorwissentafel, die jeweils zu einem Knoten gehören.
4. Normierung dieser Gesamt- (Vorwissen- und Daten-) tafeln zu bedingten Wahrscheinlichkeitstafeln.

Da die Parameterschätzung nicht eigentliches Thema dieser Arbeit ist, sei das Thema hiermit beendet. Das dargestellte Verfahren entspricht weitgehend dem in [Heckerman (March 1995)] beschrieben.

## 6 Netzstrukturbewertung

Wir verlassen nun die Parameterschätzung und werden uns dem Ziel der Bestimmung einer guten Netzstruktur, gegeben Daten und Vorwissen schrittweise nähern. Um Netzstrukturen vergleichen zu können brauchen wir eine Bewertungsformel. Zur Herleitung einer solchen Formel müssen wir unsere bislang möglichst einfach gehaltenen Annahmen verkomplizieren. Dazu wird eine Erweiterung der *Beta*-Verteilung auf den multinomialen Fall benötigt. Diese ist durch die *Dirichlet*-Verteilung gegeben, die dann noch auf den multivariaten Fall angepaßt werden muß, um ein Bayes’sches Netz nach unseren Vorgaben beschreiben zu können. Aus dieser Parameter-Verteilung soll dann eine Netzbewertungsformel abgeleitet werden, bei der die Schätzung der Parameter ausgeklammert werden soll. Für die Verwendung dieser Formel wird Vorwissen in einer bestimmten Form benötigt. Um die Herleitung dieses Vorwissens aus der normalen Form eines Bayes’schen Netzes, wie es von einem Experten vorgegeben werden kann, soll es im letzten Schritt gehen.

### 6.1 Die Dirichlet-Verteilung (multinomiale Variablen)

Bislang sind wir von einer einzigen binomial verteilten Zufallsvariable  $X$  ausgegangen. Nun kann  $X$  aber auch **multinomial** sein, d.h. mehr als zwei Ausprägungen haben, womit nun statt der Beta- die **Dirichlet-Verteilung** angenommen werden kann:

$$p(\theta) = Dir(\theta|\alpha_1, \dots, \alpha_r) \equiv \frac{\Gamma(\alpha)}{\prod_{k=1}^r \Gamma(\alpha_k)} \prod_{k=1}^r \theta_k^{\alpha_k-1} \quad (17)$$

Dabei sind	$r$	die Anzahl der möglichen (mit positiver Wahrscheinlichkeit auftretenden) Ausprägungen von $X$
	$\theta$	ein Satz von Parametern (statt dem bisher einzigen Parameter $\theta$ )
	$\theta_k$	der Parameter, der die Wahrscheinlichkeit der $k$ -ten Ausprägung wiedergibt, wobei wir wie bereits erläutert mit $r - 1$ Parametern auskommen - der Einfachheit halber soll mit $\theta_r$ aber die mit $1 - \sum_{k=1}^{r-1} \theta_k$ gegebene Wahrscheinlichkeit bezeichnet werden
	$\alpha_k$	gegeben durch: $\alpha_k = p_{Vorwissen}(X = k) UserSampleSize$ ähnlich der Beta-Verteilung gilt damit: $UserSampleSize = \alpha = \sum_{k=1}^r \alpha_k$ und $\alpha_k > 0$

## 6.2 Die Normierungskonstante der Dirichlet-Verteilung

Auch in der Dirichlet-Verteilung finden wir wieder, ähnlich wie in der Beta-Verteilung, eine Normierungskonstante:

$$\frac{\Gamma(\alpha)}{\prod_{k=1}^r \Gamma(\alpha_k)} = c_{Dir}(\alpha_1, \dots, \alpha_r) \quad (18)$$

Es gilt:

$$\int \prod_{k=1}^r \theta_k^{\alpha_k - 1} = \frac{1}{c_{Dir}(\alpha_1, \dots, \alpha_r)}$$

## 6.3 Der Multivariate Fall (volle Modellannahme)

Weiterhin enthält ein Bayes'sches Netz i.d.R. natürlich mehr als eine Variable. Daher gehen wir nun von einem Satz von (multinomialen) Zufallsvariablen  $\mathbf{X}$  aus.

Im Folgenden sei	$\mathbf{X}$	ein Satz von (multinomialen) Zufallsvariablen
	$X_i$	die $i$ -te der $n$ Variablen
	$X_i^k$	die $k$ -te von $r_i$ Ausprägungen der $i$ -ten Variable
	$S^h$	eine Netzstruktur in der Form eines DAG's über die $X_i$
	$\mathbf{Pa}_i$	der (u.U. leere) Satz von Elternvariablen v. $X_i$
	$Pa_i^j$	die $j$ -te von $q_i$ möglichen Konfigurationen der Elternausprägungen

In diesem Fall sind für jeden Knoten bei gegebener Elternzustandskonfiguration die Dirichlet-Verteilungen der Parameter, die die geschätzten bedingten Wahrscheinlichkeiten jedes Knotenzustands bei gegebener Elternkonfiguration wiedergeben, bestimmt durch:

$$p(\theta_{ij} | \mathbf{Pa}_i, S^h) = Dir(\theta_{ij} | \alpha_{ij1}, \dots, \alpha_{ijr_i}) \equiv \frac{\Gamma(\alpha_{ij})}{\prod_{k=1}^{r_i} \Gamma(\alpha_{ijk})} \prod_{k=1}^{r_i} \theta_{ijk}^{\alpha_{ijk} - 1} \quad (19)$$

Dabei ist	$r_i$	die Anzahl der möglichen (mit positiver Wahrscheinlichkeit auftretenden) Ausprägungen von $X_i$
	$\theta_{ij}$	der Satz von Parametern zum Knoten $X_i$ , bei gegebener Elternkonfiguration $\mathbf{Pa}_i = j$
	$\theta_{ijk}$	der Parameter, der die Wahrscheinlichkeit der $k$ -ten Ausprägung von $X_i$ bei gegebener $j$ -ten Elternzustandskonfiguration wiedergibt; wiederum sei $\theta_{ijr_i} = 1 - \sum_{k=1}^{r_i-1} \theta_{ijk}$
	$\alpha_{ijk}$	Das Vorwissen über das Eintreten des $k$ -ten Zustands bei gegebener $j$ -ten Elternkonfiguration in der Form von Quantitäten vergleichbar zu Beobachtungsanzahlen in einem Datensatz; Dieses Vorwissen wird bestimmt durch: $\alpha_{ijk} = p_{Vorwissen}(X = k, \mathbf{Pa}_i = j   S^h) \cdot UserSampleSize$
	$\alpha_{ij}$	die marginale Tafel gegeben durch: $\alpha_{ij} = \sum_k \alpha_{ijk}$

Unter der Annahme einer *konjugierten Verteilung* erhalten wir bei Einbeziehung eines Datensatzes  $D$ :

$$p(\theta_{ij} | \mathbf{Pa}_i, S^h) = Dir(\theta_i | \alpha_{ij1}, \dots, \alpha_{ijr_i}) \equiv \frac{\Gamma(\alpha_{ij} + N_{ij})}{\prod_{k=1}^{r_i} \Gamma(\alpha_{ijk} + N_{ijk})} \prod_{k=1}^{r_i} \theta_{ijk}^{\alpha_{ijk} + N_{ijk} - 1} \quad (20)$$

Hierbei ist	$N_{ijk}$	die aus dem Datensatz erhaltenen Anzahl von Beobachtungen in denen $X_i = k$ und $\mathbf{Pa}_i = j$
jedes	$N_{ij}$	die marginale Tafel $\sum_k N_{ijk}$
	$N_i$	kann als eine von der Netzstruktur $S^h$ abhängige Kontingenztafel zum Knoten $X_i$ gesehen werden
entsprechend ist	$\alpha_i$	die Vorwissen-Kontingenztafel zu $X_i$

Die Normierungskonstanten für die Fälle vor bzw. nach Erhalt der Daten  $D$  lauten:

$$\frac{\Gamma(\alpha_{ij})}{\prod_{k=1}^{r_i} \Gamma(\alpha_{ijk})} \quad \text{bzw.} \quad \frac{\Gamma(\alpha_{ij} + N_{ij})}{\prod_{k=1}^{r_i} \Gamma(\alpha_{ijk} + N_{ijk})} \quad (21)$$

## 6.4 Allgemeine Form der Netzbewertung

Über die bedingten Abhängigkeiten zwischen den Variablen, repräsentiert durch eine Netzstruktur über  $\mathbf{X}$ , sollen Hypothesen  $S^h$  mit einer Wahrscheinlichkeit belegt werden. Diese Wahrscheinlichkeit soll anhand des Vorwissens und der Daten  $D$  berechnet werden. Gesucht ist also  $p(S^h | D)$ .

Dem Bayes'schen Theorem folgend ergibt sich:

$$p(S^h | D) = \frac{p(S^h)p(D|S^h)}{p(D)} \quad (22)$$

es sind	$p(D)$	eine Normierungskonstante, die
(vgl. Gleichung (6))	$p(S^h)$	nicht von der Netzstruktur abhängt
	$p(D S^h)$	die a-priori-Wahrscheinlichkeit der Hypothese $S^h$ über die Netzstruktur
		die Likelihood der Daten gegeben die Netzstruktur

## 6.5 Herleitung der Netzbewertungsformel

Die a-priori-Wahrscheinlichkeit der Netzstruktur wird zunächst keine weitere Rolle spielen - wir werden ganz im Bayes'schen Sinne unser Unwissen über diese Wahrscheinlichkeit modellieren, indem wir alle Hypothesen vorab (a-priori) als gleichwahrscheinlich annehmen<sup>22</sup>. Somit bleibt  $p(D|S^h)$  allein ausschlaggebend für die

<sup>22</sup> Auch wenn durch diese, in der Bayes'schen Statistik allgemein übliche Annahme in Extremfällen Paradoxien erzeugt werden können und dieses Verfahren daher teilweise sehr kritisch gesehen wird.

Bewertung einer Netzhypothese  $S^h$ . Man beachte die Ähnlichkeit zur Likelihood - es handelt sich um eine Art Likelihood der Daten gegeben die Netzstruktur. Sie kann berechnet werden durch:

$$\begin{aligned} p(D|S^h) &= \int p(D, \theta|S^h) d\theta \\ &= \int p(D|\theta, S^h) p(\theta|S^h) d\theta \end{aligned} \quad (23)$$

Das wäre, um noch einmal auf unser einfachstes Beispiel einer binomial verteilten Variable  $X$  zurückzukommen (in dem es auch nur eine mögliche Netzhypothese  $S^h$  gibt):

$$\begin{aligned} p(D|S^h) &= \int \theta^{k_{opf}} (1-\theta)^{z_{ahl}} \theta^{\alpha_{kopf}-1} (1-\theta)^{\alpha_{zahl}-1} c_{Beta}(\alpha_{kopf}, \alpha_{zahl}) d\theta \\ p(D|S^h) &= c_{Beta}(\alpha_{kopf}, \alpha_{zahl}) \int \theta^{k_{opf}+\alpha_{kopf}-1} (1-\theta)^{z_{ahl}+\alpha_{zahl}-1} d\theta \\ p(D|S^h) &= \frac{c_{Beta}(\alpha_{kopf}, \alpha_{zahl})}{c_{Beta}(\alpha_{kopf} + k_{opf}, \alpha_{zahl} + z_{ahl})} \\ p(D|S^h) &= \frac{\Gamma(\alpha)}{\Gamma(\alpha_{kopf})\Gamma(\alpha_{zahl})} \frac{\Gamma(\alpha_{kopf} + k_{opf})\Gamma(\alpha_{zahl} + z_{ahl})}{\Gamma(\alpha + N)} \end{aligned} \quad (24)$$

Dieses Ergebnis entspricht der *marginalen Likelihood* aus Gleichung (12), da bei nur einer möglichen Netzstruktur  $S^h$  gilt:  $p(D|S^h) = p(D)$ .

Durch die Erweiterung auf eine multinomiale Variable und die Verwendung der Normierungskonstanten der Dirichlet-Verteilung  $c_{Dir}$ , anstatt der  $c_{Beta}$ -Konstanten erhalten wir:

$$p(D|S^h) = \frac{\Gamma(\alpha)}{\Gamma(\alpha + N)} \prod_{k=1}^r \frac{\Gamma(\alpha_k + N_k)}{\Gamma(\alpha_k)} \quad (25)$$

$$\left| \begin{array}{ll} \text{wobei} & N_k \text{ die Anzahl der Beobachtungen} \\ & \text{im Datensatz } D \text{ ist, mit } X = k \end{array} \right|$$

## 6.6 Bewertung der Netzstruktur

Die Gleichung (25) muß nun im letzten Schritt noch auf den multivariaten, multinomialen Fall - also die Grundvoraussetzung eines Bayes'schen Netzes, wie wir es annehmen - erweitert werden. Damit ergibt sich nun eine Bewertungsformel für Netzstrukturen, wie sie zuerst in [Cooper & Herskovits (1992)] beschrieben wurde:

$$p(D|S^h) = \prod_i \prod_j \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \prod_k \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})} \quad (26)$$

$$\left| \begin{array}{lll} \text{Es sind die} & N_{ijk} & \text{die Anzahlen der Beobachtungen in } D \text{ in denen} \\ & & X_i = k \text{ und } \mathbf{Pa}_i = j \\ & N_{ij} & \text{die Werte der marginalen Tafeln } \sum_k N_{ijk} \\ & \alpha_{ijk} & \text{das Vorwissen über die Häufigkeit des} \\ & & \text{ Auftretens von } X_i = k \text{ u. } \mathbf{Pa}_i = j, \\ & \alpha_{ij} & \text{die Werte der marginalen Tafeln } \sum_k \alpha_{ijk} \end{array} \right|$$

## 6.7 Bewertung von Abhängigkeiten

Die Annahme der Abhängigkeit zweier Werte ist ein symmetrisches Konzept. Gibt die Information, daß schlechtes Wetter herrscht, Information darüber her welcher Prozentteil der Menschen einen Schirm mit sich führt, so gibt auch die Information über den Anteil der Schirmträger Aufschluß über das momentane Wetter. Daher sollte auch die **Bewertung** von Abhängigkeiten identisch sein für die Fälle:

$$\text{Wetter} \rightarrow \text{Schirmmitnahme} \quad \text{und} \quad \text{Schirmmitnahme} \rightarrow \text{Wetter}.$$

In der nun eingeführten Bewertungsformel (Gleichung (26)) ist die gleiche Bewertung der beiden Fälle gewährleistet.

**Bsp.:** Nehmen wir an, zwei Merkmale  $A$  und  $B$  sind laut Vorwissen unabhängig. Das Vorwissen über die Verteilungen von  $A$  und  $B$  sei gegeben durch:

$p_{V_{orw}}(A, B)$				$UserSampleSize = 128$			
				$\alpha_{i=B, j \in \{a_1, a_2\}, k} = \alpha_{i=B, j \in \{b_1, b_2\}, k}$			
		$A$		$p_{V_{orw}}(B)$		$A$	
		$a_1$	$a_2$			$a_1$	$a_2$
$B$	$b_1$	0.125	0.375	0.5	$\cdot 128 =$	16	48
	$b_2$	0.125	0.375	0.5		16	48
$p_{V_{orw}}(A)$	$\sum$	0.25	0.75			32	96

Desweiteren sei ein Datensatz gegeben, der das Vorwissen zu 100% bestätigt und im Beobachtungsumfang der  $UserSampleSize$  entspricht, so daß gilt:  $N_{ijk} = \alpha_{ijk}$ . Damit erhalten wir für  $p(D|S^h)$  (nach Ersetzen von  $\Gamma(x)$  durch  $(x-1)!$ ):

$$\begin{aligned} \left( \frac{127!}{63! \cdot 63!} \cdot \frac{127! \cdot 127!}{255!} \right) \cdot \left( \frac{63!}{15! \cdot 47!} \cdot \frac{31! \cdot 95!}{127!} \cdot \frac{63!}{15! \cdot 47!} \cdot \frac{31! \cdot 95!}{127!} \right) &\approx 5.7 \cdot 10^{-71} \quad S^h = B \rightarrow A \\ \left( \frac{127!}{31! \cdot 95!} \cdot \frac{63! \cdot 191!}{255!} \right) \cdot \left( \frac{31!}{15! \cdot 15!} \cdot \frac{31! \cdot 31!}{63!} \cdot \frac{95!}{47! \cdot 47!} \cdot \frac{95! \cdot 95!}{191!} \right) &\approx 5.7 \cdot 10^{-71} \quad S^h = A \rightarrow B \\ \left( \frac{127!}{31! \cdot 95!} \cdot \frac{63! \cdot 191!}{255!} \right) \cdot \left( \frac{127!}{63! \cdot 63!} \cdot \frac{127! \cdot 127!}{255!} \right) &\approx 8.1 \cdot 10^{-71} \quad S^h = A \quad B \end{aligned}$$

Das Netz, daß die Unabhängigkeit wiedergibt erhält tatsächlich die höchste Bewertung, die beiden anderen Netze aber unterscheiden sich in ihren Bewertungen nicht. Auch wenn jeweils 16 gleiche Beobachtungen (Vorwissen und Daten) zu einer zusammengefaßt werden ( $\alpha'_{ijk} := \alpha_{ijk}/16$ ;  $N'_{ijk} := N_{ijk}/16$ ) ändert sich nichts Grundsätzliches:

$$\begin{aligned} p(D|S^h) &\approx 0.0000228 \quad S^h = B \rightarrow A \\ p(D|S^h) &\approx 0.0000228 \quad S^h = A \rightarrow B \\ p(D|S^h) &\approx \mathbf{0.0000286} \quad S^h = A \quad B \end{aligned}$$

Entsprechendes gilt auch für andere mit einem Faktor aus den  $\alpha$ -Tafeln erhaltbaren Kontingenztafeln.

Anders verhält es sich, wenn der Datensatz je 16 Beobachtungen für jede der Ausprägungskombinationen  $A = a_1, B = b_1$  und  $A = a_2, B = b_1$  und je 48 Beobachtungen für  $A = a_1, B = b_2$  und  $A = a_2, B = b_2$  enthält, was ebenfalls auf die Unabhängigkeit von  $A$  und  $B$  hindeuten würde.

Dann ergibt die Berechnung von  $p(D|S^h)$ :

$$\begin{aligned} \left( \frac{127!}{63! \cdot 63!} \cdot \frac{95! \cdot 159!}{255!} \right) \cdot \left( \frac{63!}{15! \cdot 47!} \cdot \frac{31! \cdot 63!}{95!} \cdot \frac{63!}{15! \cdot 47!} \cdot \frac{63! \cdot 95!}{159!} \right) &\approx 2.6 \cdot 10^{-78} \quad S^h = B \rightarrow A \\ \left( \frac{127!}{31! \cdot 95!} \cdot \frac{95! \cdot 159!}{255!} \right) \cdot \left( \frac{31!}{15! \cdot 15!} \cdot \frac{31! \cdot 63!}{95!} \cdot \frac{95!}{47! \cdot 47!} \cdot \frac{63! \cdot 95!}{159!} \right) &\approx 2.6 \cdot 10^{-78} \quad S^h = A \rightarrow B \\ \left( \frac{127!}{63! \cdot 63!} \cdot \frac{95! \cdot 159!}{255!} \right) \cdot \left( \frac{127!}{31! \cdot 95!} \cdot \frac{95! \cdot 159!}{255!} \right) &\approx 2.2 \cdot 10^{-78} \quad S^h = A \quad B \end{aligned}$$

Nun werden die Netze  $A \rightarrow B$  bzw.  $B \rightarrow A$  gegenüber der Unabhängigkeitsannahme bevorzugt. Der Datensatz bestätigt hier nicht mehr das Vorwissen, obwohl sowohl Vorwissen als auch Daten einzeln auf Unabhängigkeit hindeuten. Trotzdem ist dieses Ergebnis nicht überraschend, da eine gemeinsame Tafel  $\alpha_{ijk} + N_{ijk}$  durchaus eine Abhängigkeit erkennen läßt. Unterschiedliche (Un-)Abhängigkeitsannahmen führen also u.U. zur Annahme einer Abhängigkeit. Bei konkreten Berechnungen können allerdings Rundungsfehler zu leicht unterschiedlichen Ergebnissen, bei eigentlich äquivalenten Netzen führen, da einzelnen Werte differieren bzw. in unterschiedlichen Reihenfolgen auftreten.

Wir haben nun gesehen, wie unterschiedliche Unabhängigkeitsannahmen zusammengenommen auf eine Abhängigkeit schließen lassen. Auch das Gegenteil kann eintreten, d.h. unterschiedliche Abhängigkeitsannahmen können sich gegenseitig eliminieren, wie natürlich auch verstärken (bestätigen). Um diesen Zusammenhang zu illustrieren bewerten wir zwei Netze durch  $f_1(x, y) = p(D|A \rightarrow B)$  und  $f_2(x, y) = p(D|A \quad B)$  entsprechend den abgebildeten Tafeln.

Es ist zu erkennen, wie die im Vorwissen enthaltene Abhängigkeit durch  $x = 2.5$  und  $y = 0.5$  aufgehoben wird, so daß das kantenlose Netz favorisiert wird, während mit  $x = 0.5$  und  $y = 2.5$  die Abhängigkeit bestätigt wird und so das zugehörige Netz ( $A \rightarrow B$ ) bevorzugt wird.

$\alpha$

		A		
		$a_1$	$a_2$	$\Sigma$
B	$b_1$	2	4	6
	$b_2$	4	2	6
$\Sigma$		6	6	

$N$

		A		
		$a_1$	$a_2$	$\Sigma$
B	$b_1$	x	y	x+y
	$b_2$	3-x	3-y	6-x-y
$\Sigma$		3	3	

f1(x,y) —  
f2(x,y) - - -

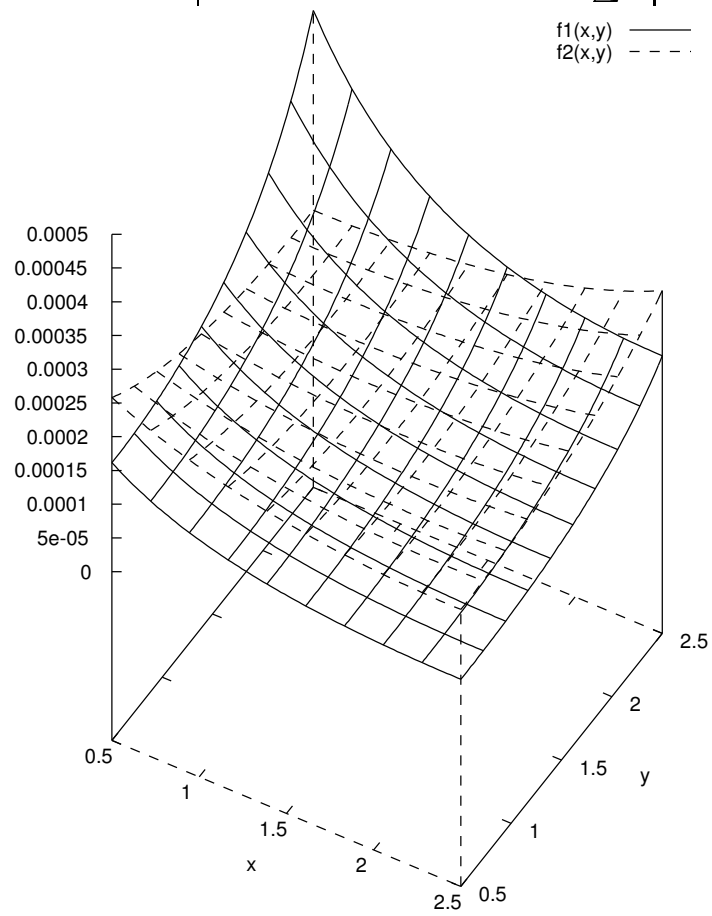


Abbildung 9: Plot von Netzbewertungen mit gnuplot



## 6.8 Rücksicht auf die Parameterschätzung Und Vermeidung von *overfitting*

Wie an Gleichung (26) zu erkennen ist müssen wir, zur Bestimmung einer Netzstruktur unter Verwendung dieser Gleichung, die Parameter  $\theta$  selbst nicht bestimmen. Es ist jedoch wichtig festzuhalten, daß die bislang als Normierungskonstanten bezeichneten Teile der Dirichlet-Verteilung, aus denen Gleichung (26) zusammengesetzt ist zwar die Parameter  $\theta$  nicht enthalten jedoch von deren Struktur, gegeben durch die Netzstruktur  $S^h$  und damit auch nicht zuletzt von deren Anzahl abhängen (je mehr Parameter zu schätzen sind desto kleiner werden die Beobachtungsanzahlen sowohl im imaginären Datensatz  $D_{Vorwissen}$  als auch im wirklichen Datensatz  $D$ , auf die sich die Schätzung gründet). Im Gegensatz zur Parameterschätzung geht hier also tatsächlich in starkem Maße der Beobachtungsumfang in die Berechnung ein. Es wird damit gewährleistet, daß jeder Parameter, der nach der Bestimmung der Netzstruktur zu bestimmen wäre, auch durch einen ausreichenden Beobachtungsumfang fundiert werden kann. Dieser Umstand ist besonders im Sinne der Vermeidung von *Overfitting* - also der Überschätzung des zugrundeliegenden Wissens - hervorzuheben.

Abhängigkeiten sind letzten Endes Ungleichgewichte der  $\alpha_{ijk}$  und der  $N_{ijk}$  gegenüber den Werten, die Unabhängigkeit bedeuten würden:  $\frac{\alpha_{ij}}{r_i}$  bzw.  $\frac{N_{ij}}{r_i}$ . In dem Maße, wie einzelne  $\alpha_{ijk}$  und  $N_{ijk}$  unter diesem Richtwert liegen, überschreiten ihn die restlichen Werte. Da aber  $\Gamma(x)$  mit steigendem  $x$  immer schneller wächst, wird der Wert von  $\frac{\Gamma(\alpha_{ij})}{\prod_k \Gamma(\alpha_{ijk})}$  für eine gegebene Elternzustandskonfiguration  $j$  je stärker die Abhängigkeit von  $X_i$  und seinen Elternknoten  $\mathbf{Pa}_i$  ist immer kleiner werden. Noch stärker aber wird der Wert von  $\frac{\prod_k \Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ij} + N_{ij})}$  dank der durch den Datensatz vergrößerten einzelnen Parameter der  $\Gamma$ -Funktion wachsen, sofern sich die Ungleichgewichte der  $\alpha_{ijk}$  und  $N_{ijk}$  gegenseitig noch verstärken und nicht etwa aufheben.

D.h.:  $|(\alpha_{ijk} + N_{ijk}) - \frac{\alpha_{ij} + N_{ij}}{r_i}| \geq |(\alpha_{ijk} - \frac{\alpha_{ij}}{r_i}) + (N_{ijk} - \frac{N_{ij}}{r_i})| \forall j, k$

Damit wird der Gesamtwert umso größer je stärker die Abhängigkeit von  $X_i$  und seinen Elternknoten  $\mathbf{Pa}_i$  ist. Halten wir dagegen Vorwissen ( $\alpha$ ) und Datensatz ( $N$ ) und damit die Grundlage zur Bestimmung der Abhängigkeiten als gegeben fest, so wachsen mit der Vergrößerung von  $\mathbf{Pa}_i$ , also der Hinzunahme weiterer Elternknoten zum Knoten  $X_i$  die Anzahl der  $\alpha_{ijk}$  und der  $N_{ijk}$  jeweils um den Faktor  $r_e$  (Anzahl der Ausprägungen von  $X_e$ ) für einen weiteren Elternknoten  $X_e$ . Damit schrumpfen zwangsläufig die einzelnen Werte der  $\alpha_{ijk}$  und  $N_{ijk}$ , da die Gesamtsumme jeweils gleich bleibt ( $\sum_{j,k} \alpha_{ijk} = UserSampleSize$ ,  $\sum_{j,k} N_{ijk} = \text{Beobachtungsumfang von } D$ ). Damit kann aber u.U. trotz aus den Daten (scheinbar) ableitbarer Abhängigkeit von  $X_i$  zum neuen Elternknoten die Bewertung für das Netz ohne die entsprechende Kante höher liegen als für das Netz mit dieser zusätzlichen Kante. Das ist eben dann der Fall, wenn die einzelnen Werte und damit auch die einzelnen Ungleichgewichte zu gering werden, als daß sie durch das Wachstum gegenüber der Unabhängigkeitsannahme die Verkleinerung der Einzelwerte (durch die Vergrößerung der Tafeln) ausgleichen könnten. Umfangreiche Berechnungen zu diesem Sachverhalt werden noch in Abschnitt 8.2 und Kapitel 10, insbesondere im Abschnitt 10.4 durchgeführt.

## 6.9 Gewinnung der $\alpha$ - (Vorwissen-) Tafeln

Von besonderem Interesse ist wie die  $\alpha_{ijk}$ -(Vorwissen)-Tafeln zu erhalten sind. Während die  $N_{ijk}$ -(Kontingenz)-Tafeln "einfach"<sup>23</sup> mittels Durchzählen der Beobachtungen, die zu jeder  $N_{ijk}$ -Zelle gehören, erhalten werden, müssen das Vorwissen bzw. die Hyperparameter der Dirichlet-Verteilung aus den bedingten Wahrscheinlichkeiten die z.B. von einem Experten korrespondierend zu einer ebenfalls von ihm vorgegebenen Netzstruktur formuliert worden sind und dem bereits eingeführten *UserSampleSize*-Wert gewonnen werden.

Es sei mit	$p_{Vorw}$	allgemein das Vorwissen über Wahrscheinlichkeitsverteilungen bezeichnet
------------	------------	-------------------------------------------------------------------------

Sind zu einer Netzstruktur  $S^h$  die **gemeinsamen** Wahrscheinlichkeitstafeln für jeden Knoten und seine Eltern aufgestellt, die das Vorwissen über die Verteilung widerspiegeln und gibt man dazu einen *UserSampleSize*-Wert vor, der einen Gewichtungsfaktor dieses Vorwissen darstellt, so ließen sich die  $\alpha_{ijk}$  (für die initiale Netzstruktur  $S^h$ ) mithilfe der Technik der *equivalent sample size* ([Heckerman (March 1995)], siehe auch Abschnitt 5.4) noch einfach berechnen durch:

$$\alpha_{ijk} = UserSampleSize \cdot p_{Vorw}(X_i = k, Pa_i = j | S^h) \quad (27)$$

Das Vorwissen, daß durch ein bestehendes Netz gegeben ist, liegt jedoch i.d.R. in **bedingten** Wahrscheinlichkeitstafeln der Form  $p_{Vorw}(X_i^k | Pa_i^j, S^h)$  vor, wobei wir zunächst davon ausgehen, daß die zu testenden

<sup>23</sup>wenn auch bei großen Datensätzen rechenintensiv

Netzhypothese  $S^h$  die bereits bestehende Netzstruktur ist, zu der die bedingten Wahrscheinlichkeitstabellen aufgestellt wurden. Aus den bedingten Wahrscheinlichkeitstabellen der einzelnen Knoten gegeben ihre Eltern ließe sich eine vollständige Wahrscheinlichkeitstabelle aller Knoten gewinnen indem alle bedingten Wahrscheinlichkeitstabellen miteinander multipliziert würden:

$$p_{V_{orw}}(\mathbf{X}|S^h) = \prod_i p_{V_{orw}}(X_i|\mathbf{Pa}_i, S^h) \quad (28)$$

*Die Tafel  $p_{V_{orw}}(\mathbf{X}|S^h)$  kann damit, obwohl i.d.R. ungleich größer als alle bedingten Wahrscheinlichkeitstabellen zusammen, natürlich nur (exakt) soviel Information über (Un-) Abhängigkeiten enthalten wie durch  $S^h$  und die bedingten Tabellen  $p_{V_{orw}}(X_i|\mathbf{Pa}_i, S^h)$  gegeben war.*

Damit sind die gesuchten gemeinsamen Wahrscheinlichkeitstabellen  $p_{V_{orw}}(X_i, \mathbf{Pa}_i|S^h)$  zu berechnen durch:

$$p_{V_{orw}}(X_i, \mathbf{Pa}_i|S^h) = \sum_{\mathbf{X} \setminus \{X_i, \mathbf{Pa}_i\}} p_{V_{orw}}(\mathbf{X}|S^h) \quad (29)$$

also jede vollständige Wahrscheinlichkeitstabelle marginalisiert über alle Merkmale außer dem jeweiligen Knoten  $X_i$  selbst und dessen Eltern  $\mathbf{Pa}_i$ .

Dieser Ansatz ist obwohl in der Theorie einfach im allgemeinen unberechenbar aufwendig, da  $p(\mathbf{X}|S^h)$  mit einigen Knoten schnell sehr groß wird. Wir wollen diesen Weg aber als Referenz für unseren weiteren Überlegungen im Hinterkopf behalten.

Die korrekten gemeinsamen Wahrscheinlichkeitstabellen für eine Knotenmenge, wie hier für einen Knoten und seine Eltern kann wie in Kapitel 3 beschrieben über den Junction-Tree-Algorithmus berechnet werden (für Knoten + Elternknoten gab es jeweils mindestens eine *Home-Clique*, es können aber auch die Tabellen zu anderen Knotenmengen aus dem Junction-Tree abgeleitet werden).

Jedoch ist auch dieses Verfahren relativ aufwendig, wenn sich die Netzstruktur ständig ändert und deshalb jedesmal ein neuer Junction-Tree aufgestellt werden müsste.

Um weniger aufwendig die gemeinsamen Wahrscheinlichkeitstabellen für jeden Knoten und seine Eltern zu erhalten vernachlässigen wir Abhängigkeiten, wie sie durch Zyklen entstehen. Nachträgliche Versuche mit über Junction-Trees gewonnene "perfekte" Tabellen haben gezeigt, daß sich praktisch keine Unterschiede ergaben und dieser Ansatz daher gerechtfertigt ist.

Für diesen Ansatz werden die gemeinsamen Wahrscheinlichkeitstabellen der Eltern und deren Eltern benötigt. Das Problem wird also auf die Elterntabellen verlagert, es ergibt sich eine rekursive Berechnungsstruktur.

An dem Punkt, an dem die Tabelle zu einem beliebigen Knoten  $X_i$  berechnet wird ergibt sich jedoch eine entscheidende Vereinfachung:

$$\begin{aligned} p_{V_{orw}}(X_i, \mathbf{Pa}_i|S^h) &= p_{V_{orw}}(X_i|\mathbf{Pa}_i, S^h) \prod_{pa \in \mathbf{Pa}_i} \left( \sum_{\mathbf{X} \setminus \{X_i, \mathbf{Pa}_i\}} \frac{p_{V_{orw}}(pa, \mathbf{Pa}_{pa}|S^h)}{\sum_{pa} p(pa, \mathbf{Pa}_{pa}|S^h)} \right) \\ &= p_{V_{orw}}(X_i|\mathbf{Pa}_i, S^h) \prod_{pa \in \mathbf{Pa}_i} \left( \sum_{\mathbf{X} \setminus \{X_i, \mathbf{Pa}_i\}} \frac{p_{V_{orw}}(pa, \mathbf{Pa}_{pa}|S^h)}{p(\mathbf{Pa}_{pa}|S^h)} \right) \\ &= p_{V_{orw}}(X_i|\mathbf{Pa}_i, S^h) \prod_{pa \in \mathbf{Pa}_i} (p(pa|\mathbf{Pa}_{pa} \cap \mathbf{Pa}_i, S^h)) \end{aligned} \quad (30)$$

$\mathbf{Pa}_{pa}$ $(\bigcup_{pa \in \mathbf{Pa}_i} = \text{Menge der Großeltern von } X_i)$ $\mathbf{Pa}_{pa} \cap \mathbf{Pa}_i$	sind die Elternknoten eines Elternknotens $pa$ von $X_i$ aus $\mathbf{Pa}_i$ sind nur die Knoten aus $\mathbf{Pa}_{pa}$ , die auch in $\mathbf{Pa}_i$ sind
------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------

Die Vereinfachung - im Sinne einer effizienten Berechnung durch Kleinhaltung der Tabellen - liegt in der Vorab-Marginalisierung der Elterntabellen über Nicht-Elternmerkmale. Nur Elternmerkmale sind in der Tabelle zu  $X_i$  dargestellt. Damit kann aber über sonstige Merkmale, die nur in den Elterntabellen auftreten, vorab marginalisiert werden.

*Es gilt für zwei mehrdimensionale Tabellen  $P$  und  $Q$  allgemein:*

$$Q \cdot \sum_v P = \sum_v (P \cdot Q) \quad (31)$$

P			Q			P Q	
			q1	q2	q3		
$\downarrow$ v $\downarrow$	p11	p12	p13	q1 p11	q2 p12		q3 p13
	p21	p22	p23	q1 p21	q2 p22		q3 p23
	p31	p32	p33	q1 p31	q2 p32		q3 p33
$\sum_v$	p11+p21+p31	p12+p22+p32	p13+p23+p33	q1 ( p11+p21+p31 )	q2 ( p12+p22+p32 )	q3 ( p13+p23+p33 )	

$\sum_v P Q = Q \sum_v P$

Abbildung 10: Darstellung zu Gleichung (31)

*Damit ist die Reihenfolge der Marginalisierung in einem solchen Fall irrelevant.*

|    **wenn**     $r_v^Q = 1$     d.h. Dimension  $v$  ist nicht in  $Q$  repräsentiert    |

Wie an Gleichung (30) zu erkennen ist erstellen wir für die Berechnung jedes Knotens  $X_i$  ein Bayes'sches Teil-Netz bestehend aus dem Sub-Graphen zu  $X_i$  und den Elternknoten  $\mathbf{Pa}_i$  sowie den zugehörigen bedingten Wahrscheinlichkeitstafeln zu diesem Sub-Graphen. Durch Ausmultiplizieren dieser Tafeln erhält man die gesuchte gemeinsame Wahrscheinlichkeitstafel für  $X_i$  und  $\mathbf{Pa}_i$ . Benötigt werden aber die gemeinsamen Wahrscheinlichkeitstafeln für jeden  $pa \in \mathbf{Pa}_i$  und dessen Eltern  $\mathbf{Pa}_{pa}$ .

*Ein DAG enthält jedoch mindestens einen Knoten ohne Eltern (wie auch mind. einen ohne Kinder), so daß für diese(n) Knoten direkt die gesuchte Wahrscheinlichkeitstafel  $p(X_i, \mathbf{Pa}_i | S^h)$  mit  $\mathbf{Pa}_i = \emptyset$  - also der a-priori-Wahrscheinlichkeit -  $p(X_i | S^h)$  gegeben ist.*

Von solchen Knoten ausgehend können zunächst die Tafeln von deren Kindern berechnet werden, danach die Tafeln der Kindeskinder ("Enkel") usw., bis alle Knotentafeln ermittelt wurden. Man bedenke in diesem Zusammenhang auch:

*Unverbundene Teil-DAG's innerhalb eines DAG's bilden wieder ein DAG - wie **jeder** Sub-Graph eines DAG's wieder ein DAG ist - in diesem Fall aber bei Trennung der Teil-Graphen die bedingten Wahrscheinlichkeitstafeln identisch denen des Gesamt-DAG's sind und somit die Berechnungsschritte auf solche Teil-DAG's getrennt angewendet werden können, ohne Veränderungen der bedingten Wahrscheinlichkeitstafeln erklären zu müssen.*

## 7 Netzstruktursuche

### 7.1 Schrittweise Verbesserung der Netzstruktur ( "Greedy-Search" )

Nachdem wir in der Lage sind ein vorgegebenes Netz zu bewerten, müssen wir nun ein Verfahren entwickeln ein solches Netz mit veränderten Strukturen zu vergleichen, um eine verbesserte Struktur zu finden.

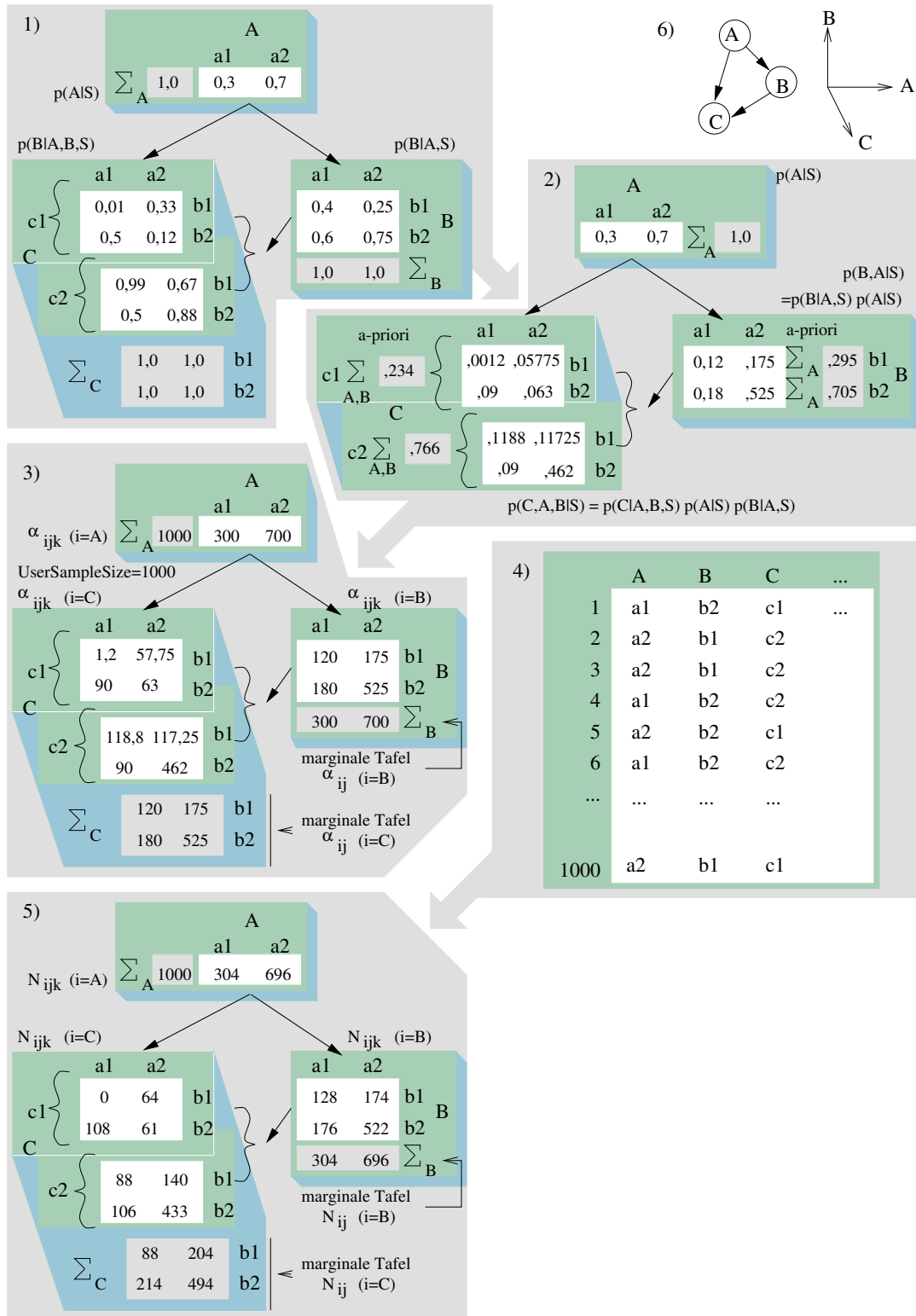


Abbildung 11: Berechnungsstruktur mit Bsp.-Werten 1) bedingte Wahrscheinlichkeiten (Vorwissen) 2) gemeinsame Wahrscheinlichkeiten f.  $X_i$  u.  $\mathbf{Pa}_i$  3)  $\alpha_{ijk}$  ("Vorwissen-Kontingenztafeln") entsprechend  $UserSampleSize = 1000$  4) Datensatz  $D$  5)  $N_{ijk}$  (Kontingenztafeln) aus  $D$  zur Netzstruktur  $S$  6) Netzstruktur  $S$  und Dimensionsaufteilung von  $A$ ,  $B$  und  $C$  in den Tafeln

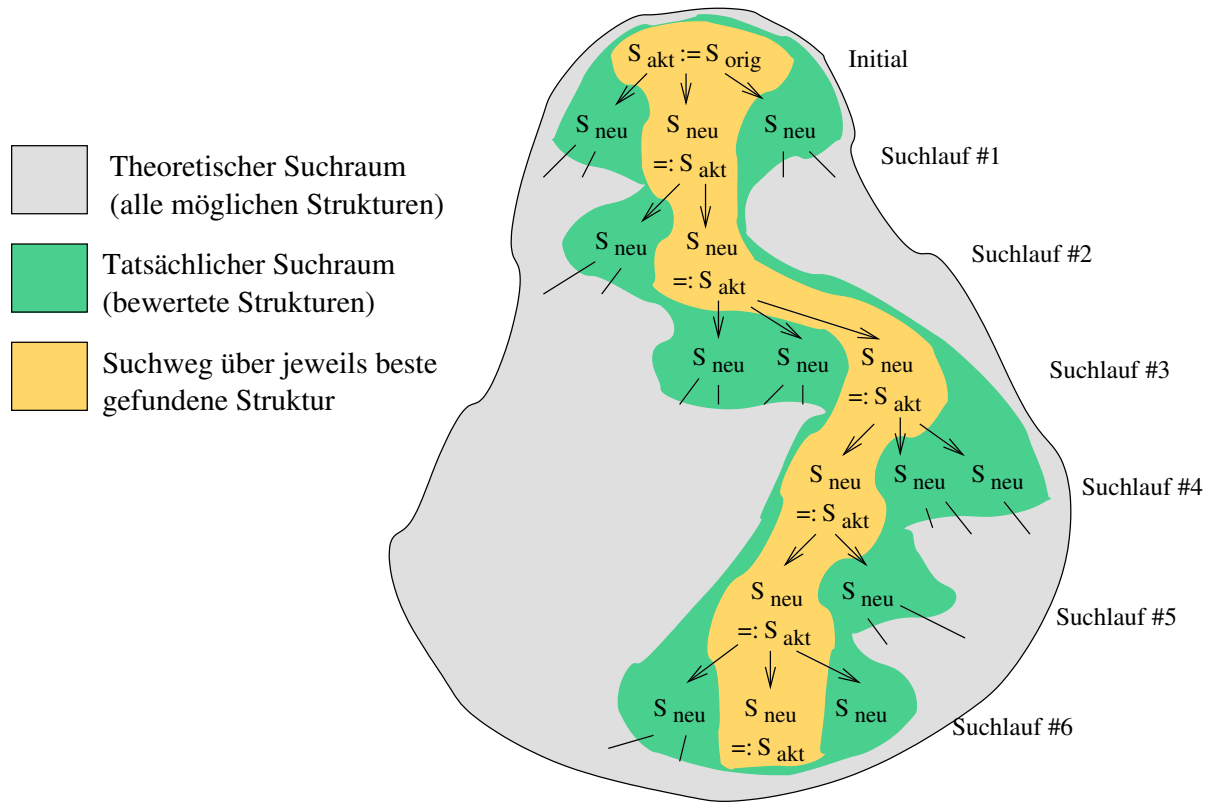


Abbildung 12: Netzstruktursuche mit **Greedy-Search**

Wir bezeichnen mit	$S_{orig}^h$ $S_{neu}^h$	das ursprünglich gegebene Netz eine zu testende abweichende Struktur
	$S_{akt}^h$	eine zuvor als $S_{neu}^h$ getestete und da verbessert angepaßte Netzstruktur, von der ausgehend weitere Veränderungen getestet werden (initial ist $S_{akt}^h := S_{orig}^h$ )
	$N'_{ijk}$ ( $N'_{ij}$ )	die aus $D$ abgeleiteten Kontingenztafeln zu $S_{neu}^h$
	$\alpha'_{ijk}$ ( $\alpha'_{ij}$ )	die aus dem Vorwissen zu $S_{orig}^h$ abgeleiteten $\alpha$ -Tafeln zu $S_{neu}^h$
	$p_{Vorw}$	allgemein das Vorwissen über Wahrscheinlichkeiten
	$p_{orig}, p_{neu}, p_{akt}$	das zu speziellen Netzstrukturen ( $S_{orig}^h, S_{neu}^h, S_{akt}^h$ ) aus $p_{Vorw}$ abgeleitete Vorwissen

Das Verfahren, das wir entwickeln werden besteht nun darin ein gegebenes Netz sukzessive durch Veränderungen (Hinzufügen bzw. Löschen) einzelner Kanten zu verbessern, bis (auf diese Weise) keine weitere Verbesserung mehr erreicht werden kann. Ein solches Suchverfahren heißt **Greedy-Search** [Heckerman (March 1995)]. Eine einzelne Kantenänderungen hat, wie leicht zu erkennen ist, nur Einfluß auf die Elternmenge eines einzelnen Knotens  $X_i$ . Beim Hinzufügen einer Kante wird die Elternmenge eines Knotens um einen Elterknoten erweitert, beim Löschen entsprechend vermindert.

## 7.2 Effiziente Berechnung

Im Sinne einer effizienten Berechnung besonders hervorzuheben ist, daß die Berechnungsformel für  $p(D|S^h)$  (Gleichung (26)) **separabel**, dh. für jeden Knoten  $X_i$  getrennt berechenbar ist.

Der Wert für einen Knoten bei einer gegenüber der vorhergehenden Netzstruktur  $S_{akt}^h$  geänderten Netzhypothese  $S_{neu}^h$  muß nur neu berechnet werden, wenn es in der Elternmenge dieses Knotens Veränderungen gegeben hat.

Für den Fall, daß sich in einer neuen Netzstruktur  $S_{neu}^h$  nur Veränderungen in der Elternmenge  $\mathbf{Pa}_d$  eines einzelnen Knotens  $X_d$  ergeben haben, kann die Bewertung für  $S_{neu}^h$  berechnet werden durch:

$$p(D|S_{neu}^h) = \frac{p(D|S_{akt}^h) \prod_j \left( \frac{\Gamma(\alpha'_{dj})}{\Gamma(\alpha'_{dj} + N'_{dj})} \prod_k \frac{\Gamma(\alpha'_{djk} + N'_{djk})}{\Gamma(\alpha'_{djk})} \right)}{\prod_j \left( \frac{\Gamma(\alpha_{dj})}{\Gamma(\alpha_{dj} + N_{dj})} \prod_k \frac{\Gamma(\alpha_{djk} + N_{djk})}{\Gamma(\alpha_{djk})} \right)} \quad (32)$$

$$\left| \begin{array}{ll} p(D|S_{akt}^h) & \text{ist das Produkt der Bewertungen der einzelnen Knoten } X_i, \\ & \text{das gegeben ist durch: } \prod_i \prod_j \left( \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \prod_k \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})} \right) \\ \prod_j \left( \frac{\Gamma(\alpha_{dj})}{\Gamma(\alpha_{dj} + N_{dj})} \prod_k \frac{\Gamma(\alpha_{djk} + N_{djk})}{\Gamma(\alpha_{djk})} \right) & \text{ist die "alte" Bewertung zum Knoten } X_d \text{ in } S_{akt}^h \\ \prod_j \left( \frac{\Gamma(\alpha'_{dj})}{\Gamma(\alpha'_{dj} + N'_{dj})} \prod_k \frac{\Gamma(\alpha'_{djk} + N'_{djk})}{\Gamma(\alpha'_{djk})} \right) & \text{ist die geänderte Bewertung zum Knoten } X_d \text{ in } S_{neu}^h \end{array} \right|$$

Denn es gilt allgemein:  $\frac{y'_p \cdot \prod_{p=1}^n y_p}{y_q} = \prod_{p=1}^{q-1} y_p \cdot y'_q \cdot \prod_{p=q+1}^n y_p$ , wobei hier  $y_p$  für die Bewertung des Knotens  $X_p$  steht und  $y'_q$  für die neue von  $y_p$  abweichende Bewertung des Knotens  $X_q$  steht (mit  $p, q \in \{1, \dots, n\}$  und  $n$  = Anzahl der Knoten). Da die Gesamtbewertung der alten Netzstruktur mit  $p(D|S_{akt}^h) = \prod_{p=1}^n y_p$ , sowie der alte Einzelwert des Knotens  $X_q$  mit  $y_p$  bereits vorliegen, reduziert sich die Bewertung der neuen Netzstruktur auf die Neubewertung des Knotens  $X_q$  durch den Wert  $y'_q$  und eine einfache Multiplikation und Division:  $p(D|S_{akt}^h) \cdot y'_q / y_q$ . Damit ist diese Berechnung wesentlich effizienter als eine gesamte Neuberechnung entsprechend Gleichung (26).

### 7.3 Veränderte Kontingenz- und Vorwissen-Tafeln

Es soll weiterhin auch zur Bewertung veränderter Netzstrukturen die Formel aus Gleichung (26) verwendet werden. Das verlangt aber die Gewinnung veränderter Kontingenz und  $\alpha$ -Tafeln.

Die neuen Kontingenztafeln ( $N'_i$ ) sind vom Verfahren her einfach (aber u.U. rechenintensiv) erneut aus den Daten abzuleiten.

Von größerer Problematik ist wiederum das Finden der Vorwissen-Tafeln ( $\alpha$ -Tafeln). So wie die neuen Kontingenztafeln entsprechend  $S_{neu}$  aus dem Datensatz  $D$  abgeleitet werden, so müssen die  $\alpha'$ -Tafeln nun ihrerseits das Vorwissen über die **neue** Netzstruktur  $S_{neu}^h$  wiedergeben, obwohl die gegebenen bedingten Wahrscheinlichkeitstafeln zu einem abweichenden Netz  $S_{orig}^h$  bzw.  $S_{akt}^h$  gehören. Die  $\alpha'$ -Tafeln müssen also entsprechend angepaßt werden. Dabei geht es mehr um eine technische Frage; wie an Gleichung (26) zu erkennen ist, müssen auch die  $\alpha$  ( $\alpha'$ )-Tafeln in ihrer Größe und Dimensionalität zu den Kontingenztafeln und beide zur Netzstruktur passen. Zusätzliche Information - wie das bei den Kontingenztafeln durchaus möglich und gewünscht ist - darf dabei in die  $\alpha'$ -Tafeln nicht einfließen, da es nur um eine andere Darstellung des **selben** Vorwissens geht.

Bezogen auf die gemeinsame Vorwissen-Wahrscheinlichkeitstafel ( $p_{Vorw}(\mathbf{X}|S_{orig}^h)$ ) soll nur eine andere marginale Wahrscheinlichkeitstafel für den jeweils in seiner Elternmenge veränderten Knoten erzeugt werden, um dann durch Multiplikation mit der *UserSampleSize* die  $\alpha'_i$ -Tafel zu erhalten.

$$p_{neu}(X_i, \mathbf{Pa}_i^{neu} | S_{neu}^h) = \sum_{\mathbf{X} \setminus X_i, \mathbf{Pa}_i^{neu}} p(\mathbf{X} | S_{orig}^h) \quad (33)$$

$$\alpha'_i = UserSampleSize \cdot p_{neu}(X_i, \mathbf{Pa}_i^{neu} | S_{neu}^h) \quad (34)$$

$$\left| \quad \mathbf{Pa}_i^{neu} \quad \text{ist die in } S_{neu}^h \text{ veränderte Elternmenge von } X_i \quad \right|$$

Allerdings muß auch hier wieder der Aufwand, dafür tatsächlich die evtl. sehr große gemeinsame Wahrscheinlichkeitstafel aufzustellen, vermieden werden.

Die Angaben  $p_{orig}$ ,  $p_{neu}$ ,  $p_{akt}$  bezeichnen also immer **dasselbe** Vorwissen; da wir mit diesen Bezeichnungen aber insbesondere auch die Form der mehrdimensionalen Tafeln meinen, in denen dieses Wissen repräsentiert ist, unterscheiden wir durch die verschiedenen Angaben die Größe und Form dieser Tafeln entsprechend den unterschiedlichen Netzstrukturen.

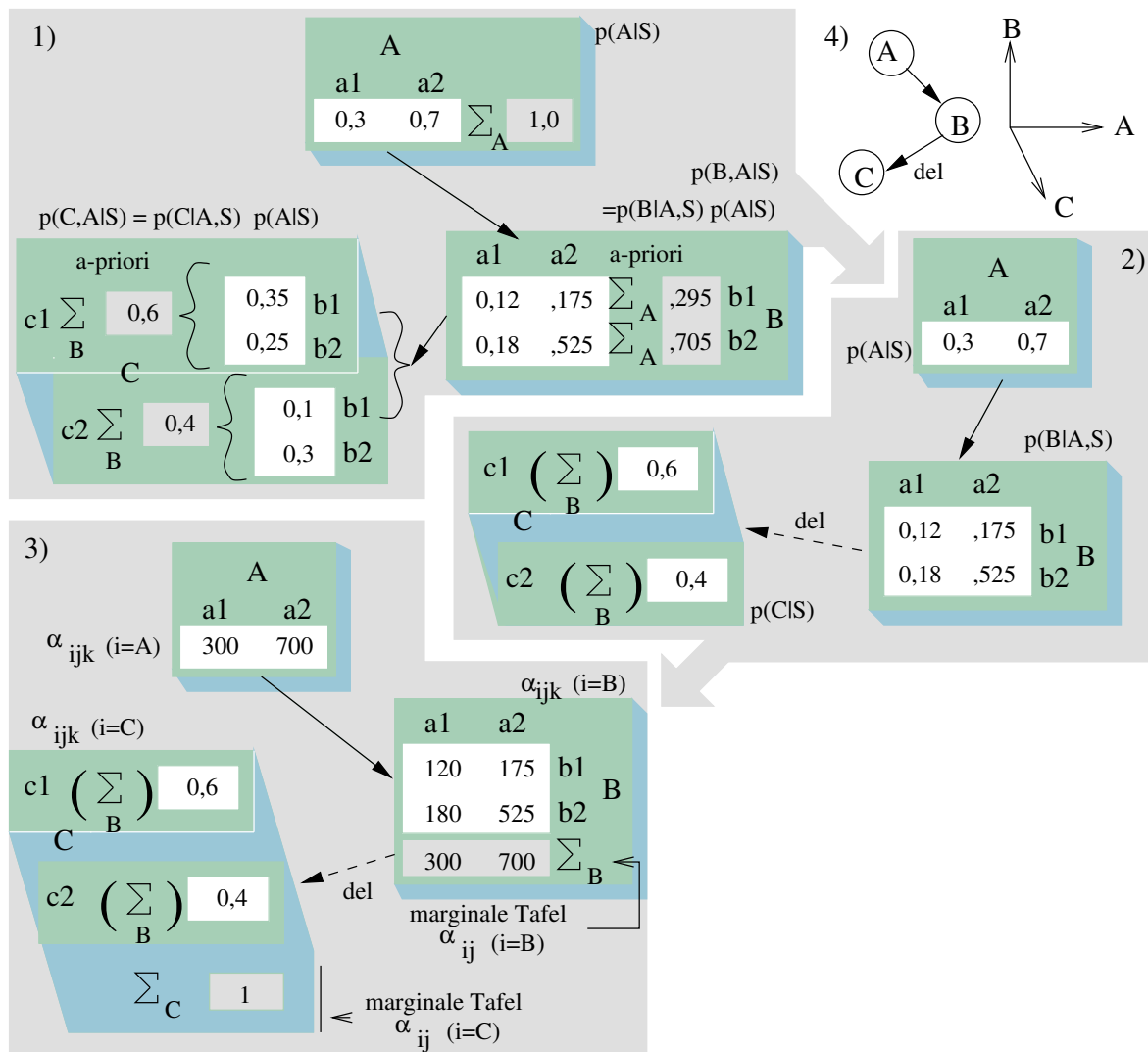


Abbildung 13: Berechnungsbeispiel für das Löschen einer Kante  $B \rightarrow C$

- 1) Gemeinsame Wahrscheinlichkeitstafeln von  $X_i$  und  $\mathbf{Pa}_i$  zu  $S_{akt}^h$  (Vorwissen)
- 2) Wsk.-Tafeln über nicht  $\mathbf{Pa}_C$  marginalisiert.
- 3)  $\alpha_{ijk}$  ("Vorwissen-Kontingenztafeln") entspr.  $UserSampleSize = 1000$
- 4) Struktur u. Dimensionsaufteilung

### 7.3.1 Entfernen einer Kante

Betrachten wir zunächst den Fall, daß gegenüber der ursprünglichen Annahme die Knoten  $\mathbf{Pa}_i^{\text{del}}$  aus  $\mathbf{Pa}_i$  entfernt wurden. Die neue Wahrscheinlichkeitstafel, zur Berechnung von  $\alpha_{ijk}$  erhält man durch:

$$p_{\text{neu}}(X_i, \mathbf{Pa}_i^{\text{neu}} | S_{\text{neu}}^h) = \sum_{\mathbf{Pa}_i^{\text{del}}} p_{\text{akt}}(X_i, \mathbf{Pa}_i^{\text{akt}} | S_{\text{akt}}^h) \left[ = \sum_{\mathbf{X} \setminus \{X_i, \mathbf{Pa}_i^{\text{neu}}\}} p(\mathbf{X} | S^h) \right] \quad (35)$$

$$\left| \begin{array}{l} \mathbf{Pa}_i^{\text{akt}} \text{ ist die Elternmenge von } X_i \text{ in } S_{\text{akt}}^h \\ \mathbf{Pa}_i^{\text{del}} \text{ ist die Menge der aus } \mathbf{Pa}_i^{\text{akt}} \text{ entfernten} \\ \text{Knoten } (\mathbf{Pa}_i^{\text{del}} = \mathbf{Pa}_i^{\text{akt}} \setminus \mathbf{Pa}_i^{\text{neu}}) \end{array} \right|$$

dh. es wird die ursprüngliche Tafel über die Merkmale marginalisiert, die nicht länger Elternknoten von  $X_i$  sind.

*Unter der Annahme, daß bereits die alte Wahrscheinlichkeitstafel  $p_{\text{akt}}(X_i, \mathbf{Pa}_i^{\text{akt}} | S_{\text{akt}}^h)$  so erzeugt wurde, daß sie der gesuchten marginalen Tafel von  $p_{\text{Vorw}}(\mathbf{X} | S^h)$  entsprach (!), ist mit dem beschriebenen Verfahren in natürlicher Weise gewährleistet, daß auch die neue Tafel der gesuchten marginalen Tafel von  $p_{\text{Vorw}}(\mathbf{X} | S^h)$  entspricht.*

### 7.3.2 Hinzufügen einer Kante

Neben dem Entfernen von Elternknoten zu einem Merkmal  $X_i$  können auch neue Elternknoten  $\mathbf{Pa}_i^{\text{add}}$  zu  $\mathbf{Pa}_i^{\text{akt}}$  hinzukommen.

Nachdem wir in Abschnitt 6.8 die gemeinsamen Wahrscheinlichkeitstafeln für jeden Knoten mit seinen Eltern erzeugt haben, benutzen wir hier diese Tafeln und ein ganz ähnliches Verfahren, um die Tafeln an die durch eingefügte Kanten veränderte Netzstruktur anzupassen.

Da die Tafeln nun aber alle (hier auch für  $X_i$ ) in der Form gemeinsamer Wahrscheinlichkeitstafeln vorliegen, müssen aus diesen wieder bedingte Wahrscheinlichkeitstafeln erzeugt werden, wobei über die Nicht-Eltern von  $X_i$  vorher marginalisiert werden kann.

Zu beachten ist dabei daß die Tafel zu  $X_i$  selbst Merkmale enthalten kann die nicht (mehr) Eltern von  $X_i$  sind, weil sie zuvor gelöscht wurden (siehe Abschnitt 7.3.1). Dennoch werden immer wieder die Tafeln zu  $S_{\text{orig}}^h$  herangezogen, da evtl. auch aus  $S_{\text{orig}}^h$  entfernte Kanten im Laufe der Berechnungen wieder eingefügt werden können, in diesem Fall aber das ursprünglich vorhandene Vorwissen zu dieser Kante verloren ginge, wenn an  $S_{\text{akt}}^h$  angepaßte (entsprechend marginalisierte) Tafeln verwendet würden.

Die Berechnungsformel könnte im Prinzip genauso für die Berechnung der neuen Tafel eines Knotens bei aus der Elternmenge entfernten Knoten genutzt werden, da dieses Verfahren aber offensichtlich wesentlich aufwendiger ist als das unter Abschnitt 7.3.1 beschriebene Verfahren bleiben wir beim Entfernen von Kanten bei der zuerst beschriebenen Methode.

Zusätzlich ist bei dem Hinzufügen von Kanten in einen DAG zu beachten, daß die neue Struktur weiterhin der Definition eines DAG's genügt, also insbesondere keine Zyklen enthält, die die Möglichkeit eröffnen würden sich unter Beachtung der Kantenrichtungen im Kreis zu bewegen.

Das Vorwissen zum Knoten  $X_i$  mit den neuen Elternknoten  $\mathbf{Pa}_i^{\text{add}}$  ist dann gegeben durch:

$$p_{\text{neu}}(X_i, \mathbf{Pa}_i^{\text{neu}} | S_{\text{neu}}^h)$$

$$= \sum_{\mathbf{X} \setminus \{X_i, \mathbf{Pa}_i^{\text{neu}}\}} p_{\text{neu}}(X_i, \mathbf{Pa}_i^{\text{add}} \cup \mathbf{Pa}_i^{\text{orig}} | S_{\text{neu}}^h)$$

$$= p_{\text{orig}}(X_i | \mathbf{Pa}_i^{\text{neu}} \cap \mathbf{Pa}_i^{\text{orig}}, S_{\text{neu}}^h) \prod_{pa \in \mathbf{Pa}_i^{\text{neu}}} p_{\text{orig}}(pa | \mathbf{Pa}_{\text{pa}}^{\text{neu}} \cap \mathbf{Pa}_{\text{pa}}^{\text{orig}} \cap \mathbf{Pa}_i^{\text{neu}}, S_{\text{neu}}^h) \quad (36)$$

$$= \left( \frac{\sum_{\mathbf{X} \setminus \{X_i, \mathbf{Pa}_i^{\text{neu}}\}} p_{\text{orig}}(X_i, \mathbf{Pa}_i^{\text{orig}} | S_{\text{neu}}^h)}{\sum_{\mathbf{X} \setminus \{\mathbf{Pa}_i^{\text{neu}}\}} p_{\text{orig}}(X_i, \mathbf{Pa}_i^{\text{orig}} | S_{\text{neu}}^h)} \right) \prod_{pa \in \mathbf{Pa}_i^{\text{neu}}} \left( \frac{\sum_{\mathbf{X} \setminus \{X_i, \mathbf{Pa}_i^{\text{neu}}\}} p_{\text{orig}}(pa, \mathbf{Pa}_{\text{pa}}^{\text{orig}} | S_{\text{neu}}^h)}{\sum_{\mathbf{X} \setminus \{X_i, \mathbf{Pa}_i^{\text{akt}}\}} p_{\text{orig}}(pa, \mathbf{Pa}_{\text{pa}}^{\text{orig}} | S_{\text{neu}}^h)} \right)$$



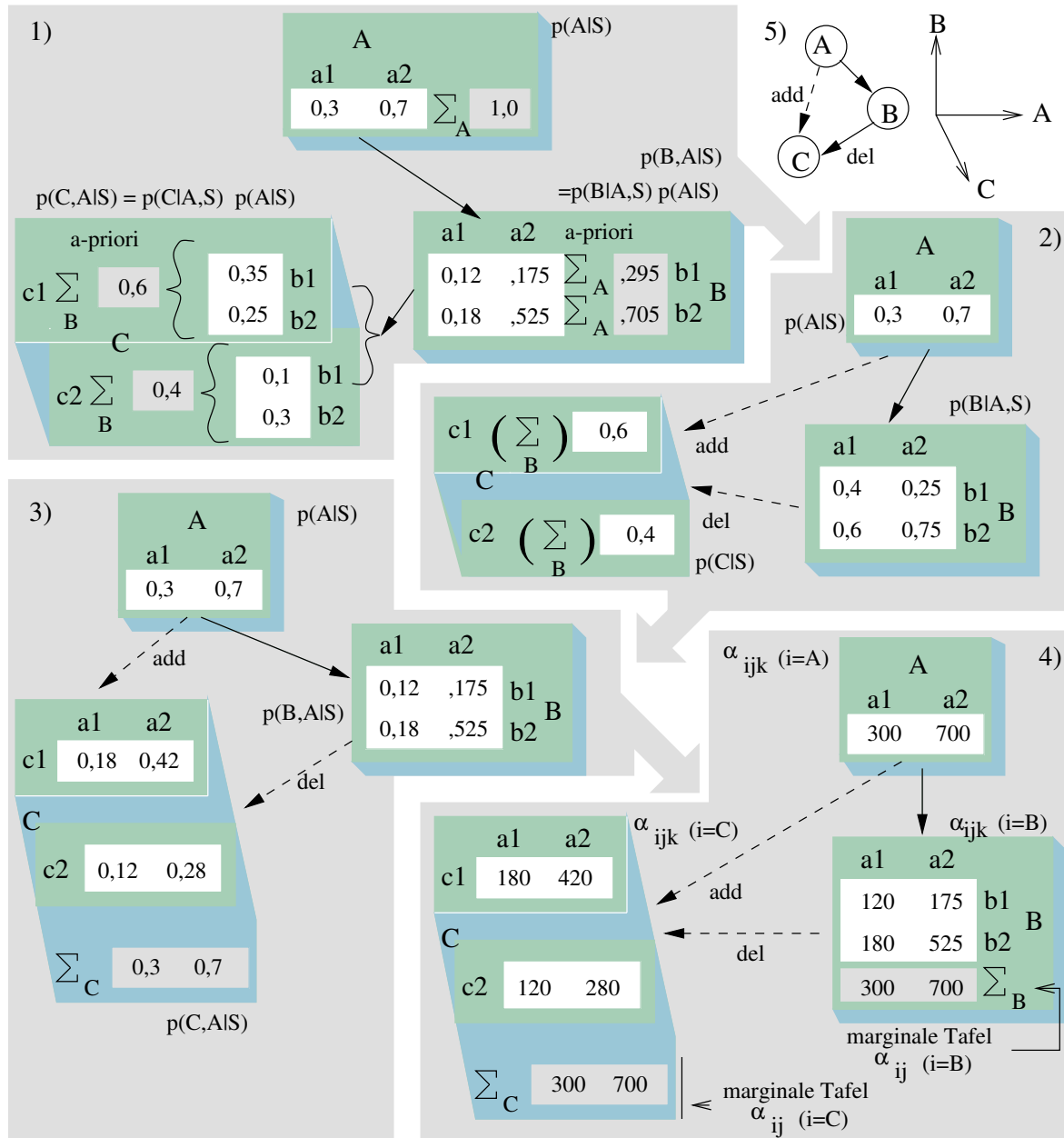


Abbildung 14: Rechenbeispiel für das Hinzufügen einer Kante  $A \rightarrow C$  (Die Kante  $B \rightarrow C$  war in  $S_{orig}^h$  gegeben wurde aber schon vorher entfernt)

- 1) Gemeinsame Wahrscheinlichkeitstafeln von  $X_i$  und  $\mathbf{Pa}_i$  zu  $S_{akt}^h$  (Vorwissen)
- 2) Wsk.-Tafeln über nicht  $\mathbf{Pa}_C$  marginalisiert und in bed. Wsk.-Tafeln überführt ( $p(X_i, \mathbf{Pa}_i | S^h)$ )
- 3) Ausmultiplizieren der bedingten Wsk.-Tafeln zu  $C$  und  $\mathbf{Pa}_C = \{A\}$
- 4)  $\alpha_{ijk}$  ("Vorwissen-Kontingenztafeln") entsprechend  $UserSampleSize = 1000$
- 5) Netzstruktur und Dimensionsaufteilung von  $A$ ,  $B$  und  $C$

Durch Ausmultiplizieren der bedingten Wahrscheinlichkeitstafeln zum Sub-Graphen des ursprünglich vorgegebenen Netzes  $S_{orig}^h$  mit den Knoten  $X_i$  und seinen Eltern im aktuell zu testenden Netz  $S_{neu}^h$  entsteht - wie beim Gesamt-Netz - die gemeinsame (Vorwissen-) Wahrscheinlichkeitstafel der enthaltenen Knoten ( $p_{neu}(X_i, \mathbf{Pa}_i^{neu} | S_{neu}^h)$ ).

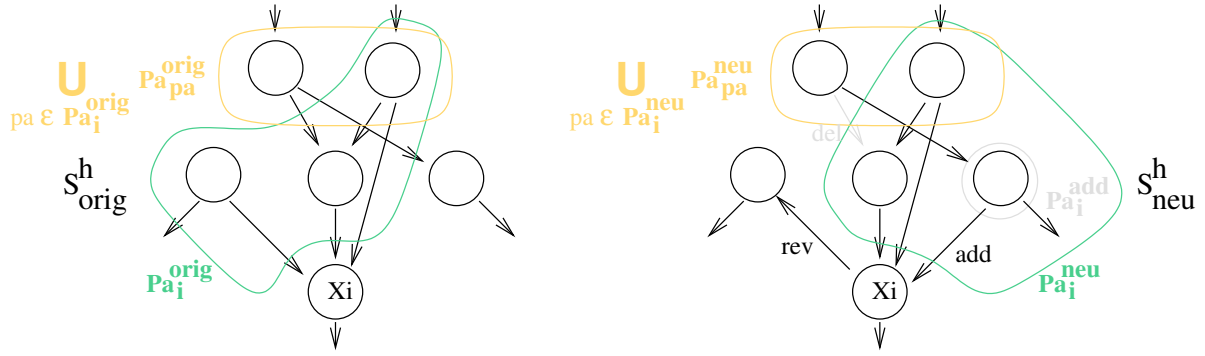


Abbildung 15:

$$= \left( \frac{p_{orig}(X_i, \mathbf{Pa}_i^{\text{neu}} \cap \mathbf{Pa}_i^{\text{orig}}, S_{neu}^h)}{p_{orig}(\mathbf{Pa}_i^{\text{neu}} \cap \mathbf{Pa}_i^{\text{orig}}, S_{neu}^h)} \right) \prod_{pa \in \mathbf{Pa}_i^{\text{neu}}} \left( \frac{p_{orig}(pa, \mathbf{Pa}_{pa}^{\text{neu}} \cap \mathbf{Pa}_{pa}^{\text{orig}} \cap \mathbf{Pa}_i^{\text{neu}}, S_{neu}^h)}{p_{orig}(\mathbf{Pa}_{pa}^{\text{neu}} \cap \mathbf{Pa}_{pa}^{\text{orig}} \cap \mathbf{Pa}_i^{\text{neu}}, S_{neu}^h)} \right)$$

$$= \left[ \sum_{\mathbf{X} \setminus \{X_i, \mathbf{Pa}_i^{\text{neu}}\}} p_{V_{orig}}(\mathbf{X} | S^h) \right]$$

$\mathbf{Pa}_i^{\text{orig}}$	ist die Menge der ursprünglichen Eltern von $X_i$ in $S_{orig}^h$
$\mathbf{Pa}_i^{\text{neu}}$	ist die Menge der Eltern von $X_i$ im zu bewertenden Netz $S_{neu}^h$
$\mathbf{Pa}_i^{\text{akt}}$	ist die Menge der Eltern von $X_i$ im vorherigen Netz $S_{akt}^h$
$\mathbf{Pa}_i^{\text{add}}$	ist die Menge der zu $\mathbf{Pa}_i^{\text{akt}}$ hinzugekommenen Knoten
$\mathbf{Pa}_i^{\text{add}} = \mathbf{Pa}_i^{\text{neu}} \setminus \mathbf{Pa}_i^{\text{akt}}$	$\mathbf{Pa}_i^{\text{neu}} = \mathbf{Pa}_i^{\text{akt}} \cup \mathbf{Pa}_i^{\text{add}}$
$\mathbf{Pa}_{pa}^{\text{orig}}$	sind die Elternknoten eines Elternknotens $pa$ von $X_i$ aus $\mathbf{Pa}_i$ in $S_{orig}^h$ ; $\bigcup_{pa \in \mathbf{Pa}_i^{\text{orig}}} =$ Menge der Großeltern von $X_i$
$\mathbf{Pa}_i^{\text{add}} \cup \mathbf{Pa}_i^{\text{orig}}$	sind die in $S_{neu}^h$ gegenüber $S_{orig}^h$ hinzugekommenen Kanten (diese Menge kann nach mehreren Veränderungsschritten u.U. gleich $\mathbf{Pa}_i^{\text{orig}}$ sein)
$\mathbf{Pa}_i^{\text{neu}} \cap \mathbf{Pa}_i^{\text{orig}}$	sind nur die gegenüber $S_{orig}^h$ hinzugekommenen Kanten (diese Menge kann u.U. leer sein z.B. wenn $\mathbf{Pa}_i^{\text{orig}} = \emptyset$ oder wenn alle Eltern von $X_i$ in $S_{orig}^h$ aus $S_{neu}^h$ entfernt wurden)
$\mathbf{Pa}_{pa}^{\text{neu}} \cap \mathbf{Pa}_{pa}^{\text{orig}} \cap \mathbf{Pa}_i^{\text{neu}}$	sind die Eltern von $pa \in \mathbf{Pa}_i$ in $S_{neu}^h$ , die auch in $S_{orig}^h$ schon Eltern von $pa \in \mathbf{Pa}_i$ waren und gleichzeitig Eltern von $X_i$ in $S_{neu}^h$ sind (auch diese Menge kann leer sein)
del / rev (in Abbildung 15)	zuvor gelöschte / umgedrehte Kante (siehe auch 7.3.1 / 7.4.1)

### 7.3.3 Vereinfachtes Hinzufügen einer Kante

Das zuvor beschriebene Verfahren ist offensichtlich recht aufwendig.

Wenn eine Kante  $Pa_i^{\text{add}} \rightarrow X_i$  eingefügt wird, die nicht bereits in  $S_{orig}^h$  enthalten war, über die also kein Vorwissen vorliegt und wenn diese Kante in  $S_{orig}^h$  keine Verbindung zu anderen aktuellen Elternknoten ( $\mathbf{Pa}_i^{\text{akt}}$ ) hat, also auch kein Vorwissen über Kanten zu anderen Eltern gibt, kann die neue Tafel  $p_{neu}(X_i, \mathbf{Pa}_i^{\text{neu}} | S_{neu}^h)$  durch Multiplikation der alten Tafel zu  $S_{akt}^h$  mit der a-priori-Wahrscheinlichkeitstafel von  $Pa_i^{\text{add}}$  gewonnen werden.

Damit ist die Berechnung für diesen Fall wie folgt möglich:

$$\begin{aligned}
 p_{neu}(X_i, \mathbf{Pa}_i^{\text{neu}} | S_{neu}^h) &= p_{neu}(X_i, \mathbf{Pa}_i^{\text{akt}} \cup Pa_i^{\text{add}} | S_{neu}^h) \\
 &= p_{akt}(X_i, \mathbf{Pa}_i^{\text{akt}} | S_{akt}^h) \cdot p_{V_{orig}}(Pa_i^{\text{add}} | S^h) \\
 &= \left[ \sum_{\mathbf{X} \setminus \{X_i, \mathbf{Pa}_i^{\text{neu}}\}} p_{V_{orig}}(\mathbf{X} | S^h) \right]
 \end{aligned} \tag{37}$$

$$\left| \begin{array}{l} Pa_i^{add} \text{ ist der neue Elternknoten von } X_i \text{ in } S_{neu}^h \\ p_{V_{orw}}(Pa_i^{add}|S^h) \text{ ist die a-priori-Wahrscheinlichkeitstafel zu } Pa_i^{add} \\ \text{sie kann vorab zu jedem Knoten } X_i \text{ erhalten werden durch:} \\ p_{V_{orw}}(X_i|S^h) = \sum_{\mathbf{X} \setminus X_i} p_{orig}(X_i, \mathbf{Pa}_i^{orig}|S_{orig}^h) \end{array} \right|$$

Dieser Fall ist z.B. auch auf das in Abbildung 14 dargestellte Beispiel anwendbar.

## 7.4 Fortführung des Verfahrens über mehrere Veränderungsschritte

In der Beschriebenen Art und Weise lassen sich alle Kantenänderungen, die im Löschen bzw. Hinzufügen einzelner Kanten mit demselben Zielknoten  $X_i$  bestehen, einzeln berechnen. Für unser angestrebtes Verfahren wird es sich in einem Berechnungsschritt immer nur um das Löschen bzw. Hinzufügen einer einzelnen Kante drehen, so daß  $\mathbf{Pa}_i^{add}$  bzw.  $\mathbf{Pa}_i^{del}$  immer nur einen Knoten umfassen.

Ausgehend von der Grundberechnung von  $p(D|S_{orig}^h)$  können wir somit schrittweise einzelne Kantenänderungen vornehmen, bewerten und das Netz entsprechend der besten gefundenen Kantenänderung verbessern, bis das Netz durch keine weitere Veränderung einer einzelnen Kante mehr zu verbessern ist. Dieses Verfahren birgt noch ein paar Probleme, denen wir teilweise begegnen können.

**Def.:** Haben wir uns entsprechend der Bewertungen aller durch eine Kantenänderung aus  $S_{akt}^h$  (im ersten Schritt aus  $S_{orig}^h$ ) erzeugbaren Netze für ein (das beste) Netz  $S_{neu}^h$  entschieden, so setzen wir für die weiteren Iterationsschritte:

$$\begin{aligned} S_{akt}^h &:= S_{neu}^h \\ N_{ijk} &:= N'_{ijk} \\ \alpha_{ijk} &:= \alpha'_{ijk} \end{aligned}$$

### 7.4.1 Umdrehen von Kanten

Ein Problem ist die **Änderung der Richtung** einer Kante. Zwar gilt:

**Def.:** Die Änderung der Richtung einer Kante ist als zusammengesetzte Operation aus Löschen der Kante und Einfügen in der umgekehrten Richtung erklärbar.

Es ist aber nicht nur möglich, sondern auch notwendig beide Teil-Operationen gemeinsam, in einem Schritt vorzunehmen und zu bewerten. Diese Möglichkeit ist insofern von Bedeutung, da Greedy durch seine in Einzelschritten erfolgende Verbesserung der Netzstruktur kaum die "Hürde" nehmen würde eine an sich (ungeachtet der Richtung) vernünftige Kante zu löschen, um sie im darauffolgenden Schritt in der umgekehrten Richtung wieder einzufügen. Die gleichzeitige Bewertung der beiden Operationen Löschen und Einfügen, die einem Umkehren einer Kante gleichkommen, ist aber problemlos zu bewerkstelligen, da sich jede der beiden Teiländerungen auf die Elternmenge eines anderen Knotens auswirkt. Beide Operationen werden also, wenn von  $S_{akt}^h$  ausgehend alle möglichen Änderungsschritte bewertet werden, ohnehin schon berechnet und müssen neben der getrennten Betrachtung nur auch zusammen gesehen werden.

Die Berechnungsvorschrift für eine effiziente Berechnung unter Ausnutzung der separablen Bewertungsformel aus Gleichung (32) muß in dem Fall, daß die Kante zwischen  $X_d$  und  $X_p$  umgedreht wurde wie folgt erweitert werden:

$$p(D|S_{neu}^h) = \frac{p(D|S_{akt}^h) \cdot \prod_j \left( \frac{\Gamma(\alpha'_{dj})}{\Gamma(\alpha'_{dj} + N'_{dj})} \cdot \prod_k \frac{\Gamma(\alpha'_{djk} + N'_{djk})}{\Gamma(\alpha'_{djk})} \right) \cdot \prod_j \left( \frac{\Gamma(\alpha'_{pj})}{\Gamma(\alpha'_{pj} + N'_{pj})} \cdot \prod_k \frac{\Gamma(\alpha'_{pjk} + N'_{pjk})}{\Gamma(\alpha'_{pjk})} \right)}{\prod_j \left( \frac{\Gamma(\alpha_{dj})}{\Gamma(\alpha_{dj} + N_{dj})} \cdot \prod_k \frac{\Gamma(\alpha_{djk} + N_{djk})}{\Gamma(\alpha_{djk})} \right) \cdot \prod_j \left( \frac{\Gamma(\alpha_{pj})}{\Gamma(\alpha_{pj} + N_{pj})} \cdot \prod_k \frac{\Gamma(\alpha_{pjk} + N_{pjk})}{\Gamma(\alpha_{pjk})} \right)} \quad (38)$$

$$\left| \begin{array}{l} p(D|S_{akt}^h) \text{ ist wieder das Produkt der Bewertungen der einzelnen Knoten,} \\ \text{das gegeben ist durch: } \prod_i \prod_j \left( \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \prod_k \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})} \right) \\ \prod_j \left( \frac{\Gamma(\alpha_{dj})}{\Gamma(\alpha_{dj} + N_{dj})} \prod_k \frac{\Gamma(\alpha_{djk} + N_{djk})}{\Gamma(\alpha_{djk})} \right) \text{ ist die "alte" Bewertung zum Knoten } X_d \text{ in } S_{akt}^h \\ \prod_j \left( \frac{\Gamma(\alpha_{pj})}{\Gamma(\alpha_{pj} + N_{pj})} \prod_k \frac{\Gamma(\alpha_{pjk} + N_{pjk})}{\Gamma(\alpha_{pjk})} \right) \text{ ist die "alte" Bewertung zum Knoten } X_p \text{ in } S_{akt}^h \\ \prod_j \left( \frac{\Gamma(\alpha'_{dj})}{\Gamma(\alpha'_{dj} + N'_{dj})} \prod_k \frac{\Gamma(\alpha'_{djk} + N'_{djk})}{\Gamma(\alpha'_{djk})} \right) \text{ ist die geänderte Bewertung zum Knoten } X_d \text{ in } S_{neu}^h \\ \prod_j \left( \frac{\Gamma(\alpha'_{pj})}{\Gamma(\alpha'_{pj} + N'_{pj})} \prod_k \frac{\Gamma(\alpha'_{pjk} + N'_{pjk})}{\Gamma(\alpha'_{pjk})} \right) \text{ ist die geänderte Bewertung zum Knoten } X_p \text{ in } S_{neu}^h \end{array} \right|$$

### 7.4.2 Lokale Maximierung

Auch wenn - womit wir zum zweiten Problem des Greedy-Search-Algorithmus kommen - bei der Richtungsänderung einzelner Kanten das Problem der lokalen Maximierung bei Greedy noch umgangen werden kann, so ist es insgesamt für die Struktursuche doch eine entscheidende Schwäche. Da nämlich jeweils nur lokal maximiert, also nur einen Schritt weit vorausgeblickt und so mit dem Erreichen lokaler Maxima die Suche abgebrochen wird, kann das globale Maximum (also **das** beste Netz) verfehlt werden. Andererseits wäre es aber unmöglich alle denkbaren Netzstrukturen zu bewerten [Chickering et al. (1995)], so daß Greedy eine sinnvolle Heuristik darstellt. Zudem sind wir bisher von Vorwissen ausgegangen, das sofern sinnvoll gewählt einen guten Ausgangspunkt für die Suche darstellt, was das Problem des Verfehlens des globalen Maximums reduziert.

In einfacher Weise zu berücksichtigen ist zu dem exakten Vorwissen über Struktur und bedingte Wahrscheinlichkeiten, solches Vorwissen das in der Form gegeben wird, daß bestimmte nicht im Vorwissen enthaltene Kanten nicht oder nur in einer bestimmten Richtung gesetzt werden dürfen, oder bestehende Kanten nicht gelöscht oder gelöscht aber nicht umgedreht werden dürfen. Auf diese Weise wird der Suchraum von vornherein eingeschränkt, was sowohl der Suchgeschwindigkeit als auch dem - auf das globale Maximum abzielenden - Suchweg zugute kommt. Desweiteren sollte ein gelerntes Netz noch von einem Experten (Supervisor) begutachtet werden, um in einem weiteren Suchlauf evtl. weitere Einschränkungen vorzugeben, um so zu einem möglichst guten Netz zu gelangen. Ein solches Verfahren heißt **Überwachtes Lernen** bzw. **Supervised Learning**. Es existieren auch Ansätze für Unsupervised Learning; siehe dazu z.B. [Heckerman (March 1995)]. Dabei wird insbesondere versucht durch die automatische Einführung unbeobachteter / unbeobachtbarer Variablen als Eltern von Knoten einer Teilknotenmenge die paarweisen Abhängigkeiten unter allen Merkmalen in einer solchen Teilknotenmenge zu erklären und damit aufzubrechen. Weiterhin sollte dabei größerer Wert darauf gelegt werden kausale Abhängigkeiten zu identifizieren.

## 8 Umgang mit fehlendem Vorwissen

Wir haben nun ein Verfahren entwickelt, um eine Netzstruktur, zu bewerten und mit schrittweise um eine Kante veränderten Netzen zu vergleichen. Wir haben uns dabei neben den Daten auch immer auf Vorwissen gestützt. Das Vorwissen betraf hauptsächlich zwei Ebenen:

Erstens gingen wir von Vorwissen betreffend die Netzstruktur aus. Diese Art des Vorwissens als Ausgangspunkt für den Greedy-Search-Algorithmus soll weiterhin gegeben werden (können) unabhängig von der Zweiten Vorwissenart: den eigentlichen bedingten Wahrscheinlichkeiten zum initialen Netz.

Um die Aquirierung dieses Vorwissens wollen wir jetzt umhinkommen und ausschließlich von den Daten ausgehen.

### 8.1 Fehlendes Vorwissen über die Netzstruktur

Besteht schon weitgehende Unklarheit über die Struktur des Netzes kann ein gänzlich kantenloses Netz vorgegeben werden. Dieser Ansatz erhöht aber die Gefahr, daß der Suchalgorithmus bei lokalen Maxima steckenbleibt, die u.U. relativ "weit" von der bestmöglichen Struktur entfernt sein können.

### 8.2 Vorwissen durch "Bayes'sches Unwissen" ersetzen

Eine Möglichkeit besteht darin eine Annahme über die Struktur vorzugeben und die Wahrscheinlichkeitstabellen zum Vorwissen mit "Unwissen", also im Bayes'schen Sinne gemäß der Annahme der Gleichwahrscheinlichkeit aller Ausprägungen jedes Merkmals und der Annahme der totalen Unabhängigkeit (siehe Abschnitt 1.5) aller Merkmale zu füllen, diesem "Vorwissen" aber nur geringes Gewicht beizumessen, also die *UserSampleSize* klein zu wählen. Dieses Vorgehen ist aber problematisch und wirft einige Fragen auf.

#### 8.2.1 Wahl der *UserSampleSize*

Wie klein beispielsweise muß die *UserSampleSize* gewählt werden? Wählt man im Extremfall einen (zu) großen Wert, so führt dies dazu, daß ein Netz mit zu vielen Kanten erstellt wird. Ein hoher Wert für den Benutzer-Beobachtungsumfang bedeutet ja nicht, daß das vorgegebene (kantenlose) Netz bereits mit hoher Wahrscheinlichkeit wahr ist und sich in Kombination mit den Daten daher wenig verändern dürfte, sondern es ergibt sich ein größerer Gesamtbeobachtungsumfang (Benutzer und Daten), der dazu führt, daß auch die durch die gleichverteilte Vorwissen-Vorgabe relativierten Ungleichgewichte im Datensatz wichtig genug genommen werden, um aus ihnen Abhängigkeiten, also Kanten im Netz abzuleiten.

Wird dagegen ein extrem kleiner Wert gewählt, so zeigt sich, daß nur relativ wenige Kanten in ein leeres Netz eingefügt werden, bzw. insgesamt nur wenige bedingte Abhängigkeiten als für eine gute Struktur hilfreich angesehen werden.

Beide Sachverhalte sind zwei Seiten einer Medaille, wobei sich letzterer (noch) schwieriger intuitiv erfassen oder erklären läßt. Wir beleuchten diesen Punkt daher anhand eines einfachen Beispiels und einer mathematischen Betrachtung des Problems.

### 8.2.2 Beispielrechnung zur Wahl der *UserSampleSize*

### 8.2.3 Mathematische Erklärung des beobachteten Effekts

Im in Abbildung 16 dargestellten Beispiel ist zu sehen, daß bei relativ hohen *UserSampleSize*-Werten das Netz  $S^2$  mit der Kante  $A \rightarrow B$  favorisiert wird, während bei kleineren Werten das Netz  $S^1$  ohne diese Kante bevorzugt wird, obwohl aus den Daten eindeutig eine bedingte Abhängigkeit von  $A$  und  $B$  ableitbar wäre.

Während in jedem in Anhang 12.2 aufgeführten Bruch gemäß Gleichung (26) die Differenz zwischen Zähler und Nenner bei veränderten *UserSampleSize*-Werten gleich bleibt, sinken aber die absoluten Werte von Zähler und Nenner bei kleiner werdenden *UserSampleSize*-Werten ab. Es ändert sich zwischen  $S^1$  und  $S^2$  jeweils nur die Bewertung für Knoten  $B$ , während die Bewertung des Knotens  $A$  (immer die ersten drei Brüche) sich zwischen  $S^1$  und  $S^2$  bei gleicher *UserSampleSize* nicht unterscheidet. Jeder Bruch (Zähler und Nenner) zu  $B$  spaltet sich gegenüber  $S^1$  in  $S^2$  in zwei Brüche (asymmetrisch) auf. In den beiden Beispielnetzen haben die Berechnungen für Knoten  $B$  die Struktur:

$$f_1(x) = \left( \frac{\Gamma(4x)}{\Gamma(a+b+c+d+4x)} \cdot \frac{\Gamma(a+b+2x)}{\Gamma(2x)} \cdot \frac{\Gamma(c+d+2x)}{\Gamma(2x)} \right) \quad \text{für } S^1$$

$$f_2(x) = \left( \frac{\Gamma(2x)}{\Gamma(a+c+2x)} \cdot \frac{\Gamma(2x)}{\Gamma(b+d+2x)} \cdot \frac{\Gamma(x+a)}{\Gamma(x)} \cdot \frac{\Gamma(x+b)}{\Gamma(x)} \cdot \frac{\Gamma(x+c)}{\Gamma(x)} \cdot \frac{\Gamma(x+d)}{\Gamma(x)} \right) \quad \text{für } S^2$$

Dabei war im Beispiel konkret:  $a = 1, b = 4, c = 3, d = 2$  und  $4x = \text{UserSampleSize}$ . Mit  $x \in \{1, 2, 3, 4, \dots\}$  und Ersetzung von  $\Gamma(x)$  durch  $(x-1)!$  erhalten wir nach Kürzen:

$$f_1(x) = \left( \frac{2x \cdot (2x+1) \cdot \dots \cdot (a+b+2x-1) \cdot 2x \cdot (2x+1) \cdot \dots \cdot (c+d+2x-1)}{4x \cdot (4x+1) \cdot \dots \cdot (a+b+c+d+4x-1)} \right) \quad \text{für } S^1$$

$$f_2(x) = \left( \frac{x \cdot \dots \cdot (x+a-1) \cdot x \cdot \dots \cdot (x+b-1) \cdot x \cdot \dots \cdot (x+c-1) \cdot x \cdot \dots \cdot (x+d-1)}{2x \cdot (2x+1) \cdot \dots \cdot (a+c+2x-1) \cdot 2x \cdot (2x+1) \cdot \dots \cdot (b+d+2x-1)} \right) \quad \text{für } S^2$$

Nach Ausmultiplizieren der Polynome erhalten wir:

$$f_1(x) = \left( \frac{1}{(4x)^{a+b+c+d} + \dots} \cdot \frac{(2x)^{a+b} + \dots}{1} \cdot \frac{(2x)^{c+d} + \dots}{1} \right) \quad \text{für } S^1$$

$$f_2(x) = \left( \frac{1}{(2x)^{a+c} + \dots} \cdot \frac{1}{(2x)^{a+c} + \dots} \cdot \frac{x^a + \dots}{1} \cdot \frac{x^b + \dots}{1} \cdot \frac{x^c + \dots}{1} \cdot \frac{x^d + \dots}{1} \right) \quad \text{für } S^2$$

Von Interesse sind nur die größten Summanden. Durch weiteres Ausmultiplizieren und Kürzen erhalten wir:

$$f_1(x) = \left( \frac{2^{a+b+c+d} \cdot x^{a+b+c+d} + \dots}{4^{a+b+c+d} \cdot x^{a+b+c+d} + \dots} \right) \rightarrow \frac{1}{2^{a+b+c+d}} \quad \text{für } S^1 \text{ und } x \rightarrow \infty$$

$$f_2(x) = \left( \frac{x^a \cdot x^b \cdot x^c \cdot x^d + \dots}{2^{a+c} \cdot 2^{b+d} \cdot x^{a+c} \cdot x^{b+d} + \dots} \right) \rightarrow \frac{1}{2^{a+b+c+d}} \quad \text{für } S^2 \text{ und } x \rightarrow \infty$$

Beide Bewertungsterme von  $B$  in  $S^1$  bzw.  $S^2$  nähern sich also bei steigendem  $x$  (steigender *UserSampleSize*) demselben Wert und damit auch einander an. Im konkreten Beispiel liegt der Grenzwert bei  $1/2^{10} = 1/1024 \approx 0,000976$ . Für  $x \rightarrow 0$  streben beide Funktionen gegen 0, da  $\Gamma(2x)/(\Gamma(x) \cdot \Gamma(x)) \rightarrow 0$  für  $x \rightarrow 0$ . Beide Bewertungsfunktionen von  $B$  in  $S^1$  bzw.  $S^2$ , in Abhängigkeit zu  $x$  (*UserSampleSize*), sind stetige Funktionen, die gemeinsamen Grenzwerten entgegenstreben. Desweiteren haben  $f_1(x)$  und  $f_2(x)$  nur max. einen Schnittpunkt<sup>24</sup>. Damit gilt, daß wenn es ein  $x_1$  gibt für das die Bewertung von  $B$  in  $S^1$  größer ist als die in  $S^2$  und wenn es ein  $x_2$  gibt mit  $x_2 > x_1$  für das die Bewertung zu  $S^2$  besser ist, dann liegt ein Schnittpunkt im Intervall  $]x_1, x_2[$ . Damit kann es keine weiteren Schnittpunkte geben, insbesondere nicht für Werte größer  $x_2$ .

<sup>24</sup>  $f_1(x)$  ist rechtsgekrümmt und nähert sich dem Grenzwert für  $x \rightarrow \infty$  von unten.  $f_2(x)$  ist links- recht- linksgekrümmt und nähert sich dem Grenzwert von oben.

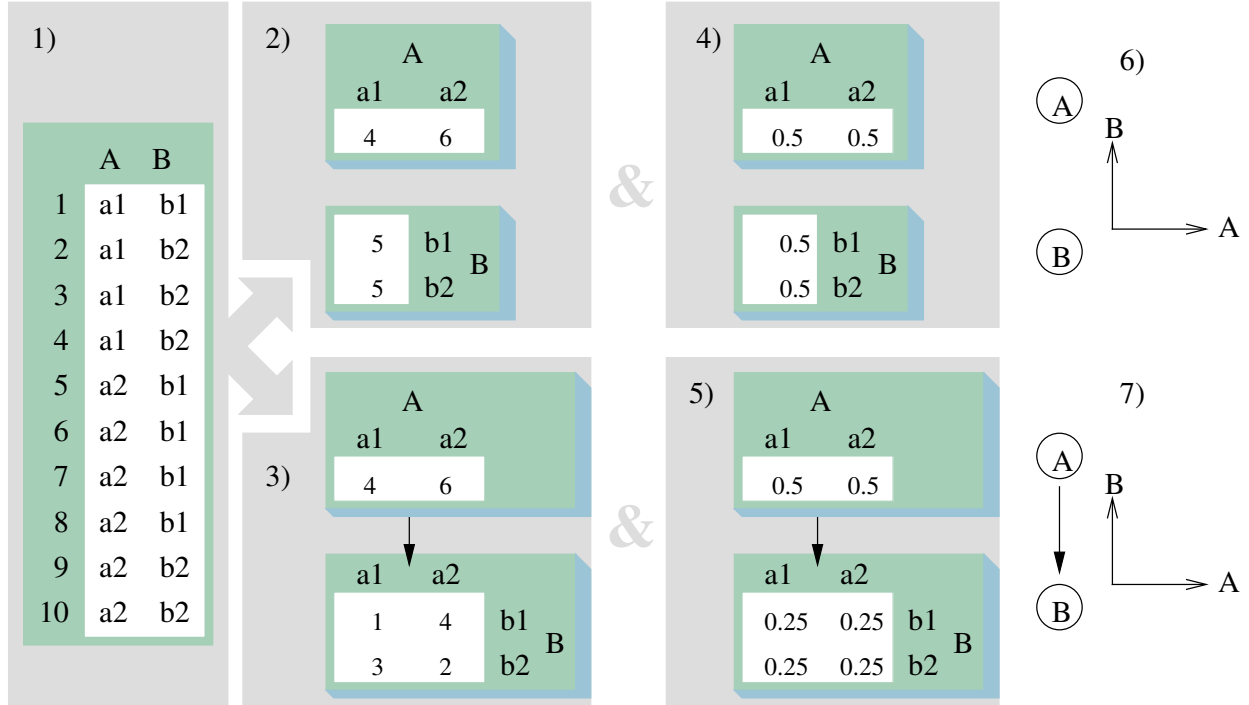


Abbildung 16: Kombination von Wissen aus Daten mit “Bayes’schem Unwissen” **anhand eines Beispiels**

- 1) Datensatz als einzige echte Informationsquelle
- 2)  $N_{ijk}$ -Tabellen zum Netz ohne Kante  $S^1$
- 3)  $N_{ijk}$ -Tabellen zum Netz mit Kante  $S^2$
- 4) nicht informative  $\frac{\alpha_{ijk}}{SampleSize}$ -Tabellen zum Netz ohne Kante  $S^1$   
( $= p_{Vorw}(A|S^1)$  bzw.  $p_{Vorw}(B|S^1)$ )
- 5) nicht informative  $\frac{\alpha_{ijk}}{SampleSize}$ -Tabellen zum Netz mit Kante  $S^2$   
( $= p_{Vorw}(A|S^2)$  bzw.  $p_{Vorw}(A, B|S^2)$ )
- 6)  $S^1$ : Netzstruktur und Dimensionsaufteilung von A und B
- 7)  $S^2$ : Netzstruktur und Dimensionsaufteilung von A und B

**Bsp.:** Bewertungen der beiden Netze  $S^1$  und  $S^2$  mit unterschiedlichen *SampleSize*-Werten:

$N = 10$	$SampleSize = 100$	$\log(p(D S^1)) = -13.9404$	$\log(p(D S^2)) = -\mathbf{13.9144}$
$N = 10$	$SampleSize = 10$	$\log(p(D S^1)) = -14.4757$	$\log(p(D S^2)) = -\mathbf{14.4017}$
$N = 10$	$SampleSize = 1$	$\log(p(D S^1)) = -\mathbf{16.4664}$	$\log(p(D S^2)) = -17.1796$
$N = 10$	$SampleSize = 0.1$	$\log(p(D S^1)) = -\mathbf{20.1899}$	$\log(p(D S^2)) = -22.9379$

(Detaillierte Berechnung: siehe Anhang 12.2)

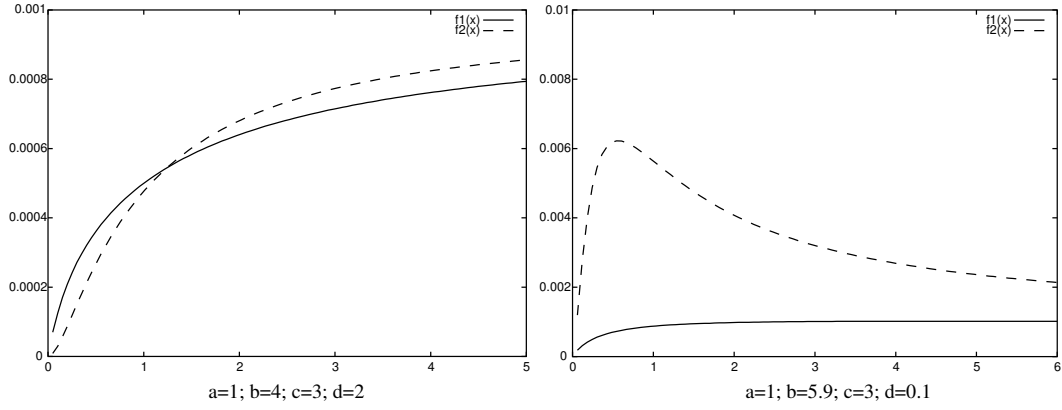


Abbildung 17: plot von  $f_1$ ,  $f_2$  mit gnuplot

Der Schnittpunkt liegt für  $a = 1, b = 4, c = 3, d = 2$  (wie Beispiel) bei  $x \approx 1,24377$ . Wird das Ungleichgewicht der Werte, also die Abhängigkeit verstärkt, z.B. mit  $a = 1, b = 5,9, c = 3, d = 0,1$ , so liegt der Schnittpunkt schon bei  $x \approx 0,00575$  und ließe sich beliebig weiter gegen 0 treiben, indem die Ungleichgewichte der Werte weiter verstärkt würden. Umgekehrt liegt der Schnittpunkt für reduzierte Ungleichgewichte höher; z.B. für  $a = 1, b = 3,5, c = 3, d = 2,5$  erst bei  $x \approx 10004100,00316$ . Mit einer weiteren Reduzierung der Ungleichgewichte kann der Wert gegen  $\infty$  getrieben werden. Damit kann außer im erreichbaren Grenzfalle der Unabhängigkeit für  $S^2$  immer eine bessere Bewertung erreicht werden, indem die *UserSampleSize* hinreichend hoch gewählt wird. Umgekehrt aber kann immer (da hier der Grenzfalle nicht erreichbar ist) für  $S^1$  eine bessere Bewertung erzielt werden, indem die *UserSampleSize* hinreichend klein gewählt wird. Dieses Ergebnis gilt auch für kompliziertere Netzstrukturen.

#### 8.2.4 Die “ $\alpha + 1$ ”-Methode

Die Wahl der *UserSampleSize* erwies sich als unerwartet schwierig. Eigentlich soll hier ja das Fehlen von Vorwissen modelliert werden. Intuitiv müßte das einer Wahl von *UserSampleSize* := 0 entsprechen. Dieses ist jedoch nicht möglich, da damit alle  $\alpha$ -Tafel-Zelle gleich 0 wären und somit diese Wahl an der daraus resultierenden Division durch 0 in der Bewertungsformel (Gleichung (26)) scheitert.

Wie man aber an der *Beta*-Verteilung (siehe Abbildung 5, wie auch der *Dirichlet*-Verteilung erkennt und wie auch in Kapitel 5 erläutert wurde, ist das Minimum für eine Schätzung mindestens eine Beobachtung für jeden Fall ( $\alpha$ -Tafel-Zelle). Es bietet sich also zur Modellierung von Unwissen an, jede  $\alpha$ -Tafel-Zelle mit 1 zu initialisieren. Mit der Wahl von *UserSampleSize* := 0 bleibt es bei  $\alpha_{ijk} = 1 \forall i, j, k$ . Aber auch bei vorhandenem Vorwissen kann diese Methode verwendet werden indem statt Gleichung (27) nun gilt:

$$\alpha_{ijk} = (p(X_i = k, P_{a_i} = j | S^h) \cdot UserSampleSize) + 1 \quad (39)$$

Damit erhöht sich die Gesamtzellensumme jeder  $\alpha$ -Tafel von  $\sum \alpha_i = UserSampleSize$  um die Anzahl der Zellen in  $\alpha_i$ . Diese von der Größe der Tafel abhängige Erhöhung des enthaltenen Beobachtungsumfangs scheint zunächst etwas problematisch, ist aber durch die Annahme von  $\alpha_{ijk} = 1 \forall i, j, k$  als Null-Vorwissen-Level gedeckt. Leider ist aber auch damit das eigentliche Problem nicht ausreichend gelöst.

##### Ein Beispiel:

Gegeben sei ein Datensatz mit 40 Beobachtungen für die binomial verteilten Merkmale  $A$  und  $B$ . Die Kontingenztafel zu diesem Datensatz sei gegeben durch:

		$A$	
		$a_1$	$a_2$
$B$	$b_1$	12	8
	$b_2$	8	12

Weiterhin seien “Unwissen”-Wahrscheinlichkeitstafeln zum kantenlosen DAG bzw. zum DAG  $A \rightarrow B$  gegeben durch:

durch:

$A$		$B$	$b_1$	0.5
$a_1$	$a_2$		$b_2$	0.5
0.5	0.5			

bzw.

$A$	
$a_1$	$a_2$
0.5	0.5

47

		$A$	
		$a_1$	$a_2$
$B$	$b_1$	0.25	0.25
	$b_2$	0.25	0.25

Mit  $UserSampleSize := 0$  (dabei sind die Werte in den Wahrscheinlichkeitstafeln irrelevant) wird das Netz ohne Kante favorisiert, ebenso bei  $UserSampleSize := 1$  und  $UserSampleSize := 10$ . Bei  $UserSampleSize := 100$  (und grösser) wird aber das Netz  $A \rightarrow B$  favorisiert (siehe Anhang 12.3). Nehmen wir dagegen an, wir hätten vorab schon die Hälfte der Daten erhalten und diese unserem Vorwissen zugeordnet und nur die andere Hälfte als Daten verwendet, so daß nun gilt:



$\alpha$ -Tafel zu  $A$  (beide Netze):

		$A$	
		$a_1$	$a_2$
		10+1	10+1

$\alpha$ -Tafel zu  $B$   
im kantenlosen DAG:

$B$	$b_1$	10 + 1
	$b_2$	10 + 1

$\alpha$ -Tafel zu  $B$  in  $A \rightarrow B$ :

		$A$	
		$a_1$	$a_2$
$B$	$b_1$	6 + 1	4 + 1
	$b_2$	4 + 1	6 + 1

Kontingenztafel:

		$A$	
		$a_1$	$a_2$
$B$	$b_1$	6	4
	$b_2$	4	6

In diesem Fall bestätigen sich Vorwissen und Daten gegenseitig zu 100%. Nun wird aber das Netz  $A \rightarrow B$  klar favorisiert, auch unabhängig davon, ob nun “+1” oder nicht (siehe Anhang 12.3). Der Gesamtbeobachtungsumfang liegt jedoch weiterhin bei 40 Beobachtungen.

Das eigentliche Problem ist also nicht gelöst. Die Initialisierung aller  $\alpha$ -Tafel-Zellen mit dem Wert 1 ist für die Modellierung eines Null-Vorwissen-Levels problematisch. Zwar ist es nicht mehr möglich durch Verkleinerung der *UserSampleSize* im Extremfall immer ein kantenloses Netz zu erhalten, aber es ist stattdessen möglich *UserSampleSize* := 0 zu wählen, jedoch wird diese Wahlmöglichkeit den damit geweckten Erwartungen, daß es beispielsweise egal sein müßte wann Wissen einfließt (also ob es z.T. schon zum Vorwissen gehört) nicht gerecht. Auch liegt die mit diesem Verfahren erhaltene Kantenanzahl, wie Versuche bestätigen, eher zu niedrig und nicht, wie durch die ausschließliche Verwendung vom Zufall beeinflusster Daten zu erwarten wäre, zu hoch (*overfitting*). Die bisher gewählte Herangehensweise, daß in jedem Fall Vorwissen anzugeben ist scheint vor diesem Hintergrund die intuitivere Variante darzustellen.

### 8.3 Gewinnung des Vorwissens aus den Daten

Aus den genannten Problemen des ersten Versuchs ohne Vorwissen auszukommen, wollen wir noch eine andere Variante entwickeln. Wir wollen weiterhin unverändert die Gleichung (26) und die sich daraus ableitenden verschiedenen Formeln zur Bewertung von Netzstrukturen verwenden. Wir benötigen also wenigstens formal Vorwissen.

#### 8.3.1 Vorwissen als Korrektiv zu den Daten

Wenn wir uns nocheinmal überlegen, was durch das Vorwissen, so wir welches hatten, bisher geleistet wurde, so erkennen wir, daß es neben einer weiteren Wissensbasis zusätzlich zum Datensatz auch ein Korrektiv gegenüber zufälligen Schwankungen in diesem Datensatz darstellte. Wurde ein Netz  $S_{neu}^h$  mit einer zusätzlichen Kante gegenüber  $S_{orig}^h$  getestet, so mußte sich die aus dem Datensatz wahrscheinlich ableitbare, zumindest “leichte” bedingte Abhängigkeit, gegen das Vorwissen “durchsetzen”, daß diese direkte, bedingte Abhängigkeit, wie sie eine Kante impliziert tatsächlich nicht besteht. Dabei gehen wir davon aus, daß sich Vorwissen und Daten insgesamt nicht stark widersprechen, so daß i.d.R. keine Abhängigkeiten aus verschiedenen Unabhängigkeitsannahmen abgeleitet werden, wie dies bei der zuletzt in Abschnitt 8.2. dargestellten Variante mit künstlich erzeugtem Pseudo-Vorwissen aber der Fall sein könnte (vgl. Abschnitt 6.7). Etwas ähnliches wie es echtes Vorwissen leistet wollen wir nun auch erreichen ohne solches Vorwissen zu haben.

#### 8.3.2 “Doppelte” Verwendung der Daten

Da uns als einzige Informationsquelle der Datensatz  $D$  zur Verfügung steht und die eher willkürliche Wahl von Pseudo-Vorwissen Probleme barg, werden wir nun das Vorwissen ebenfalls aus den Daten gewinnen. Um dabei dieselben Daten nicht mehrfach zu verwenden, muß der Datensatz geteilt werden. Es soll hier jedoch nicht das Ziel verfolgt werden einen Teil des Datensatzes zum Aufstellen und einen anderen zum Testen des Netzes zu verwenden. In diesem Falle könnten durch eine geschickte Teilung der Daten zufällige Schwankungen in den Verteilungen der Merkmale registriert werden. Wir wollen stattdessen “nur” verschiedene Hypothesen gegenüberstellen. In dem Fall, daß eine zusätzliche Kante getestet werden soll, wollen wir eine Art Vorwissen dagegensetzen, daß der Notwendigkeit dieser Kante widerspricht. Daher benutzen wir jeweils für die  $\alpha_{ijk}$ - und die  $N_{ijk}$ -Tabellen alle Beobachtungen im Datensatz  $D$ . Da sich dadurch aber der doppelte Gesamtbeobachtungsumfang ergäbe, der tatsächlich vorhanden ist, lassen wir jede Beobachtung nicht wie

sonst üblich mit “1” in die Kontingenztafeln eingehen, sondern z.B. jeweils mit “0,5”<sup>25</sup>. Dieses Vorgehen ist äquivalent zu einer normalen Erzeugung der Kontingenztafeln für beide Größen,  $\alpha$ ’s und  $N$ ’s und einer nachträglichen Multiplikation der Tafeln mit dem jeweiligen Gewichtungsfaktor (siehe dazu auch Abschnitt 11.3.3 Gleichungen (43)-(46)). Mit diesem Verfahren ist zumindest gewährleistet, daß sich “Vorwissen” und (Rest-)Daten nur bezüglich einer evtl. dem Netz hinzuzufügenden Kante widersprechen, ansonsten aber identisch sind, so daß wie bei echtem Vorwissen keine zusätzlichen Scheinabhängigkeiten z.B. durch unterschiedliche aus Daten bzw. Vorwissen ableitbaren Unabhängigkeitsannahmen entstehen (vgl. Abschnitte 8.3.1 und 6.7). Wir greifen zur Illustration das Beispiel aus Abschnitt 6.7 noch einmal auf und wandeln es entsprechend der aktuellen Fragestellung ab, so daß nun eine Abhängigkeit zwischen  $A$  und  $B$  aus den Daten erkennbar ist.

**Bsp.:**

Kontingenztafel aus 50%  
der Daten  $D$  zu  $A$  und  $B$

		$A$		
		$a_1$	$a_2$	$\Sigma$
$B$	$b_1$	2	6	8
	$b_2$	6	2	8
$\Sigma$		8	8	

“Vorwissen” aus 50% der  
Daten zu  $A$  in  $B(\rightarrow)A$

		$A$		
		$a_1$	$a_2$	
$B$	$b_1$	4	4	
	$b_2$	4	4	
$\Sigma$		8	8	

“Vorwissen” aus 50% der  
Daten zu  $B$  in  $A(\rightarrow)B$

		$A$		
		$a_1$	$a_2$	$\Sigma$
$B$	$b_1$	4	4	8
	$b_2$	4	4	8

Damit erhalten wir (wiederum nach Ersetzen von  $\Gamma(x)$  durch  $(x-1)!$ ):

$$\left( \frac{15!}{7! \cdot 7!} \cdot \frac{15! \cdot 15!}{31!} \right) \cdot \left( \frac{7!}{3! \cdot 3!} \cdot \frac{5! \cdot 9!}{15!} \cdot \frac{7!}{3! \cdot 3!} \cdot \frac{9! \cdot 5!}{15!} \right) = p(D|S^h) \approx 2.3 \cdot 10^{-10} \quad S^h = B \leftarrow \& \rightarrow A$$

$$\left( \frac{15!}{7! \cdot 7!} \cdot \frac{15! \cdot 15!}{31!} \right) \cdot \left( \frac{15!}{7! \cdot 7!} \cdot \frac{15! \cdot 15!}{31!} \right) = p(D|S^h) \approx 1.1 \cdot 10^{-10} \quad S^h = A \quad B$$

Trotz Unabhängigkeitsannahme im “Vorwissen” wird die Abhängigkeit erkannt und in eine Kante umgesetzt. Bei noch schwächeren Abhängigkeiten kann aber eventuell auch die Unabhängigkeitsannahme präferiert werden, da der Vorteil, des Netzes mit Kante bei einer einfachen Teilung der Daten ( $\alpha_{ijk} := N_{ijk}$ ) noch weitaus stärker ausfällt (Bewertung von  $S^h = A \quad B$  unverändert):

$$\left( \frac{15!}{7! \cdot 7!} \cdot \frac{15! \cdot 15!}{31!} \right) \cdot \left( \frac{7!}{3! \cdot 3!} \cdot \frac{3! \cdot 11!}{15!} \cdot \frac{7!}{3! \cdot 3!} \cdot \frac{11! \cdot 3!}{15!} \right) = p(D|S^h) \approx 7.0 \cdot 10^{-9} \quad S^h = B \leftarrow \& \rightarrow A$$

Damit sollte veranschaulicht werden, daß diese Art der Wahl des “Vorwissens” dem eingangs formulierten Anspruch *overfitting* zu vermeiden genügt.

### 8.3.3 Vermeidung von Nullzellen

Um Nullzellen zu vermeiden wird zusätzlich auf jede  $\alpha$ -Zelle noch ein kleiner Wert aufaddiert<sup>26</sup>.

### 8.3.4 Erstellung der “Vorwissen”-Tafeln bei Netzveränderungen

Es sind die  $N_{ijk}$ ’s wie bisher die Kontingenztafeln zur aktuell zu testenden Netzhypothese  $S^h$ . Um eine Ausgangsbewertung zu  $S^h$  zu erhalten werden zunächst auch die  $\alpha_{ijk}$  zum aktuellen Netz  $S^h$  erzeugt. Um z.B. ein Netz mit einer hinzugefügten Kante zu testen werden die  $\alpha_{ijk}$  zum Netz ohne diese Kante erstellt und dann wie zuvor beschrieben an das aktuelle Netz angepaßt (Abschnitt 7.3.2 und 7.3.3). Auf diese Weise muß sich die neue Kante quasi gegen die Annahme bewähren, daß sie überflüssig ist. Schafft sie das wird wieder die neue Ausgangslage bewertet, indem auch die  $\alpha_{ijk}$ ’s entsprechend dem neuen Netz erzeugt werden. Diese Neuberechnung wird notwendig, weil wir wenn wir uns für ein Netz mit einer neuen Kante entschieden haben, diese neue Kante dann auch dem Vorwissen zugerechnet werden muß. Anders als in der ersten Variante, wo zu jeder veränderten Netzstruktur lediglich in ihrer Größe veränderte, aber immer mit Gleichverteilung gefüllte Vorwissen-Tafeln entstehen, kann (und soll) ja hier das “Vorwissen” durchaus Abhängigkeitsinformationen enthalten, jedoch nur die in  $S^h_{akt}$  und noch nicht die zusätzlich in  $S^h_{neu}$  enthaltenen, wenn eine Kante hinzugefügt wird.

<sup>25</sup> andere Verhältnisse (z.B. 0.2 : 0.8) sind auch denkbar - wir behandeln diesen Punkt noch

<sup>26</sup> z.B. 0.01 oder kleiner

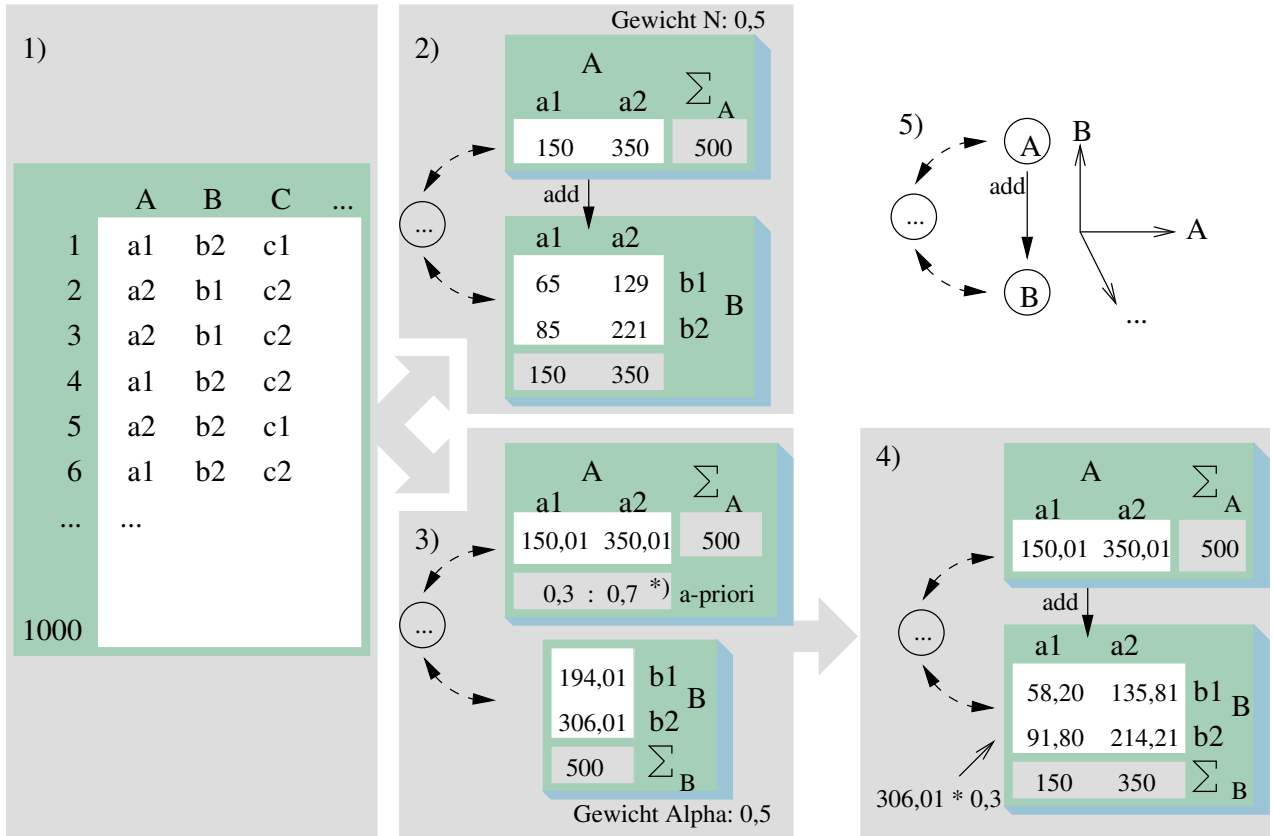


Abbildung 18: “Doppelte” Verwendung der Daten

- 1) Datensatz als einzige Informationsquelle
  - 2)  $N_{ijk}$ -Tabellen
  - 3)  $\alpha_{ijk}$ -Tabellen zum Netz ohne die neue Kante (jede Zelle +0,01)
  - 4)  $\alpha_{ijk}$ -Tabellen zum Netz mit neuer Kante ( $p(B|A, S_{neu}^h) = p(B|S_{akt}^h)p(A|S^h)$ )
  - 5) Netzstruktur und Dimensionsaufteilung von A und B (sonst. Merkmale: “...”)
- \*) Aus dem Größenverhältnis 150,01 : 350,01 zu berechnen

Da es darum geht nicht zuviele Kanten ( $\rightarrow$  overfitting) zu erzeugen, wird beim Löschen einer Kante anders verfahren: Hier werden auch die  $\alpha_{ijk}$ ’s zum aktuell zu testenden Netz, also ohne die möglicherweise zu löschende Kante, erzeugt (Eine erneute Grundbewertung entfällt bzw. ist damit bereits vorgenommen, da sie dasselbe Ergebnis liefern würde).

### 8.3.5 Wahl des Teilungsverhältnisses der Daten

Eine bedeutende Frage ist bei dieser Variante, wie man das “Vorwissen” ( $\alpha$ ) gegen die Daten gewichtet. Mit der oben angedeuteten Variante, beide Wissensformen (Daten und “Vorwissen”) jeweils aus der Hälfte (“0.5 : 0.5”) der Daten zu erzeugen, werden beide gleichgewichtet. Dieses Verhältnis hat einen Einfluß auf die Kantenanzahl und damit auf die Anpassung des Netzes an die Daten. Je größer der Anteil der Daten ist, die für das Vorwissen genutzt werden, desto weniger Kanten werden tendenziell in ein Netz eingefügt, solange dabei sich das Teilungsverhältnis in vernünftigen Grenzen bewegt. Trotzdem ist die Berechnung, wie sich in Simulationen zeigt, gegenüber Schwankungen in dieser Aufteilung relativ robust<sup>27</sup>, im Vergleich zu den eher starken und unkalkulierbaren Schwankungen bei der ersten Berechnungsform, in der die *UserSampleSize*

<sup>27</sup>tatsächlich kommen hier eher durch den Zufall bestimmte Unterschiede dadurch zustande, daß evtl. schon bei kleinen Änderungen des Datenaufteilungsverhältnisses ein anderer Suchpfad (siehe Abbildung 12) eingeschlagen wird und daher im Einzelfall oft nicht abzusehen ist, in welche Richtung sich die Kantenanzahl bei einer bestimmten Aufteilungsänderung entwickeln wird.

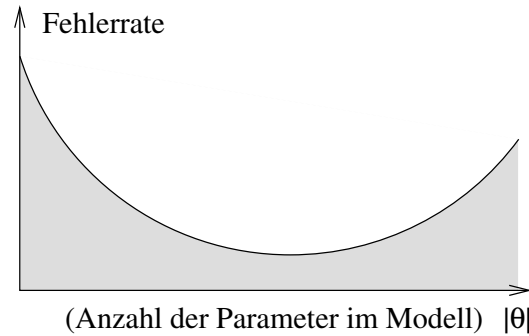


Abbildung 19: angedeutete **Entwicklung der Fehlerrate** in parametrisierten statistischen Modellen allgemein, in Abhängigkeit von der Anzahl der verwendeten Parameter

variiert wurde. Werte die dem Vorwissen ein Gewicht von 0.2 bis 0.5 zuweisen haben sich als vernünftig erwiesen.

## 8.4 (Teil-) Vorwissen über die Netzstruktur

Wie auch immer beim Umgang mit fehlendem Vorwissen über die Wahrscheinlichkeitsverteilung der Merkmale vorgegangen wird, auch hier ist es von Vorteil, wenn der Benutzer wenigstens einige Vorgaben über die Struktur des Netzes machen kann - durch Vorgabe bzw. Ausschluß einiger Kanten(-veränderungen). Wenn das nicht möglich ist, erhöht sich die Gefahr, daß der Suchalgorithmus bei lokalen Minima steckenbleibt. Eine Möglichkeit dieses Problem auf automatisiertem Wege zu reduzieren bestünde darin, zufällige Netze als Startpunkte für den Greedy-Search-Algorithmus vorzugeben. Es zeigt sich bei Tests jedoch, daß unter der hohen Anzahl verschiedener Netze der Beginn mit vollkommen unverbundenen Knoten der "Wahrheit" zumeist am nächsten kommt. Mehr Sinn macht deshalb, eine so gelernte Netzstruktur noch einmal vom Experten prüfen zu lassen. Besonderes Augenmerk sollte der Benutzer dabei auf vollständig verbundene Teilgraphen legen, da "Greedy" eine, in der Netzstruktur, noch nicht ausreichend abgebildete Abhängigkeitsstruktur oft durch zusätzliche Kanten (zumal bei großen Datensätzen) ausgleicht, obwohl die Struktur durch andere Veränderungen in dem bestehenden Netz, wie Richtungsänderungen einzelner Kanten, noch besser würde, solche Lösungen von Greedy aber nicht gefunden werden, weil deren Vorteile erst erkennbar werden, wenn über mehr als eine Kantenänderung hinweg geblickt werden könnte.

## 8.5 Einbeziehung der Netz-a-priori-Wahrscheinlichkeit

Um ein zusätzliches Korrektiv gegen zu viele Kanten (*Overfitting*) einzuführen, kommen wir noch einmal auf die bisher vernachlässigte Netz-a-priori-Wahrscheinlichkeit aus Gleichung (22) zurück. Aus der Beobachtung heraus, daß die Fehlerrate eines Modells mit zunehmender Parameterzahl wieder steigt<sup>28</sup> - eben aus der daraus resultierenden Überschätzung der Daten - könnte die a-priori-Wahrscheinlichkeit eines Netzes über die Anzahl der zu schätzenden Parameter bestimmt werden (siehe auch Abbildung 19). Dazu bietet sich ein Term aus dem *Bayesian Information Criterion* (BIC) [Schwarz (1978)] an:

$$p(S^h) := e^{-\frac{B}{2} \log N} \quad (40)$$

$N$ $B = \dim(S^h)$	ist der Umfang des Datensatzes $D$ ist die Anzahl der zu schätzenden Parameter (siehe Gleichung (1))
------------------------	------------------------------------------------------------------------------------------------------------

# 9 Nachbetrachtung

## 9.1 Uneinheitliche *UserSampleSize*

Wir haben bislang einen einheitlichen Wert *UserSampleSize* angenommen, der gleichsam den Erfahrungsschatz des Benutzers (Experten) widerspiegelt, der das Vorwissen - soweit vorhanden - in das System einbringt.

<sup>28</sup>Ein Problem, daß genauso z.B. für Regressionsmodelle oder Neuronale Netze gilt

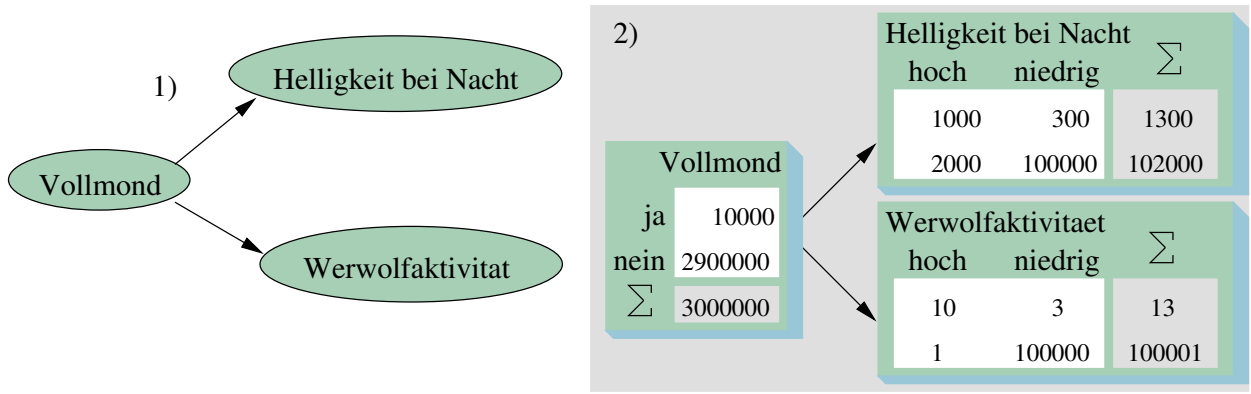


Abbildung 20: Unterschiedliche *UserSampleSize*-Werte für einzelne bedingte Wahrscheinlichkeiten

Möglicherweise wurde das Vorwissen tatsächlich selbst aus älteren Daten gewonnen. Die entsprechenden Kontingenztafeln lassen sich mit dem *UserSampleSize*-Wert aus den Wahrscheinlichkeitstafeln rückrechnen. Basiert das Vorwissen jedoch nicht auf Daten, sondern auf Expertenwissen oder physikalischen Messungen, ist dieses Vorgehen einer einheitlichen *UserSampleSize* nicht ganz unproblematisch, da nicht unbedingt einzusehen ist, daß die Angabe bedingter Wahrscheinlichkeiten umso weniger ernst genommen wird, je mehr Eltern ein Knoten hat, sprich je größer die Wahrscheinlichkeitstafel für den fraglichen Knoten wird. Anders ausgedrückt können Tafeln gleicher Größe durchaus mit unterschiedlicher Gewißheit über das enthaltene Wissen behaftet sein. Ja die für jede Elternzustandskonfiguration gegebenen bedingten Wahrscheinlichkeiten, können verschiedene Sicherheitsgrade haben, die nicht aus den Eltern-Tafeln ableitbar sein müssen.

In dem in Abbildung 20 dargestellten Fall könnte es sicherlich angemessen sein, das Vorwissen über die einzelnen Wahrscheinlichkeiten mit unterschiedlichen Sicherheiten anzunehmen. Da wie an Gleichung (26) zu sehen, jeweils das Vorwissen zu  $X_{ijk}$  (mit/ohne Daten) zum marginalen Wissen nur über die Elternzustände in's Verhältnis gesetzt wird, also Dirichlet-Verteilungen für die bedingten Wahrscheinlichkeiten angenommen werden, ist es möglich die gegebenen bedingten Wahrscheinlichkeiten zu jeder Elternzustandskonfiguration unterschiedlich zu gewichten, wie das in Abbildung 20 angedeutet ist. Entweder müssten dann aber sehr differenzierte *UserSampleSize*-Werte eingefordert werden, oder man verlangt vom Benutzer statt der bedingten Wahrscheinlichkeitstafeln gleich die "bedingten Kontingenztafeln", die der differenzierten Gewichtung des Vorwissens rechnung tragen (Abbildung 20), sonst aber nie gebraucht werden. Mit "bedingten Kontingenztafeln" sind Tafeln gemeint, in denen nur die Werte für je eine Elternzustandskonfiguration in Relation zueinander gesehen werden können, während die marginalen Werte über die Elternzustände die (unterschiedlichen) Sicherheiten widerspiegeln, mit der die bedingten Wahrscheinlichkeiten gegeben sind. Problematisch ist aber die Tatsache, daß die Tafeln nicht mehr konsistent sind, dh. unterschiedliche Tafeln spiegeln unterschiedliches Wissen über dieselben Knoten wieder. Kaum noch zu erklären ist dabei die Ableitung des Vorwissens zu gegenüber  $S_{orig}^h$  veränderten Netzhypothesen. Wir verzichten deshalb auf weitere Überlegungen in dieser Richtung und bleiben bei einem einheitlich gewählten *UserSampleSize*-Wert.

## 10 Simulationen und Testberechnungen

### 10.1 Testverfahren

Um den Lernalgorithmus, wie er nun theoretisch und im Anschluß in Grundzügen auch in der Implementierung vorgestellt ist, zu testen, bieten sich verschiedene Verfahren an. Zum einen können **reale Daten** verwendet werden um daraus die Netzstruktur zu lernen. Zum anderen kann aber auch ein komplettes Bayes'sches Netz, bestehend aus der Struktur und den bedingten Wahrscheinlichkeitstafeln vorgegeben werden, um dazu korrespondierende, möglichst perfekt die vorgegebene Wahrscheinlichkeitsverteilung abbildende Daten (**perfekte Daten**) zu erzeugen, um dann (indem wir Netzstruktur und Wahrscheinlichkeitsvorgabe wieder "vergessen") nur aus den künstlich erzeugten Daten das Netz rückzurechnen versuchen.

Der große Vorteil der letzteren Methode besteht darin, daß man das Ergebnis des Lernalgorithmus mit dem wahren Netz vergleichen kann, um den Grad der Übereinstimmung in Struktur und Bewertung zu ermitteln. Demgegenüber ist die "wahre" Netzstruktur zu echten Daten unbekannt und könnte nur von einem Experten angegeben werden. Dabei wäre jedoch nicht klar, ob Abweichungen der gelernten Netzstruktur zu der Expertengegebenen aus der "Unzulänglichkeit" der Daten, oder auch aus der möglicherweise von vornher-

ein nicht angemessenen Modellannahme eines Bayes'schen Netzes herrühren. Lediglich die Brauchbarkeit der Suchheuristik (Greedy-Search) könnte ermittelt werden, indem die Bewertung entsprechend Gleichung (26) des vorgegebenen Netzes mit der des gelernten Netzes verglichen wird. Tatsächlich birgt aber gerade der Umstand, daß sichergestellt ist, daß "perfekte" Daten weitestmöglich (abhängig vom Beobachtungsumfang des Datensatzes) eine bestimmte Netzstruktur abbilden auch das Problem, daß eine Überschätzung der aus den Daten ableitbaren Abhängigkeiten kaum möglich ist, da nur "wahre" Abhängigkeiten in den Daten enthalten sind.

Aus diesem Problem ergibt sich eine dritte Testmöglichkeit. Wir werden zu einem vorgegebenen Bayes'schen Netz (incl. bedingten Wahrscheinlichkeiten) Daten durch **Sampling** gewinnen, d.h. wir werden nicht den die Wahrscheinlichkeitsvorgabe bestmöglich wiedergebenden Datensatz erzeugen, sondern mithilfe eines Zufallsgenerators Daten simulieren, die wie echte Daten auch im Rahmen von zufälligen Schwankungen vom optimalen Datensatz abweichen können. Im Rahmen zufälliger Schwankungen liegt theoretisch jede Abweichung vom "perfekten" Datensatz, allerdings sind starke Abweichungen weniger wahrscheinlich als kleinere "Fehler". Zudem wird der Datensatz mit steigendem Beobachtungsumfang genauer - weniger von den zufälligen Schwankungen beeinflusst (schwaches Gesetz der Großen Zahlen). Mithilfe des Daten-Samplings werden möglichst viele verschiedene Datensätze erzeugt und aus jedem eine Netzstruktur rückgerechnet, um dabei die "Treffer", also die Netzstrukturen die mit dem Ausgangsnetz übereinstimmen, zu zählen und eine Trefferquote zu gewinnen. Von besonderem Interesse ist dabei die Abhängigkeit der Trefferquote vom Beobachtungsumfang.

Da wir zum Testen in jedem Fall ohne Vorwissen auskommen müssen<sup>29</sup>, werden die in Kapitel 8 dafür vorgestellten Verfahren gegeneinandergestellt. Durch Anwendung aller drei Testmethoden mit künstlich erzeugten und echten Daten können wir die Vorteile beider Ansätze nutzen, um den Lernalgorithmus umfassend zu testen.

Ein viertes Verfahren könnte mit echten oder gesampelten Daten, mit Hinblick auf die eigentliche Nutzung des zu lernenden Bayes'schen Netzes abgeleitet werden, indem mithilfe von Testdaten versucht wird eine **Klassifikation** bestimmter Merkmale in den einzelnen Fällen des Testdatensatzes vorzunehmen und die "Treffer" zu zählen (da die wahre Ausprägung ja bekannt ist). Die Testdaten sollten dabei andere als diejenigen sein, aus denen das Netz gelernt wurde (Teilung der Daten). Für dieses Verfahren müßte man sich aber auf bestimmte "Zielknoten" festlegen, die zu klassifizieren wären. In dieser Arbeit wurde aber lediglich das Ziel verfolgt die wahren Abhängigkeiten durch die Netzstruktur eines DAG's abzubilden, daher soll auch nur dieses Ziel versucht werden durch Tests zu verifizieren, d.h. wir verzichten auf diesen Ansatz zur Bestimmung der Qualität eines Bayes'schen Netzes.

## 10.2 Berechnungen mit "perfekten" Daten

### 10.2.1 Gewinnung "perfekter" Daten

Wir werden zunächst Berechnungen mit möglichst perfekten Daten zu einer vorgegebenen Netzstruktur und zugehörigen bedingten Wahrscheinlichkeiten durchführen.

Das verwendete Verfahren zur Erzeugung solcher Daten zu einem Bayes'schen Netz ist relativ simpel. Da wir zum Testen mit in ihrer Knotenanzahl sehr überschaubaren Netzen arbeiten, leisten wir uns ausnahmsweise die Erzeugung der vollständigen Wahrscheinlichkeitstafel zu allen Knoten (siehe Gleichung (1)). Danach wird jede Zelle dieser Tafel mit dem gewünschten Beobachtungsumfang multipliziert und in eine entsprechende ganzzahlig gerundete Anzahl korrespondierender Beobachtungen im zu erzeugenden Datensatz umgesetzt. Durch die Rundung der Werte wird der vorgegebene Beobachtungsumfang möglicherweise leicht über- bzw. unterschritten. Mit der notwendigen Rundung der Werte (da es nur "ganze" Fälle in einem Datensatz geben kann) wird außerdem die Wahrscheinlichkeitstafel i.d.R. nicht wirklich perfekt durch die Daten repräsentiert. Anders als bei echten Daten, in die **zusätzlich** noch zufällige Schwankungen und nichtmeßbare (nichtgemessene) Störungen hineinspielen, handelt es sich hier jedoch um ein rein numerisches Problem.

Der verwendete Simulations-Algorithmus soll folgendes leisten:

1. Zu einem gegebenen DAG selbstständig, mittels eines Zufallsgenerators bedingte Wahrscheinlichkeitstafeln erzeugen. Mit einem Parameter **rnd** wird der Zufallsgenerator initialisiert, so daß einerseits mit veränderten **rnd**-Werten verschiedene bedingte Wahrscheinlichkeiten zu einunddemselben DAG erzeugt werden können, aber andererseits auch mit gleichen **rnd**-Werten jederzeit identische Wahrscheinlichkeitstafeln reproduzierbar sind.

---

<sup>29</sup>bei echten Daten aus Expertenmangel und bei Daten-Sampling, weil sinnvollerweise nur das als wahr bekannte Netz vorgegeben werden könnte, was aber i.d.R. nicht mehr zu verbessern ist

2. Zu den so erzeugten bedingten Wahrscheinlichkeiten und dem gewünschten Beobachtungsumfang *SampleSize*, entsprechend dem oben vorgestellten Verfahren, einen künstlichen Datensatz erzeugen.
3. Ansetzen des Suchalgorithmus auf das **vorgegebene** Netz um festzustellen, ob die zufällig erzeugten bedingten Wahrscheinlichkeiten oder die nicht perfekte Repräsentation dieser Wahrscheinlichkeitsverteilung durch die Daten (insbesondere bei kleinen *SampleSize*-Werten) zu einer Veränderung ("Verbesserung") selbst des eigentlich wahren DAG's durch den Lernalgorithmus führt.
4. Versuch das vorgegebene Netz, beginnend mit einem kantenlosen DAG, aus den gesampelten Daten rückzurechnen.

Für den Lernalgorithmus selbst sind verschiedene Modi einstellbar. Das zu verwendende Verfahren zum Umgang mit dem fehlenden Vorwissen ist über `mode` einzustellen. Bei der Angabe `mode=1` wird das Vorwissen aus den Daten erzeugt, während bei `mode=2` "Unwissen"-Tafeln zum DAG erzeugt werden. Die Gewichtung des so künstlich erzeugten Vorwissens gegenüber den eigentlichen Daten wird über einen `alpha(weight)`-Wert eingestellt.

Wir führen nun einen ersten Testlauf durch, um das Verfahren anhand der Testausgaben näher zu betrachten. Das Simulations-Tool ist ebenso wie der Lernalgorithmus alleine aus S-Plus aufzurufen, wobei die Implementierung der eigentlichen Funktionalität aus den in Kapitel 11 genannten Gründen weitgehend in C/C++ erfolgte.

### 10.2.2 Versuch 1: "Vorwissen" aus Daten (relativ geringer Anteil)

Im ersten Versuch wollen wir das Vorwissen, wie im Abschnitt 8.3 beschrieben, aus den Daten erzeugen (`mode=1`) und einen relativ großen Datensatz mit 10000 Beobachtungen erzeugen. Aufruf der Funktion:

```
> SIMULATE(dag=testDAG,mode=1,alpha=0.2,sampleSize=10000)
```

Die Funktion gibt nun zunächst den DAG in Matrix-Form sowie die erzeugten bedingten Wahrscheinlichkeitstafeln aus (siehe Anhang 12.4). Intern wird dann der Datensatz mit einem Beobachtungsumfang von ca. 10000 Fällen erzeugt. Die Ausgabe dieser Daten wäre in jedem Fall zu umfangreich.

Im Anschluß wird nun versucht, mittels der Lern-Funktion der gegebene DAG (`testDAG`) unter Verwendung der Daten zu verbessern. Im Normalfall sollte das nicht möglich sein.

```
+-----+
|      FIND / IMPROVE NETSTRUCTUR OF A BAYESIAN (CAUSAL) NETWORK      |
|              BY GIVEN DATA (AND USER-KNOWLEDGE)                    |
+-----+

START
dataset ..... : is sorted (do nothing)
cases in dataset ..... : 9999
different cases in dataset ..... : 32 (-99.7 %)
number of edges in initial DAG .. : 5
possible changes excluded by user : 0
testing DAG only with data ..... : splitting data
splitting data by ..... : 0.800:0.200 (data:pseudo-alpha)
search mode ..... : retractions allowed
possible changes are ..... : add del reverse edges
max. count of changes ..... : 100
term for net-prior ..... : is not used
calculate for D(ata) & S(tructur) : p(D|S)
max. shown improves ..... : show only changes

SEARCH
[ 0] INIT rate for initial DAG          p(D|S) = exp(  -20355.2188)
+-----+
| no changes made ...                    |
+-----+
[ 1] STOP no improvement                 p(D|S) = exp(  -20355.2188)
```

Der Suchalgorithmus gibt vor dem Beginn seiner eigentlichen Berechnungen eine Beschreibung der Eingangsparameter aus: Auffällig ist die enorme Diskrepanz zwischen der **Gesamtzahl** der Fälle im Datensatz, die bei 9999 liegt (die anvisierte Zahl von 10000 wurde damit knapp verfehlt) und der Anzahl der **unterschiedlichen** Fälle im Datensatz von nur 32. Da wir mit fünf binomialen Variablen arbeiten, sind auch nur  $2^5 = 32$  verschiedene Kombinationen möglich - jeder denkbare Fall ist also mindestens einmal im Datensatz vertreten.

Danach beginnt die eigentliche Netzverbesserungsphase, die hier aber erwartungsgemäß sofort abgebrochen wird, da das gegebene Netz nicht zu verbessern ist. Im letzten Schritt wird ausgehend von einem kantenlosen DAG und den Daten versucht durch einen weiteren Aufruf des Lernalgorithmus<sup>9</sup> den `testDAG` rückzurechnen.

```

+-----+
|      FIND / IMPROVE NETSTRUCTUR OF A BAYESIAN (CAUSAL) NETWORK      |
|      BY GIVEN DATA (AND USER-KNOWLEDGE)                            |
+-----+

START
dataset ..... : is sorted (do nothing)
cases in dataset ..... : 9999
different cases in dataset ..... : 32 (-99.7 %)
number of edges in initial DAG .. : 0
possible changes excluded by user : 0
testing DAG only with data ..... : splitting data
splitting data by ..... : 0.800:0.200 (data:pseudo-alpha)
search mode ..... : retractions allowed
possible changes are ..... : add del reverse edges
max. count of changes ..... : 100
term for net-prior ..... : is not used
calculate for D(ata) & S(tructur) : p(D|S)
max. shown improves ..... : show only changes

SEARCH
[ 0] INIT rate for initial DAG                p(D|S) = exp(  -22588.0703)
+-----+
[ 1] | ADD          D ---> E                p(D|S) = exp(  -21456.3633) |
[ 2] | ADD          A ---> D                p(D|S) = exp(  -20734.4336) |
[ 3] | ADD          A ---> B                p(D|S) = exp(  -20579.4805) |
[ 4] | ADD          B ---> D                p(D|S) = exp(  -20431.7852) |
[ 5] | ADD          B ---> E                p(D|S) = exp(  -20363.3438) |
[ 6] | ADD          B ---> C                p(D|S) = exp(  -20356.0234) |
+-----+
[ 7] STOP no improvement                      p(D|S) = exp(  -20356.0234)

```

Der Lernalgorithmus wird mit den selben Parametern aufgerufen bis auf das nun kantenlose Ausgangs-DAG (“number of edges in initial DAG: 0”). In sechs Verbesserungsschritten werden sechs Kanten eingefügt. Die Ergebnisse der beiden Aufrufe der Lern-Funktion werden zusätzlich in einem Grafik-Fenster gegenübergestellt, dabei werden jeweils die Veränderungen gegenüber dem ursprünglichen DAG (`testDAG`) hervorgehoben.



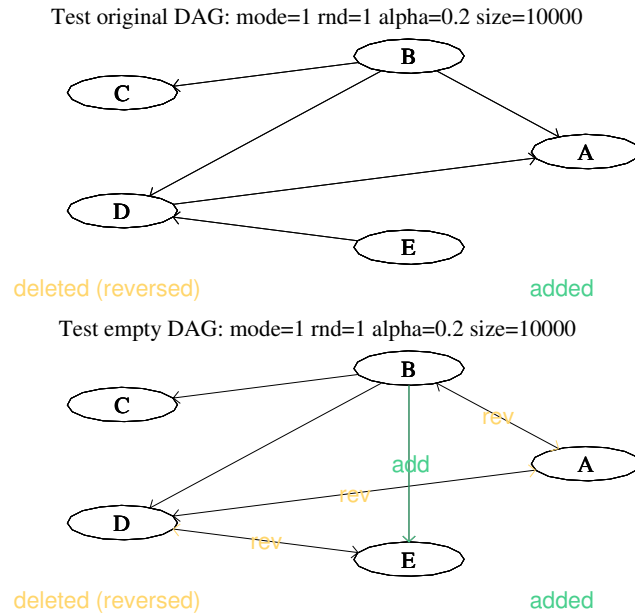


Abbildung 21: Grafische Ausgabe des Simulations-Tools

Bewertung des Ausgangs-DAG's $p(D S_{\text{testDAG}}^h)$ :	$e^{-20355.2188}$
Bewertung des verbesserten Ausgangs-DAG's:	— — — — (keine Verbesserung)
Bewertung des kantenlosen DAG's	$e^{-22588.0703}$
Bewertung des aus dem kantenlosen gelernten DAG's	$e^{-20356.0234}$

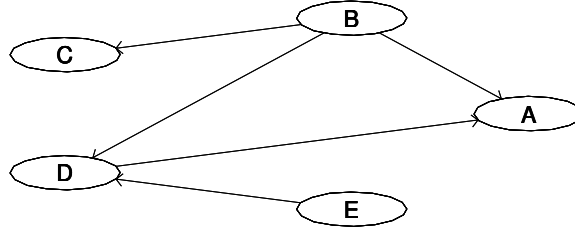
Es ist zu erkennen daß sich der gelernte DAG von dem Ausgangs-DAG in der Richtung dreier Kanten und einer zusätzlich eingefügten Kante unterscheidet. Das ursprüngliche Netz hatte die Dimension 12, das gelernte die Dimension 13 (alle Knoten haben zwei Ausprägungen) - es wird damit eine Abhängigkeit mehr dargestellt als tatsächlich vorhanden ist. Warum wurde die zusätzliche Kante eingefügt? Betrachten wir im oberen DAG die Dimension des Teilnetzes, bestehend aus den Knoten  $B$ ,  $D$  und  $E$ , so beträgt sie 6. Vergleichen wir das mit der Dimension desselben Teilnetzes **ohne** die zusätzliche Kante (add), so beträgt sie nur 5. In diesem Teilgraphen fehlt also eine bedingte Abhängigkeit, die durch die zusätzliche Kante versucht wurde darzustellen, damit stieg die Dimension dieses Teilnetzes aber um 2 auf 7 und insgesamt um 1 auf 13. Warum aber wurde statt die überzählige Kante einzufügen nicht die Kante  $D \rightarrow E$  umgedreht? Verfolgen wir den Ablauf der einzelnen Netzverbesserungsschritte, so sehen wir, daß die in ihrer Richtung verkehrte Kante  $D \rightarrow E$  die zuallererst eingefügte Kante ist. Zu diesem Zeitpunkt der Berechnung war aber nicht absehbar in welcher Richtung diese Kante letztendlich günstiger wäre; lediglich eine Abhängigkeit von  $D$  und  $E$  war erkennbar. Im vierten Schritt wurde dann die Kante  $B \rightarrow D$  richtig eingefügt. Nun aber fehlte noch eine Abhängigkeit in diesem Teilgraphen der Knoten  $B$ ,  $D$  und  $E$ , die durch die zusätzliche Kante (erfolgreich aber nicht optimal) versucht wurde darzustellen. Tatsächlich sind  $B$  und  $E$  ja auch abhängig<sup>30</sup> Um den Fehler zu diesem Zeitpunkt noch zu korrigieren wäre die gleichzeitige Betrachtung von zwei Kantenänderungen notwendig gewesen: dem Entfernen von  $B \rightarrow E$  und dem Umdrehen von  $D \rightarrow E$ ; das aber kann durch einen Greedy-Search-Algorithmus nicht geleistet werden. Daß es sich tatsächlich um ein Problem der Suchheuristik handelt erkennt man auch daran, daß die wahre Netzstruktur deutlich besser bewertet wurde als die gelernte. Wäre der Lernalgorithmus also während seiner Suche auf dieses wahre Netz gestoßen, wäre dieses auch das Netz seiner Wahl geworden.

### 10.2.3 Versuch 2: “Vorwissen” aus Daten (erhöhter Anteil)

Von Interesse bei dem im ersten Versuch verwendeten Verfahren, das eigentlich nicht vorhandene Vorwissen aus den Daten zu gewinnen und als Gegengewicht zur Vermeidung zu vieler Kanten zu benutzen, war der Anteil der für das Vorwissen herangezogenen Daten. Daher werden wir diesen Wert nun bei sonst gleichbleibenden Parametern variieren:

<sup>30</sup>Nicht untypischer Weise decken sich die Kanten des gelernten DAG's mit den Kanten des moralischen Graphen, gebildet zum Ausgangs-Graphen.

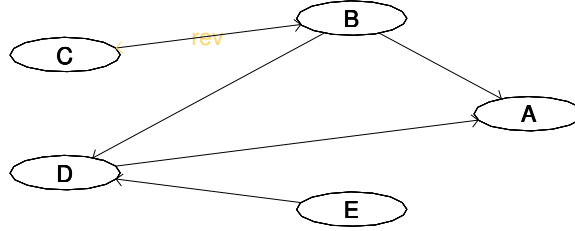
Test original DAG: mode=1 rnd=1 alpha=0.5 size=10000



deleted (reversed)

added

Test empty DAG: mode=1 rnd=1 alpha=0.5 size=10000



deleted (reversed)

added

Abbildung 22: Grafische Ausgabe des Simulations-Tools

Bewertung des Ausgangs-DAG's $p(D S_{\text{testDAG}}^h)$ :	$e^{-12719.1641}$
Bewertung des verbesserten Ausgangs-DAG's:	— — — — (keine Verbesserung)
Bewertung des kantenlosen DAG's	$e^{-14115.6992}$
Bewertung des aus dem kantenlosen gelernten DAG's	$e^{-12719.1602}$

```
> SIMULATE(dag=testDAG,mode=1,alpha=0.5,sampleSize=10000)
```

Die erzeugten bedingten Wahrscheinlichkeiten und damit auch der daraus abgeleitete Datensatz sind identisch denen des ersten Testlaufs. Auch bei der Prüfung des initialen DAG's (`testDAG`) ergibt sich keine Änderung. Wiederum wird der DAG unverändert belassen. Erst bei dem Versuch, diesen DAG beginnend mit einem kantenlosen DAG zurückzugewinnen, ergibt sich eine Veränderung:

```

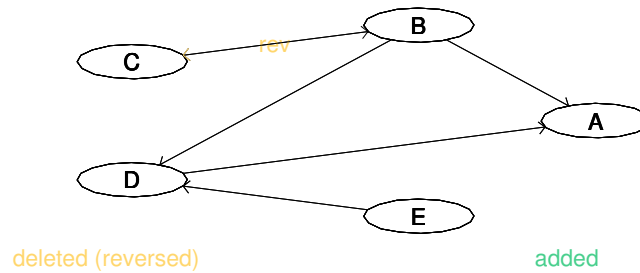
SEARCH
[ 0] INIT rate for initial DAG          p(D|S) = exp(  -14115.6992)
+-----+
[ 1] | ADD          E ---> D          p(D|S) = exp(  -13408.2734) |
[ 2] | ADD          D ---> A          p(D|S) = exp(  -12956.9336) |
[ 3] | ADD          B ---> A          p(D|S) = exp(  -12793.0586) |
[ 4] | ADD          B ---> D          p(D|S) = exp(  -12723.8945) |
[ 5] | ADD          C ---> B          p(D|S) = exp(  -12719.1602) |
+-----+
[ 6] STOP no improvement                p(D|S) = exp(  -12719.1602)

```

Wiederum wird zuerst eine Kante zwischen *E* und *D* eingefügt, diesmal aber in der richtigen Richtung. Es ist zu diesem Berechnungszeitpunkt im Grunde egal in welcher Richtung diese Kante eingefügt wird. Der Unterschied zum ersten Testlauf resultiert aus winzigen Unterschieden, die durch die veränderte Teilung des Datensatzes entstehen (siehe Abschnitt 6.7) und hier bei verändertem Vorwissen-Gewicht in die andere Richtung umgeschlagen sind.

Beide Netze unterscheiden sich nur in der Richtung einer Kante, die Dimension der Netze ist gleich. Die Bewertungen der beiden Netze sind ebenfalls fast identisch. Das errechnete Netz wurde sogar leicht besser bewertet. Warum aber wurde dann im ersten Testlauf die Kante *B* → *C* nicht ebenfalls umgedreht? Die Verwendete Methode legt dem einen Stein in den Weg. Eine zusätzliche Kante muß sich ja gegen das “Vorwissen” behaupten, daß sie überflüssig ist - erst danach würde mit einer erneuten Grundbewertung diese Kante dem Vorwissen hinzugerechnet. Diese Hürde aber nimmt die neue Kante *C* → *B* (nachdem testweise *B* → *C* entfernt wurde) in diesem Fall wegen dem so geringen Unterschied in der Bewertung der beiden Netze nicht.

Test original DAG: mode=1 rnd=1 alpha=0.0001 size=10000



Test empty DAG: mode=1 rnd=1 alpha=0.0001 size=10000

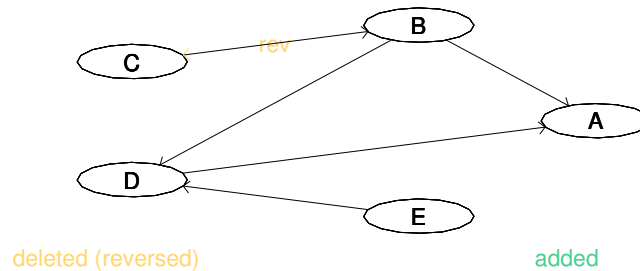


Abbildung 23: Grafische Ausgabe des Simulations-Tools

Bewertung des Ausgangs-DAG's $p(D S_{\text{testDAG}}^h)$ :	$e^{-25495.3672}$
Bewertung des verbesserten Ausgangs-DAG's:	$e^{-25495.3672}$
Bewertung des kantenlosen DAG's	$e^{-28252.5469}$
Bewertung des aus dem kantenlosen gelernten DAG's	$e^{-25495.3672}$

#### 10.2.4 Versuch 3: “Vorwissen” aus Daten (minimaler Anteil)

Mit einem weiteren Versuch nur das Vorwissen-Gewicht  $\alpha(\text{weight})$  zu verändern, wird die Problematik künstlich erzeugter Daten augenfällig:

```
> SIMULATE(dag=testDAG,mode=1,alpha=0.0001,sampleSize=10000)
```

Wir sehen hier erstmals, daß im ersten Lauf des Lernalgorithmus' auch das Ausgangsnetz verändert wurde. Beide Netze stellen aber dieselben Abhängigkeitsstrukturen dar und sind gleichwertig. Der Unterschied in der Bewertung ist winzig (erst nach der 4-ten Nachkommastelle sichtbar) und hier eher auf Rundungsfehler zurückzuführen. Die “Hürde”, diese Kante umzudrehen, ist wegen des unbedeutenden Vorwissen-Gewichts diesmal so niedrig, daß die Veränderung vorgenommen wird. Auch der eigentlich wichtige zweite Lauf des Lernalgorithmus' kommt zu demselben Ergebnis. Damit haben wir eine vollkommene Übereinstimmung. Trotzdem ist der Wert von  $\alpha=0.0001$  mit Vorsicht zu genießen, da er kein Korrektiv mehr gegen zu viele Kanten darstellt, was aber bei künstlich erzeugten Daten nicht problematisch ist - zumal bei großen Beobachtungsumfängen - jedoch bei echten Daten zu *overfitting* in der gelernten Netzstruktur führen kann und i.d.R. auch führen wird.

#### 10.2.5 Versuch 4: “Vorwissen” aus Daten (Verkleinerung des Datensatzes)

Verändern wir einmal den Beobachtungsumfang des Datensatzes, so ergibt sich auch für den Lernalgorithmus eine Problematik.

```
> SIMULATE(dag=testDAG,mode=1,alpha=0.2,sampleSize=1000)
```

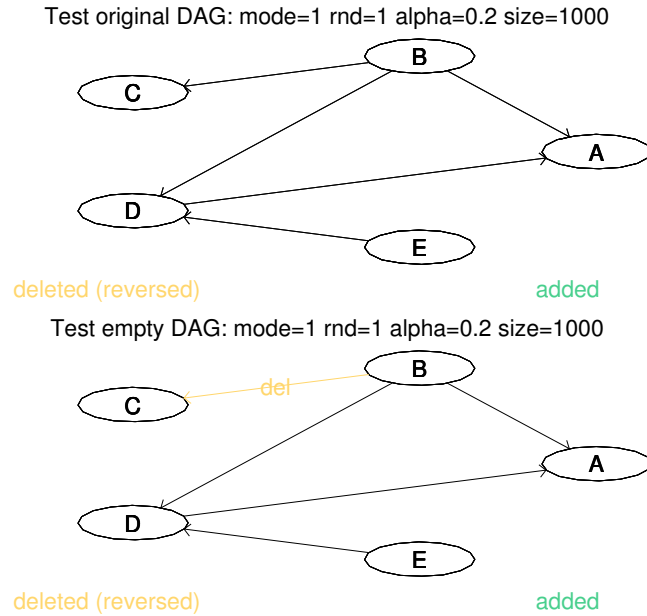


Abbildung 24: Grafische Ausgabe des Simulations-Tools

Bewertung des Ausgangs-DAG's $p(D S_{\text{testDAG}}^h)$ :	$e^{-2054.3542}$
Bewertung des verbesserten Ausgangs-DAG's:	— — — — (keine Verbesserung)
Bewertung des kantenlosen DAG's	$e^{-2271.6450}$
Bewertung des aus dem kantenlosen gelernten DAG's	$e^{-2054.4255}$

Eine Kante ( $B \rightarrow C$ ) wurde nicht eingefügt. Die eher schwache Abhängigkeit (auch vorher wurde diese Kante immer erst zuletzt eingefügt) wird in dem kleineren Datensatz und evtl. auch durch die ungenauere Repräsentation der vorgegebenen Wahrscheinlichkeiten nicht mehr in eine Kante umgesetzt. Sie wird im ersten Lauf aber aus dem Originalnetz auch nicht entfernt, da nur dem Einfügen einer Kante eine zusätzliche Hürde durch die Annahme der Überflüssigkeit dieser Kante aufgebürdet wird.

### 10.2.6 Versuch 5: “Vorwissen” aus Daten (Weitere Verkleinerung des Datensatzes)

Mit einer weiteren Verkleinerung des Datensatzes verschärft sich das Problem noch:

```
> SIMULATE(dag=testDAG,mode=1,alpha=0.2,sampleSize=100)
```

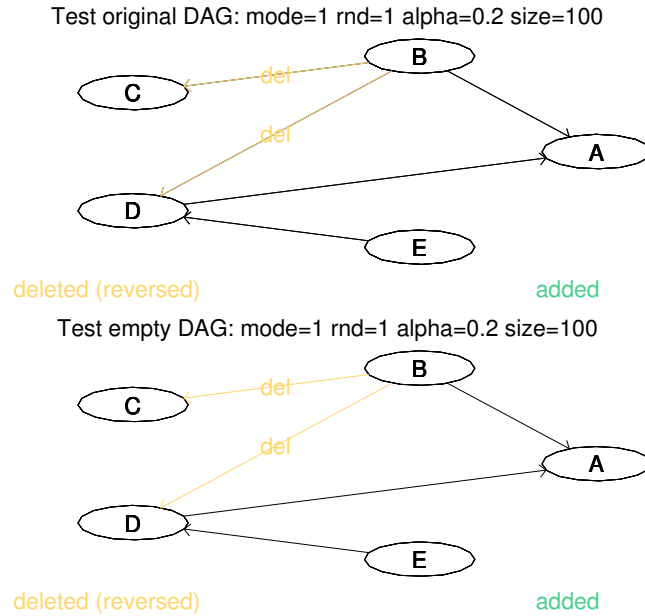


Abbildung 25: Grafische Ausgabe des Simulations-Tools

Bewertung des Ausgangs-DAG's $p(D S_{\text{testDAG}}^h)$ :	$e^{-199.3071}$
Bewertung des verbesserten Ausgangs-DAG's:	$e^{-198.2417}$
Bewertung des kantenlosen DAG's	$e^{-217.9348}$
Bewertung des aus dem kantenlosen gelernten DAG's	$e^{-198.2417}$

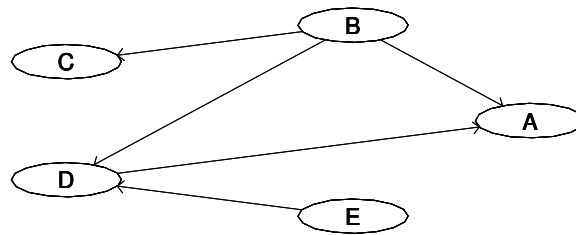
Zwei Kanten werden aus dem Originalnetz entfernt. Dieselben Kanten fehlen im vollständig gelernten Netz. Die Netze stimmen also überein. Das Fehlen der Kanten beruht hier nicht auf einem Fehler in der Lernheuristik, sondern auf den zu schwachen Abhängigkeiten und der ungenaueren Repräsentation der echten Wahrscheinlichkeiten durch die Daten, je weniger Daten zur Verfügung stehen.

### 10.2.7 Versuch 6: Mit Bayes'schem "Unwissen" (geringes Gewicht)

In einer weiteren Testreihe soll auch das zweite Verfahren zum Umgang mit fehlendem Vorwissen - der Umsetzung in Bayes'sches "Unwissen" gezeigt werden:

```
> SIMULATE(dag=testDAG,mode=2,alpha=0.001,sampleSize=10000)
```

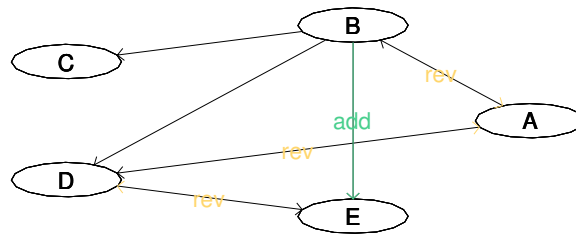
Test original DAG: mode=2 rnd=1 alpha=0.001 size=10000



deleted (reversed)

added

Test empty DAG: mode=2 rnd=1 alpha=0.001 size=10000



deleted (reversed)

added

Abbildung 26: Grafische Ausgabe des Simulations-Tools

Bewertung des Ausgangs-DAG's $p(D S_{\text{testDAG}}^h)$ :	$e^{-25484.3477}$
Bewertung des verbesserten Ausgangs-DAG's:	— — — — (keine Verbesserung)
Bewertung des kantenlosen DAG's	$e^{-28255.5547}$
Bewertung des aus dem kantenlosen gelernten DAG's	$e^{-25487.7461}$

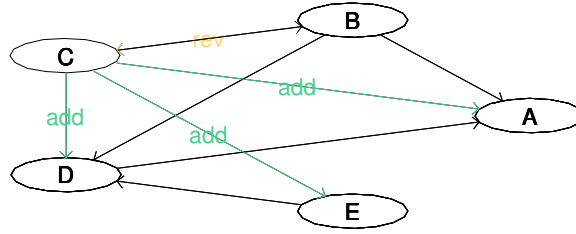
Das Gewicht des Vorwissens wurde klein gewählt. Das Ergebnis ist identisch dem des anderen Verfahrens.

### 10.2.8 Versuch 7: Mit Bayes'schem "Unwissen" (erhöhtes Gewicht)

Zum Vergleich wählen wir einen großen Wert für `alpha(weight)`:

```
> SIMULATE(dag=testDAG,mode=2,alpha=0.1,sampleSize=10000)
```

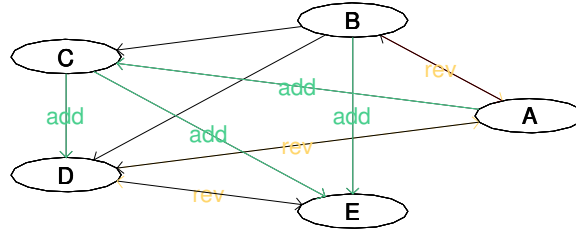
Test original DAG: mode=2 rnd=1 alpha=0.1 size=10000



deleted (reversed)

added

Test empty DAG: mode=2 rnd=1 alpha=0.1 size=10000



deleted (reversed)

added

Abbildung 27: Grafische Ausgabe des Simulations-Tools

Bewertung des Ausgangs-DAG's $p(D S_{\text{testDAG}}^h)$ :	$e^{-26561.5234}$
Bewertung des verbesserten Ausgangs-DAG's:	$e^{-26466.6797}$
Bewertung des kantenlosen DAG's	$e^{-28951.4453}$
Bewertung des aus dem kantenlosen gelernten DAG's	$e^{-26478.3125}$

Wie schon in Abschnitt 8.2 gezeigt wurde, sehen wir auch hier, daß durch die Verwischung der im Datensatz repräsentierten echten Wahrscheinlichkeiten, durch das willkürlich gewählte Vorwissen und die unangemessene Vergrößerung des Gesamtbeobachtungsumfangs von 10000 Daten um 1000 imaginäre Vorwissen-Beobachtungen nicht etwa weniger sondern mehr Kanten (Abhängigkeiten) gefunden werden. Das dieses aber auch hier, bei fast perfekt zu einem bestimmten Netz passenden Daten geschieht, zeigt wie ein zu hoher *UserSampleSize*-Wert geeignet ist, selbst geringste numerisch bedingte Schein-Abhängigkeiten in den Daten in Kanten umzusetzen.

## 10.3 Berechnungen mit dem Kredit-Datensatz

### 10.3.1 Der Kredit-Datensatz

Merkmal	Aus- präg.	Beschreibung	Merkmal	Aus- präg.	Beschreibung
Kredit	0	Rückzahlung nicht ok	in jetziger Wohnung <sup>1)</sup>	1	< 1 Jahr
	1	Rückzahlung ok		2	$\geq 1$ Jahr
laufendes Konto <sup>1)</sup>	1	kein Kontostand	Vermögen	1	Haus & Grund
	2	$\geq 0$ DM oder		2	Bauspar / Lebensvers.
	3	Gehaltskonto (min.1J.)		3	PKW, sonstiges
		kein laufendes Konto		4	nicht ermittelbar / kein
Laufzeit <sup>1)</sup>	1	$\leq 12$ Monate	Wohnung	1	Mietwohnung
(des Darlehens)	2	$> 12, \leq 24$ Monate		2	Eigentumswohnung
	3	$> 24$ Monate		3	kostenlose Wohnung
Moral <sup>1)</sup>	1	bish. keine Kredit / ok	Höhe <sup>1)</sup>	1	$\leq 2000$ DM
(bish. Rück- zahlungsmoral)	2	bei der Bank bish. ok		2	$> 2000, \leq 4000$ DM
	3	sonst		2	$> 4000$ DM
Nutzung <sup>1)</sup>	1	PKW	Alter <sup>1)</sup>	1	$\leq 25$ Jahre
(Verwendung)	2	Einrichtung		2	$> 25, \leq 35$ Jahre
	3	Ausbildung, Betrieb		3	$> 35, \leq 60$ Jahre
	4	Urlaub, sonstiges		4	$> 60$ Jahre
weitere Ratenkredite <sup>1)</sup>	1	ja	bisherige Ratenkredite <sup>1)</sup>	1	max. einen
	2	nein		2	mehr als einen
Sparkonto <sup>1)</sup>	1	$< 1000$ DM	Bürgen	1	keine
	2	$\geq 1000$ DM		2	Mitantragsteller
	3	nicht ermittelbar / kein Sparkonto		3	Bürge
Beschäfti- gungszeit <sup>1)</sup>	1	nicht / $< 1$ Jahr	Personen (unterhalts- berechtigt)	1	0 bis 2
	2	$\geq 1, < 7$ Jahre		2	3 und mehr
	3	$\geq 7$ Jahre	Telefon <sup>2)</sup>	1	nein
Ratenhöhe <sup>1)</sup>	1	$> 35$ %		2	ja
(in % des Einkommens)	2	$\geq 20, \leq 35$ %			
	3	$< 20$ %			
Familienst./ Geschlecht	1	männl. geschieden	Beruf	1	kein / nicht seßhaft
	2	weibl. verh./gesch.		2	ungelernt + seßhaft
	3	männl. ledig		3	Facharbeiter / Beamter
	4	männl. verh./verw.		4	Führungskraft / selbstst. / gehobener Beamter
	5	weibl. ledig			
Gastarbeiter <sup>2)</sup>	1	ja			
	2	nein			

<sup>1)</sup> Die Merkmale wurden teilweise (anders) kategorisiert (als in der Originalquelle)

<sup>2)</sup> Der Datensatz stammt aus den 70'er Jahren und wurde nur gemäß <sup>1)</sup> verändert  
und insbesondere nicht auf *political correctness* hin untersucht

Der Kredit-Datensatz enthält 1000 Fälle (Beobachtungen), in denen jeweils alle 21 Merkmale beobachtet wurden, der Datensatz ist also vollständig und entspricht damit unseren Bedingungen an einen Datensatz<sup>31</sup>.

Quellen: [Fahrmeir / Tutz (1994)] und [Fahrmeir / Hamerle / Tutz (1996)], sowie unter

<http://www.stat.uni-muenchen.de/data-sets/credit/credit.html>

### 10.3.2 Versuch 1: “Vorwissen” aus Daten (geringer Anteil)

Wie bei den mittels Sampling erzeugten Daten hier zunächst ein Versuch mit aus 20% der Daten gewonnenem “Vorwissen” ( $\alpha=0.2$ ).

<sup>31</sup>Problematisch ist allerdings, daß die Fälle des Kreditdatensatzes vorselektiert wurden, da nur vergebene Kredite enthalten sind, aber nicht jeder Antragssteller einen Kredit erhalten haben dürfte.



```

> learnDAG1 _ LEARN(dag=kreditDAG,dataset=kredit,alpha=0.2)
+-----+
|      FIND / IMPROVE NETSTRUCTUR OF A BAYESIAN (CAUSAL) NETWORK      |
|      BY GIVEN DATA (AND USER-KNOWLEDGE)                             |
+-----+

START
dataset ..... : sorting (2 b faster)
cases in dataset ..... : 1000
different cases in dataset ..... : 993 (-0.7 %)
number of edges in initial DAG .. : 0
possible changes excluded by user : 0
testing DAG only with data ..... : splitting data
splitting data by ..... : 0.800:0.200 (data:pseudo-alpha)
search mode ..... : retractions allowed
possible changes are ..... : add del reverse edges
max. count of changes ..... : 100
term for net-prior ..... : is not used
calculate for D(ata) & S(tructur) : p(D|S)
max. shown improves ..... : show only changes

SEARCH
[ 0] INIT rate for initial DAG                p(D|S) = exp( -13604.7637)
+-----+
[ 1] | ADD      wohnung ---> vermoeegen      p(D|S) = exp( -13433.4043) |
[ 2] | ADD      bish.raten ---> moral         p(D|S) = exp( -13269.2197) |
[ 3] | ADD      laufzeit ---> hoehe          p(D|S) = exp( -13112.0752) |
[ 4] | ADD      telefon ---> beruf           p(D|S) = exp( -13037.4902) |
[ 5] | ADD      ratenhoehe ---> hoehe        p(D|S) = exp( -12973.1250) |
[ 6] | ADD      alter ---> beschzeit        p(D|S) = exp( -12919.8643) |
[ 7] | ADD      kredit ---> lfd.konto       p(D|S) = exp( -12871.6152) |
[ 8] | ADD      wohnung ---> alter          p(D|S) = exp( -12829.0840) |
[ 9] | ADD      alter ---> fam.geschl       p(D|S) = exp( -12792.1934) |
[10] | ADD      fam.geschl ---> personen     p(D|S) = exp( -12757.9199) |
[11] | ADD      vermoeegen ---> laufzeit    p(D|S) = exp( -12724.8701) |
[12] | ADD      weit.raten ---> moral       p(D|S) = exp( -12694.6758) |
[13] | ADD      beschzeit ---> beruf        p(D|S) = exp( -12667.8418) |
[14] | ADD      beruf ---> vermoeegen      p(D|S) = exp( -12637.3711) |
[15] | ADD      hoehe ---> nutzung         p(D|S) = exp( -12615.0186) |
[16] | ADD      moral ---> kredit          p(D|S) = exp( -12595.9277) |
[17] | ADD      beschzeit ---> jetzt.wohn  p(D|S) = exp( -12576.8047) |
[18] | ADD      laufzeit ---> kredit       p(D|S) = exp( -12563.8984) |
[19] | ADD      alter ---> telefon        p(D|S) = exp( -12552.1162) |
[20] | ADD      vermoeegen ---> buergen    p(D|S) = exp( -12540.5420) |
[21] | ADD      kredit ---> sparkonto      p(D|S) = exp( -12530.5039) |
[22] | ADD      alter ---> personen       p(D|S) = exp( -12518.4424) |
[23] | ADD      wohnung ---> jetzt.wohn  p(D|S) = exp( -12509.3770) |
[24] | ADD      telefon ---> hoehe        p(D|S) = exp( -12497.2197) |
[25] | ADD      laufzeit ---> gastarb      p(D|S) = exp( -12489.8877) |
[26] | ADD      beschzeit ---> ratenhoehe  p(D|S) = exp( -12482.6611) |
[27] | ADD      nutzung ---> lfd.konto    p(D|S) = exp( -12474.6719) |
[28] | ADD      alter ---> bish.raten     p(D|S) = exp( -12469.1992) |
[29] | ADD      nutzung ---> weit.raten   p(D|S) = exp( -12464.0967) |
[30] | ADD      gastarb ---> nutzung      p(D|S) = exp( -12458.7119) |
[31] | ADD      hoehe ---> sparkonto      p(D|S) = exp( -12453.4033) |
[32] | ADD      wohnung ---> fam.geschl   p(D|S) = exp( -12444.6680) |
[33] | ADD      personen ---> gastarb     p(D|S) = exp( -12440.8037) |
[34] | ADD      gastarb ---> moral        p(D|S) = exp( -12435.5537) |
[35] | ADD      vermoeegen ---> gastarb   p(D|S) = exp( -12429.5898) |
[36] | ADD      gastarb ---> kredit       p(D|S) = exp( -12424.3672) |
[37] | REV      personen </-> fam.geschl  p(D|S) = exp( -12417.3750) |
[38] | ADD      laufzeit ---> nutzung    p(D|S) = exp( -12408.5947) |
[39] | ADD      hoehe ---> bish.raten    p(D|S) = exp( -12405.3105) |
[40] | ADD      beschzeit ---> kredit     p(D|S) = exp( -12396.1895) |
[41] | ADD      telefon ---> laufzeit    p(D|S) = exp( -12392.6973) |
[42] | ADD      fam.geschl ---> beschzeit p(D|S) = exp( -12384.1865) |
[43] | ADD      personen ---> beruf      p(D|S) = exp( -12378.7705) |
[44] | ADD      lfd.konto ---> buergen    p(D|S) = exp( -12373.9111) |
+-----+
[45] STOP no improvement                p(D|S) = exp( -12373.9111)

```

An der Ausgabe des Lernalgorithmus' abzulesen ist u.a., daß sich unter den 1000 Beobachtungen im Kredit-Datensatz 993 unterschiedliche Fälle befinden. Die vom Lernalgorithmus zunächst vorgenommene Gruppierung der Daten hat damit kaum eine Effizienzsteigerung zur Folge. Unter den rund 2,6 Mio theoretisch möglichen unterschiedlichen Ausprägungskombinationen ist damit nur eine verschwindend geringe Anzahl tatsächlich ver-

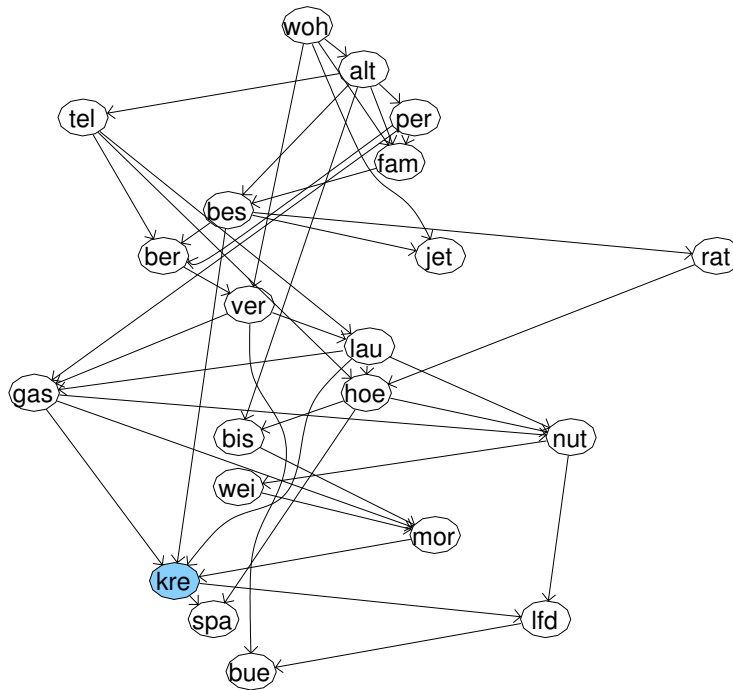


Abbildung 28: Grafische Ausgabe des Lernalgorithmus'

Bewertung des kantenlosen DAG's	$e^{-13604.7637}$
Bewertung des aus dem kantenlosen gelernten DAG's	$e^{-12373.9111}$

treten (0,000038%). Rund 2,6 Mio beträgt auch die Anzahl der Zellen einer gemeinsamen Wahrscheinlichkeits- oder auch Kontingenztafel aller 21 Merkmale. Aber eben nur zu 993 dieser Zellen liegen Daten vor und zu den allermeisten dieser 993 Zellen liegt nur genau eine Beobachtung vor. Wir sehen also wie unmöglich es wäre Abhängigkeiten unter **allen** Merkmalen aus diesen Daten abzuleiten. Zudem wäre das Ergebnis einer Rechnung mit Tafeln solcher Größe erst nach geraumer Zeit zu erwarten.

### 10.3.3 Versuch 2: "Vorwissen" aus Daten (erhöhter Anteil)

Äquivalent dem zweiten Testlauf mit Daten-Sampling variieren wir nun den Anteil der für das "Vorwissen" genutzten Daten auf 50% ( $\alpha=0.5$ ).

```
> learnDAG2 _ LEARN(dag=kreditDAG,dataset=kredit,alpha=0.5)
```

37 Kanten werden in das gelernte Netz `learnDAG2` gegenüber dem kantenlosen Netz `kreditDAG` eingefügt. Erwartungsgemäß sinkt mit der Verschiebung des Teilungsverhältnisses der Daten zugunsten des "Vorwissens" die Anzahl der Kanten im erlernten DAG `learnDAG2` gegenüber `learnDAG1`. Zum einen stehen, um Abhängigkeiten zu verifizieren weniger Daten zur Verfügung; zum anderen wird der Unabhängigkeitsannahme (Kante überflüssig) in Form des Vorwissens mehr Gewicht beigemessen.

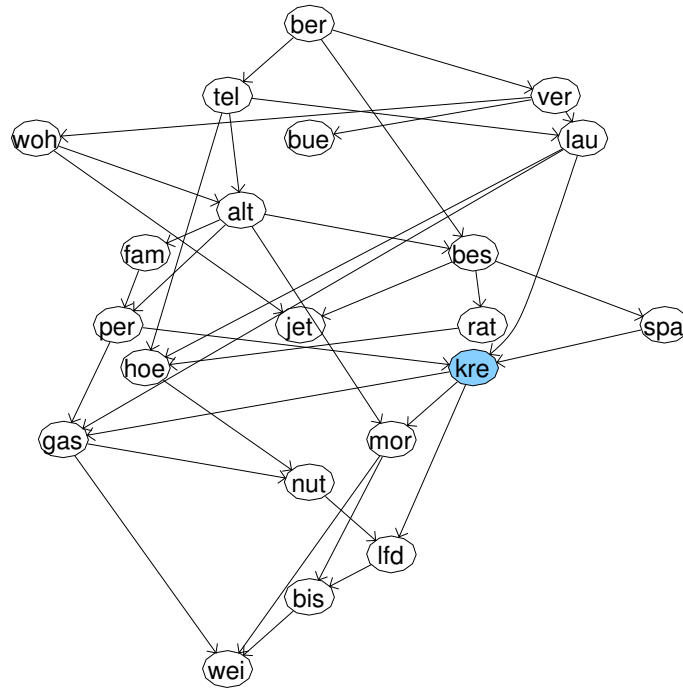


Abbildung 29: Grafische Ausgabe des Lernalgorithmus'

Bewertung des kantenlosen DAG's	$e^{-8496.6641}$
Bewertung des aus dem kantenlosen gelernten DAG's	$e^{-7722.7246}$

#### 10.3.4 Vergleich von Versuch 1 und Versuch 2

Die gelernten Netze aus Versuch 1 und Versuch 2 unterscheiden sich auf den ersten Blick deutlich. Gegenüber dem ersten Versuch sind im zweiten DAG 18 Kanten vollkommen verändert, 10 sind umgedreht worden. Das zweite DAG hat insgesamt 6 Kanten weniger als das erste. Im Vergleich mit der Ausgabe des Lernalgorithmus' zum ersten Versuch kann man erkennen, daß von den dort zuletzt gefundenen 7 Kanten im zweiten DAG 6 Kanten (ersatzlos) fehlen. Damit sind mit der Erhöhung des Vorwissen-Gewichts, von der Kante *telefon* → *laufzeit* abgesehen, die **schwächsten** 6 Abhängigkeiten fallen gelassen worden. Die restlichen 6 fehlenden Kanten werden wieder durch 6 neue Kanten ausgeglichen. Die Änderungen in diesem Bereich sind weniger im Informationsgehalt des DAG's als vielmehr nur in der Struktur des DAG's begründet. So finden wir im ersten DAG z.B. die Struktur:

$$gastarbeiter \rightarrow moral \leftarrow weit.raten$$

Im zweiten DAG sieht die Struktur dieser drei Knoten so aus:

$$gastarbeiter \rightarrow weit.raten \leftarrow moral$$

Die zugehörigen Teilgraphen im moralischen Graphen (siehe Abschnitt 2.3.1) sind aber identisch, dementsprechend auch die Abhängigkeiten unter diesen drei Knoten.

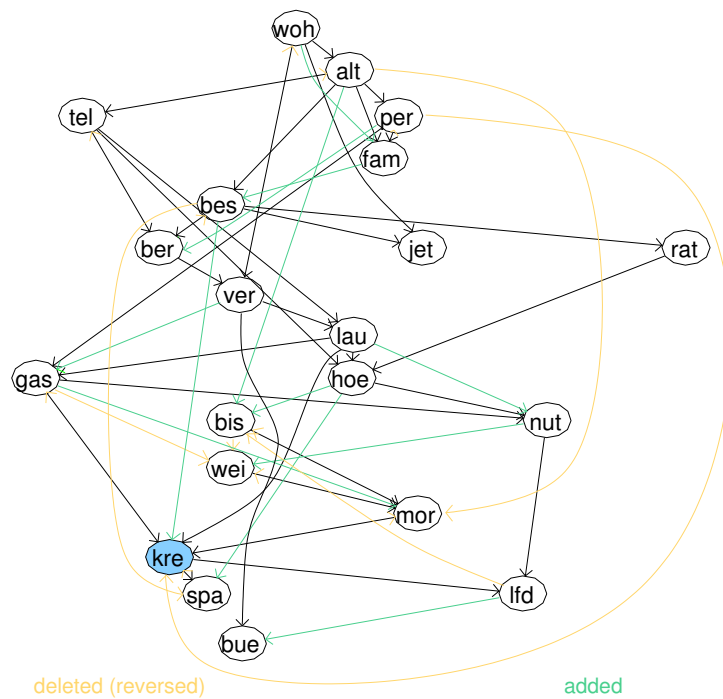


Abbildung 30: Grafischer Vergleich von Versuch 1 mit Versuch 2

	Kanten im DAG	
	43	
Gelöschte Kanten gegenüber dem Vergleichs-DAG	6	
Hinzugefügte Kanten gegenüber dem Vergleichs-DAG	12	
Umgedrehte Kanten gegenüber dem Vergleichs-DAG	10	

### 10.3.5 Versuch 3: Mit Bayes'schem "Unwissen" (geringes Gewicht)

In einem weiteren Testlauf erzeugen wir nun selbst Tafeln, die unser Unwissen über die wahren Verteilungen widerspiegeln. Damit benutzen wir die zweite Variante um mit fehlendem Vorwissen umzugehen (siehe Abschnitt 8.2).

```
> learnDAG3 _ LEARN(dag=kreditDAG,dataset=kredit,tables=dagzerotabs,alpha=0.01)
```

```
+-----+
|      FIND / IMPROVE NETSTRUCTUR OF A BAYESIAN (CAUSAL) NETWORK      |
|      BY GIVEN DATA (AND USER-KNOWLEDGE)                             |
+-----+

START
dataset ..... : sorting (2 b faster)
cases in dataset ..... : 1000
different cases in dataset ..... : 993 (-0.7 %)
number of edges in initial DAG .. : 0
possible changes excluded by user : 0
testing DAG with alpha and data . : weight alpha by sample-size
sample-size ..... : 10
search mode ..... : retractions allowed
possible changes are ..... : add del reverse edges
max. count of changes ..... : 100
term for net-prior ..... : is not used
calculate for D(ata) & S(tructur) : p(D|S)
max. shown improves ..... : show only changes

SEARCH
[ 0] INIT rate for initial DAG                p(D|S) = exp( -17113.7031)
+-----+
[ 1] | ADD    vermoeegen ---> wohnung          p(D|S) = exp( -16908.6680) |
[ 2] | ADD    moral ---> bish.raten            p(D|S) = exp( -16707.0742) |
[ 3] | ADD    laufzeit ---> hoehe             p(D|S) = exp( -16517.8750) |
[ 4] | ADD    beruf ---> telefon              p(D|S) = exp( -16428.7676) |
[ 5] | ADD    ratenhoehe ---> hoehe            p(D|S) = exp( -16364.3477) |
[ 6] | ADD    beschzeit ---> alter            p(D|S) = exp( -16305.3672) |
[ 7] | ADD    kredit ---> lfd.konto           p(D|S) = exp( -16248.1660) |
[ 8] | ADD    fam.geschl ---> personen         p(D|S) = exp( -16209.9141) |
[ 9] | ADD    alter ---> fam.geschl           p(D|S) = exp( -16172.8457) |
[10] | ADD    vermoeegen ---> beruf            p(D|S) = exp( -16136.4297) |
[11] | ADD    hoehe ---> vermoeegen           p(D|S) = exp( -16103.0947) |
[12] | ADD    moral ---> weit.raten           p(D|S) = exp( -16071.1895) |
[13] | ADD    beruf ---> beschzeit            p(D|S) = exp( -16042.5391) |
[14] | ADD    kredit ---> moral               p(D|S) = exp( -16021.0234) |
[15] | ADD    beschzeit ---> jetzt.wohn       p(D|S) = exp( -15999.7266) |
[16] | ADD    hoehe ---> nutzung              p(D|S) = exp( -15979.4668) |
[17] | ADD    wohnung ---> alter              p(D|S) = exp( -15961.9316) |
[18] | ADD    kredit ---> laufzeit            p(D|S) = exp( -15948.9707) |
[19] | ADD    vermoeegen ---> buergen         p(D|S) = exp( -15938.9258) |
[20] | ADD    kredit ---> sparkonto           p(D|S) = exp( -15929.1475) |
[21] | REV    alter </-> beschzeit            p(D|S) = exp( -15920.6377) |
[22] | ADD    buergen ---> gastarb            p(D|S) = exp( -15913.3145) |
[23] | ADD    wohnung ---> jetzt.wohn         p(D|S) = exp( -15907.4355) |
[24] | ADD    alter ---> personen             p(D|S) = exp( -15902.6934) |
[25] | ADD    laufzeit ---> gastarb           p(D|S) = exp( -15899.2129) |
[26] | ADD    weit.raten ---> bish.raten      p(D|S) = exp( -15897.7822) |
[27] | ADD    kredit ---> weit.raten         p(D|S) = exp( -15897.1797) |
[28] | ADD    gastarb ---> nutzung            p(D|S) = exp( -15896.8252) |
[29] | REV    moral </-> kredit               p(D|S) = exp( -15896.8242) |
+-----+
[30] STOP no improvement                      p(D|S) = exp( -15896.8242)
```

Mit dem Wert  $\alpha=0.01$  geben wir dem "Vorwissen" ein Gewicht entsprechend 10 Beobachtungen ( $1000 \cdot 0.01 = 10$ ). Es entsteht damit ein Gesamtbeobachtungsumfang (Daten und Vorwissen) von 1010 Beobachtungen. Es werden mit einer Anzahl von 27 weniger Kanten gefunden als in den beiden Versuchen zuvor. Ist die Gewichtung des Vor-(Un-)Wissens zu hoch oder zu niedrig (oder genau richtig)? Zum einen bedenke man, wie klein die Werte in einer einzelnen  $\alpha_i$ -Tafel werden sobald der Knoten  $X_i$  Eltern bekommt, wenn die Gesamtsumme aller Zellen dieser Tafel nur 10 beträgt. Schnell sinken die Werte einzelner Zellen unter 1 ab (siehe Abschnitt 8.2.1 - 8.2.3). Zum anderen ist die Erhöhung des Gesamtbeobachtungsumfangs um 10 - eigentlich ungerechtfertigte - Beobachtungen bei (nur) 1000 echten Beobachtungen schon nicht mehr ganz bedeutungslos.

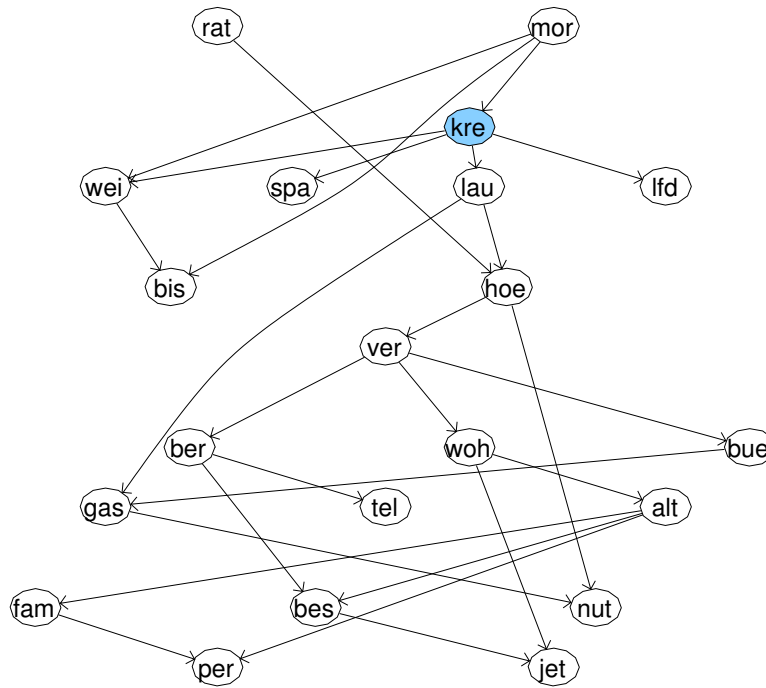


Abbildung 31: Grafische Ausgabe des Lernalgorithmus'

Bewertung des kantenlosen DAG's	$e^{-17113.7031}$
Bewertung des aus dem kantenlosen gelernten DAG's	$e^{-15896.8242}$

### 10.3.6 Versuch 4: Mit Bayes'schem "Unwissen" (minimales Gewicht)

Um den sich schon im letzten Versuch andeutenden Effekt zu verdeutlichen, nehmen wir hier zunächst eine Verringerung des Vorwissen-Gewichts auf  $\alpha=0.001 \equiv$  **einer** Beobachtung vor.

```
learnDAG4 _ LEARN(dag=kreditDAG,dataset=kredit,tables=dagzerotabs,alpha=0.001)
```

22 Kanten werden gelernt. Das aber scheint, auch ohne die beste Netzstruktur zu kennen, angesichts der ersten beiden Versuche zu wenig zu sein. Das Gegengewicht gegen die Daten wird kleiner, dennoch sinkt die Kantenanzahl. Die Werte der einzelnen  $\alpha$ -Tafel-Zellen sind hier sicher zu klein (siehe Abschnitt 8.2.1 - 8.2.3) Die Erhöhung des Gesamtbeobachtungsumfangs um 1 zu den 1000 Beobachtungen im Kredit-Datensatz ist hier dagegen als unproblematisch gering anzusehen.

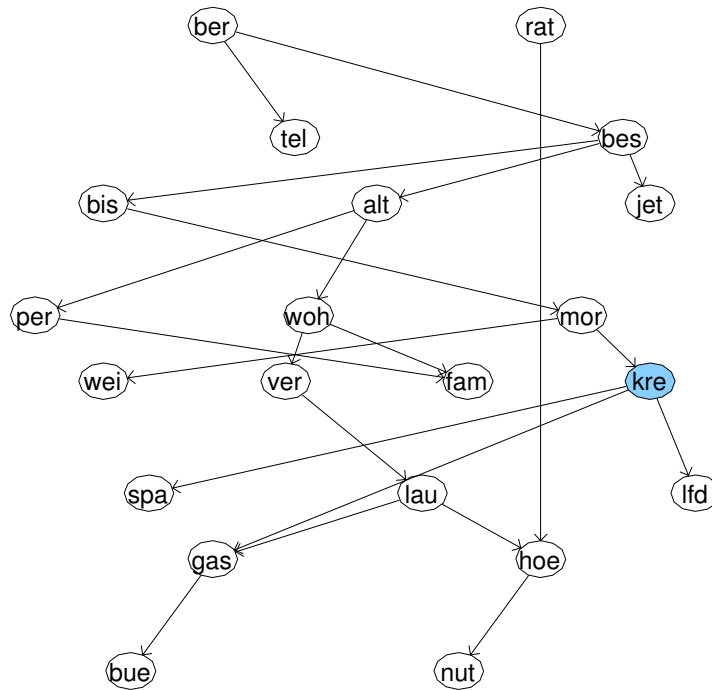


Abbildung 32: Grafische Ausgabe des Lernalgorithmus'

Bewertung des kantenlosen DAG's	$e^{-17112.2012}$
Bewertung des aus dem kantenlosen gelernten DAG's	$e^{-16052.5811}$

### 10.3.7 Vergleich von Versuch 3 und Versuch 4

Die hier verglichenen Netze unterscheiden sich weitaus deutlicher in ihrem Informationsgehalt, als die zuvor verglichenen Netze aus Versuch 1 und Versuch 2. Insgesamt hat das Netz aus Versuch 4 fünf Kanten weniger. Es fehlen aber 9 Kanten des Netzes aus Versuch 3. Von den im Netz aus Versuch 4 hinzugekommenen vier Kanten ersetzt jedoch keine eine der neun fehlenden Kanten, bezogen auf den moralischen Graphen. Die Wahl der *UserSampleSize* hat hier also nicht nur Auswirkungen auf die Kantenanzahl, sondern auch auf den Informationsgehalt - die Abhängigkeitsstrukturen im gelernten Netz.

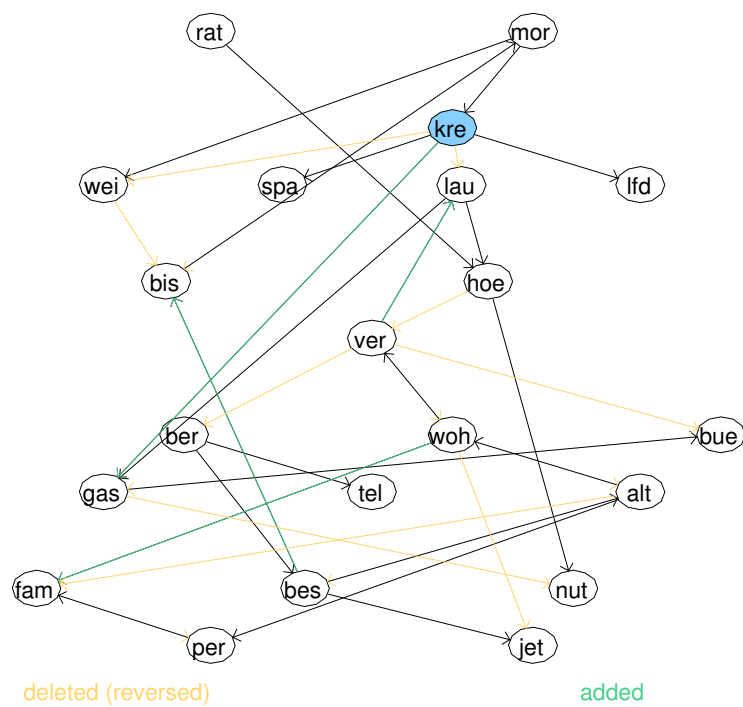


Abbildung 33: Grafischer Vergleich von Versuch 4 mit Versuch 3

	Kanten im DAG	22
Gelöschte Kanten gegenüber dem Vergleichs-DAG	9	
Hinzugefügte Kanten gegenüber dem Vergleichs-DAG	4	
Umgedrehte Kanten gegenüber dem Vergleichs-DAG	6	



### 10.3.8 Versuch 5: Mit Bayes'schem “Unwissen” (rel. hohes Gewicht)

Mit einer Erhöhung des Vorwissen-Gewichts auf  $\alpha=0.1 \equiv 100$  Beobachtungen testen wir nun den entgegengesetzten Fall gegenüber dem letzten Testlauf.

```
learnDAG4 _ LEARN(dag=kreditDAG,dataset=kredit,tables=dagzerotabs,alpha=0.1)
```

Diesmal werden 49 Kanten gelernt. Das aber scheint wiederum zu viel zu sein (*overfitting*). Indem wir unserem “Unwissen” ein Gewicht von 10% der Daten verleihen, wird der Gesamtbeobachtungsumfang ungerechtfertigt stark erhöht und damit die aus den Daten ableitbaren Abhängigkeiten verzerrt und das nicht unbedingt in Richtung Unabhängigkeit, wie an der Kantenanzahl abzulesen ist.

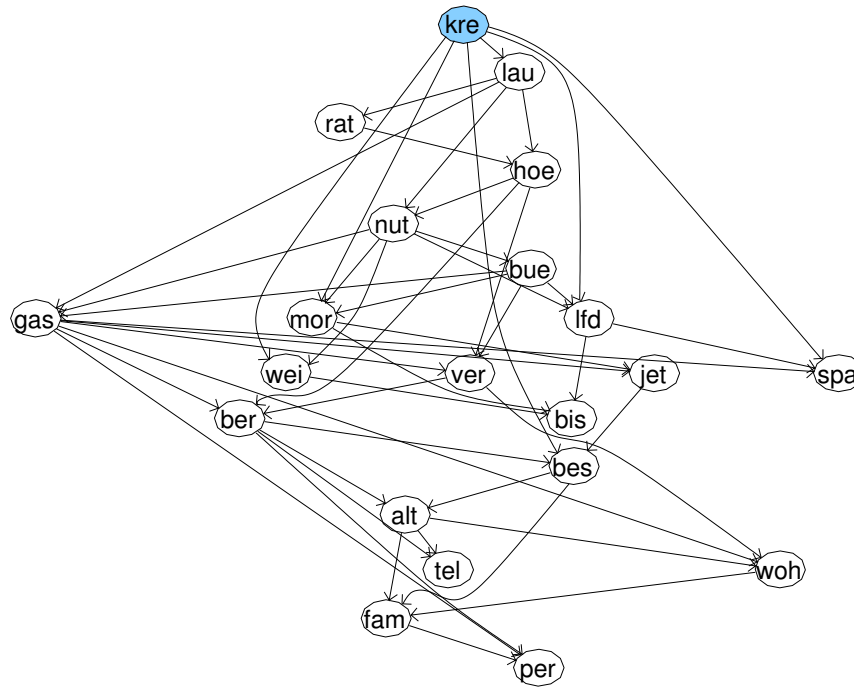


Abbildung 34: Grafische Ausgabe des Lernalgorithmus'

Bewertung des kantenlosen DAG's	$e^{-17552.6523}$
Bewertung des aus dem kantenlosen gelernten DAG's	$e^{-16164.5723}$

## 10.4 Daten-Sampling

### 10.4.1 Verfahren zum Daten-Sampling

Um von zufälligen Schwankungen nicht freie Daten zu erzeugen benutzen wir folgendes Verfahren:  
Haben wir eine Wahrscheinlichkeitsverteilung zu einem Merkmal vorgegeben zB:

$p(A)$

	$a_1$	$a_2$	$a_3$
Wahrscheinlichkeitsverteilung:	0.2	0.5	0.3
Einteilung in Segmente:	$\leq 0.2$	$> 0.2 \text{ und } \leq 0.7$	$> 0.7 \text{ und } \leq 1$

so erzeugen wir eine Zufallszahl im Intervall  $]0,1]$  und entscheiden uns für die Ausprägung des Merkmals, in dessen Segment die Zufallszahl gehört. Lautet die Zufallszahl 0.4532, so entscheiden wir uns bei der konkret zu erzeugenden Beobachtung für die Ausprägung  $a_2$  des Merkmals  $A$ , da  $0.2 < 0.4532 \leq 0.7$ . Für jede zu erzeugende Beobachtung beginnen wir mit diesem Verfahren bei den elternlosen Knoten und vergeben dann schrittweise die Ausprägungen der Merkmale deren Elternausprägungen bereits erzeugt wurden, so daß jeweils die bedingte Wahrscheinlichkeitsverteilung  $p(X_i | \mathbf{Pa}_i, S^h)$  für die zufällige Erzeugung der Ausprägungen in der beschriebenen Art herangezogen werden können. Dieses Verfahren wird für einen Datensatz entsprechend oft des gewünschten Beobachtungsumfangs wiederholt.

### 10.4.2 Versuch 1: $B \rightarrow A \rightarrow C$

Das Vorgegebene Netz ist  $B \rightarrow A \rightarrow C$ . Zu diesem Netz gibt es aber zwei äquivalente Netze, die exakt die gleichen Abhängigkeiten abbilden. Diese Netze sind:  $B \leftarrow A \leftarrow C$  und  $B \leftarrow A \rightarrow C$ . Bei der Bestimmung der Treffer müssen diese beiden Netze berücksichtigt werden. Es werden jeweils 1000 verschiedene Datensätze erzeugt und zu jedem eine Netzstruktur gelernt.

**Vorwissen aus 20% der Daten**

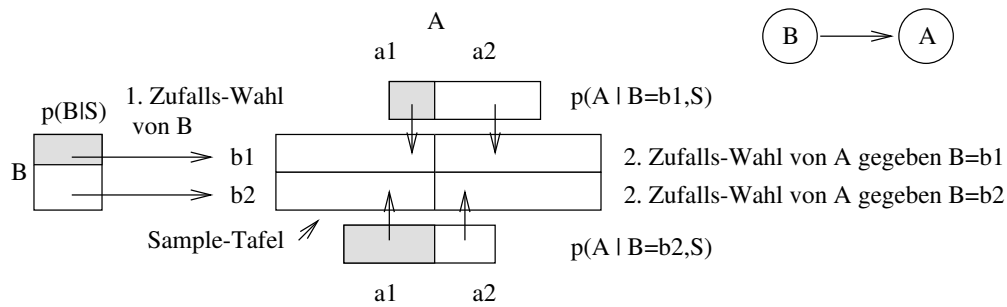


Abbildung 35: Sampling

Netz:	$B \rightarrow A \rightarrow C$	$B \leftarrow A \leftarrow C$	$B \leftarrow A \rightarrow C$	total
Beobachtungsumfang:				
10	16	3	314	333 $\approx$ 33%
25	26	52	435	513 $\approx$ 51%
50	43	111	591	745 $\approx$ 75%
125	69	161	641	871 $\approx$ 87%
250	80	142	669	891 $\approx$ 89%
500	105	137	640	882 $\approx$ 88%
1000	95	112	676	883 $\approx$ 88%
10000	108	158	630	896 $\approx$ 90%

Die drei Netze sind äquivalent. Trotzdem fällt auf, daß das dritte Netz überdurchschnittlich oft präferiert wird. Das liegt aber ausschließlich an der Suchheuristik, wie die Ausgabe eines einzelnen Suchlaufs für einen Datensatz mit dem Beobachtungsumfang = 1000 zeigt:

SEARCH

```
[ 0] INIT rate for initial DAG          p(D|S) = exp(  -1218.9229)
+-----+
|      A ---> B                          -1205.7817* |
|      B ---> A                          -1205.7817* |
|      A ---> C                          -1065.2100* |
|      C ---> A                          -1065.2100* |
|      B ---> C                          -1213.9038 |
|      C ---> B                          -1213.9038 |
[ 1] | ADD      A ---> C          p(D|S) = exp(  -1013.2998) |
+-----+
|      A ---> B                          -1000.1587* |
|      B ---> A                          -1000.1587* |
|      C ---> B                          -1008.2808 |
[ 2] | ADD      A ---> B          p(D|S) = exp(   -996.8511) |
+-----+
[ 3] STOP no improvement                p(D|S) = exp(   -996.8511)
```

Hier sind neben den tatsächlichen Veränderungen auch Testbewertungen der möglichen Veränderungen dargestellt, sofern sie eine Verbesserung der Netzstruktur bewirken. Obwohl gleich gut bewertet, wird die Kante  $A \rightarrow C$  vor der umgekehrten Kante  $C \rightarrow A$  bewertet. Es wird aber bei gleichen Bewertungen die zuerst gefundene Lösung beibehalten. Dasselbe gilt im zweiten Schritt für die Kante  $A \rightarrow B$ , die wiederum vor der gleichwertigen Kante  $B \rightarrow A$  bewertet und somit als Lösung beibehalten wird. Nur relativ selten ergeben sich durch die konkreten Werte andere Lösungswege, in deren Verlauf dann auch das erste bzw. das zweite Netz gefunden werden.

**UserSampleSize=0, “ $\alpha + 1$ ”-Methode**

Netz: Beobachtungsumfang:	$B \rightarrow A \rightarrow C$	$B \leftarrow A \leftarrow C$	$B \leftarrow A \rightarrow C$	total
10	207	4	75	$286 \approx 29\%$
25	393	3	24	$420 \approx 42\%$
50	523	2	9	$534 \approx 53\%$
125	754	0	1	$755 \approx 76\%$
250	890	0	0	$890 \approx 89\%$
500	968	0	0	$968 \approx 97\%$
1000	987	0	0	$987 \approx 99\%$
10000	1000	0	0	$1000 \approx 100\%$

#### 10.4.3 Versuch 2: $B \rightarrow A \quad C$

Das Vorgegebene Netz ist  $B \rightarrow A \quad C$ . Zu diesem Netz gibt es ein äquivalentes Netz:  $B \leftarrow A \quad C$ .

**Vorwissen aus 20% der Daten**

Netz: Beobachtungsumfang:	$B \rightarrow A \quad C$	$B \leftarrow A \quad C$	total
10	11	500	$511 \approx 51\%$
25	58	531	$589 \approx 59\%$
50	46	413	$459 \approx 46\%$
125	83	610	$693 \approx 69\%$
250	131	623	$754 \approx 75\%$
500	126	648	$774 \approx 77\%$
1000	102	660	$762 \approx 76\%$
10000	143	645	$788 \approx 79\%$

Es gilt wieder, wie im Versuch 1, daß die Kante  $A \rightarrow B$  vor  $B \rightarrow A$  bewertet wird und daher bei Gleichwertigkeit präferiert wird, so daß das Vorgegebene Netz seltener gefunden wird als das äquivalente zweite Netz.

**UserSampleSize=0, “ $\alpha + 1$ ”-Methode**

Netz: Beobachtungsumfang:	$B \rightarrow A \quad C$	$B \leftarrow A \quad C$	total
10	256	66	$322 \approx 32\%$
25	395	18	$413 \approx 41\%$
50	488	5	$493 \approx 49\%$
125	614	0	$614 \approx 61\%$
250	779	0	$779 \approx 78\%$
500	834	0	$834 \approx 83\%$
1000	879	0	$879 \approx 88\%$
10000	969	0	$969 \approx 97\%$

#### 10.4.4 Versuch 3: $B \rightarrow A \leftarrow C$

Das Vorgegebene Netz ist  $B \rightarrow A \leftarrow C$ . Zu diesem Netz gibt es aber mehrere Netze, die den gleichen moralischen Graphen aufweisen. Obwohl sie nicht exakt den gleichen Informationsgehalt aufweisen, sind sie doch recht ähnlich. Diese Netze sind:  $A \rightarrow C \leftarrow B$  und  $A \rightarrow B \leftarrow C$ , sowie alle vollständig verbundenen DAG's (3-Kanten-DAG's). Bei der Bestimmung der Treffer müssen diese Netze also ebenfalls berücksichtigt werden. Es werden wieder jeweils 1000 verschiedene Datensätze erzeugt und zu jedem eine Netzstruktur gelernt. Da die Gefahr, daß mit für das Endergebnis ungünstigen Kanten begonnen wird relativ hoch ist, geben wir in einer parallelen Testreihe eine Kante ( $B \rightarrow A$ ) vor und beginnen mit diesem DAG den Lernalgorithmus. Diese Ergebnisse sind in Klammern angegeben.

**Vorwissen aus 20% der Daten**

Netz: Beob.umf.:	$B \rightarrow A \leftarrow C$	$A \rightarrow C \leftarrow B$	$A \rightarrow B \leftarrow C$	3 Kanten	total
10	12 (235)	54 (12)	25 (1)	64 (89)	155 $\approx$ 15% (337 $\approx$ 34%)
25	43 (236)	53 (16)	12 (0)	47 (43)	176 $\approx$ 18% (295 $\approx$ 30%)
50	58 (265)	46 (14)	6 (1)	69 (38)	179 $\approx$ 18% (318 $\approx$ 32%)
125	135 (471)	28 (2)	0 (0)	186 (45)	249 $\approx$ 25% (518 $\approx$ 52%)
250	153 (595)	1 (0)	0 (0)	333 (59)	487 $\approx$ 49% (654 $\approx$ 65%)
500	207 (734)	0 (0)	0 (0)	487 (74)	694 $\approx$ 69% (808 $\approx$ 81%)
1000	187 (854)	0 (0)	0 (0)	709 (67)	896 $\approx$ 90% (921 $\approx$ 92%)
10000	232 (891)	0 (0)	0 (0)	768 (109)	1000 $\approx$ 100% (1000 $\approx$ 100%)

Nimmt man die drei ähnlichen Netze zusammen, so steigert sich die Trefferquote auf erstaunliche 100%. Dennoch wird das vorgegebene Netz auch dann nur mit einem Anteil von 23% gefunden. Hier wurde mit Absicht ein Netz gewählt, daß der Suchheuristik Probleme bereitet. Zu oft werden die Kanten  $A \rightarrow C$  bzw.  $A \rightarrow B$ , den eigentlich richtigen, umgekehrten Kanten  $C \rightarrow A$  bzw.  $B \rightarrow A$  zunächst für gleichwertig gehalten und da in der Suchreihenfolge früher gefunden auch favorisiert. Später kann durch die begrenzte Blickweite des Suchalgorithmus' über eine Kantenänderung hinweg, dieser Fehler nur noch durch eine zusätzliche Kante ausgeglichen werden, wodurch die Abhängigkeiten abgebildet werden jedoch mit zu vielen Parametern (*overfitting*). Erst indem der Suchalgorithmus mit der Vorgabe einer Kante auf die richtige Fährte gesetzt wird, erreicht er zu 89% wirklich das richtige Ergebnis. In den restlichen 11% der erzeugten Daten ist dagegen der Einfluß zufälliger Schwankungen offenbar auch bei diesem relativ hohen Beobachtungsumfang von 10000 Fällen noch hoch genug, um einen vollständig verbundenen Graphen als richtiger erscheinen zu lassen.

**UserSampleSize=0, “ $\alpha + 1$ ”-Methode**

Netz: Beob.umf.:	$B \rightarrow A \leftarrow C$	$A \rightarrow C \leftarrow B$	$A \rightarrow B \leftarrow C$	3 Kanten	total
10	33	12	86	53	184 $\approx$ 18%
25	35	2	89	77	204 $\approx$ 20%
50	24	1	79	101	205 $\approx$ 21%
125	63	0	35	130	228 $\approx$ 23%
250	197	0	7	135	239 $\approx$ 24%
500	408	0	0	138	546 $\approx$ 55%
1000	658	0	0	124	682 $\approx$ 68%
10000	978	0	0	22	1000 $\approx$ 100%

Bei dieser Variante wird offenbar eine “Vorgabekante” weniger stark benötigt, da bei 10000 Beobachtungen fast ausschließlich das korrekte Netz gelernt wird.

```

SEARCH
[ 0] INIT rate for initial DAG          p(D|S) = exp( -18658.7266)
+-----+
[ 1] | ADD          A ---> C          p(D|S) = exp( -17643.3438) |
[ 2] | ADD          B ---> A          p(D|S) = exp( -17409.2188) |
[ 3] | REV          C </-> A          p(D|S) = exp( -17361.1094) |
+-----+
[ 4] STOP no improvement                p(D|S) = exp( -17361.1094)

```

Wie an der Ausgabe zu erkennen ist, wird zwar in den allermeisten Fällen zunächst die Kante zwischen A und C in der letztlich falschen Richtung eingefügt, jedoch danach umgedreht, was hier problemlos ist, da dem keine Hürde entgegensteht, wie das bei der 20%igen Verwendung der Daten als Vorwissen der Fall gewesen ist.

#### 10.4.5 Vergleich der beiden Testvarianten

Ein wenig überraschend ist, daß die “ $\alpha + 1$ ”-Methode offenbar sensitiver gegenüber **Kantenrichtungen** ist, auch bei im Informationsgehalt äquivalenten Netzen. Das hängt ausschließlich mit der konkret vorgegebe-

nen Wahrscheinlichkeitsverteilung zusammen. Es gilt hierbei, daß die Schwankungen von  $p(X_i|\mathbf{Pa}_i, S^h)$  gegenüber  $\frac{1}{r_i}$  ( $r_i$  = Anzahl der Ausprägungen von  $X_i$ ) jeweils größer sind als die umgekehrten Schwankungen der  $p(Pa|X_i, S^{h'}) \forall Pa \in \mathbf{Pa}_i$  ( $S^{h'} = \text{Netz mit Kante } X_i \rightarrow Pa \text{ statt } Pa \rightarrow X_i$ ) (vgl. Abschnitt 6.7). Werden andere Wahrscheinlichkeitsverteilungen gewählt, so kann sich dieser Effekt in sein Gegenteil verkehren, d.h. es wird dann jeweils die umgekehrte Kante der eigentlich vorgegebenen präferiert. So erhält man z.B. mit anderen Wahrscheinlichkeitsvorgaben für das Netz aus Versuch 1 bei 1000 Testläufen:

**UserSampleSize=0, “ $\alpha + 1$ ”-Methode (Wahrscheinlichkeits-Zufalls-Generator-Init: rnd)**

	Netz:	$B \rightarrow A \rightarrow C$	$B \leftarrow A \leftarrow C$	$B \leftarrow A \rightarrow C$	total
Beobachtungsumfang:					
(rnd = 11)	1000	143	766	0	809 $\approx$ 81%
(rnd = 42)	1000	47	856	0	893 $\approx$ 89%
(rnd = 88)	1000	0	861	123	984 $\approx$ 98%

Dieses scheinbare “Indiz” für die Kantenrichtung wird aber abgeschwächt, wenn nur 80% der Daten verwendet werden und demgegenüber “Vorwissen” steht, daß eine jeweils neue Kante als überflüssig annimmt, also Unabhängigkeit ausdrückt, ansonsten aber die vorgegebene, aus den Daten ableitbare Wahrscheinlichkeitsverteilung durchaus korrekt abbildet. Dann werden oft verschiedene Kantenrichtungen als gleichwertig betrachtet, da sich die Bewertungen im Rahmen der Darstellungsgenauigkeit nicht mehr unterscheiden (siehe auch Abschnitt 6.7).

Zum Vergleich die Variante mit “Vorwissen-Gewinnung” aus den Daten:

**Vorwissen aus 20% der Daten (Wahrscheinlichkeits-Zufalls-Generator-Init: rnd)**

	Netz:	$B \rightarrow A \rightarrow C$	$B \leftarrow A \leftarrow C$	$B \leftarrow A \rightarrow C$	total
Beobachtungsumfang:					
(rnd = 11)	1000	117	148	623	888 $\approx$ 89%
(rnd = 42)	1000	109	141	626	876 $\approx$ 88%
(rnd = 88)	1000	162	172	585	919 $\approx$ 92%

Dieser Unterschied in den beiden Lernmethoden ist also Ergebnis der Wahl der Wahrscheinlichkeitsverteilung und kann nicht als Qualitätsmerkmal der “ $\alpha + 1$ ”-Methode, auch bezogen auf anderweitig erzeugte bzw. echte Daten, gesehen werden.

Weniger überraschend ist dagegen, daß die Variante, in der 20% der Daten als “Vorwissen” benutzt werden, bei relativ geringen Beobachtungsumfängen, eine höhere Trefferquote erzielt, d.h. die zufälligen Schwankungen besser auszugleichen in der Lage ist, wohingegen bei hohen Beobachtungsumfängen diese Schwankungen keine so große Rolle mehr spielen (schwaches Gesetz der Großen Zahlen) und sich die Ergebnisse beider Lernverfahren annähern und schließlich die “ $\alpha + 1$ ”-Methode u.U. sogar eine höhere Trefferquote erzielt, da wie beschrieben, bei Verwendung von 20% der Daten als “Vorwissen” nach wie vor öfter mit falschen Kantenrichtungen begonnen wird, die später durch die verwendete Such-Heuristik (Greedy) nicht mehr zu berichtigen sind. Dieses gilt aber je nach Wahrscheinlichkeitsverteilung nur in bestimmten Fällen. In den beiden oben dargestellten Simulationsstudien erreicht in der ersten Testreihe die “Vorwissen-Aus-Daten-Methode” auch bei dem relativ hohen Beobachtungsumfang die bessere Trefferquote.

Man bedenke dabei, daß die Frage was ein “hoher” Beobachtungsumfang ist, von der Komplexität der Netzstruktur abhängt - scheint in den drei Beispielen ein Umfang von 1000 oder sogar 10000 Beobachtungen schon als fast übertrieben hoch, so waren die 1000 Beobachtungen im Kredit-Datensatz zuvor eher knapp bemessen.

## 11 Implementierung und Technisches

### 11.1 Vorraussetzungen und Grundlagen

#### 11.1.1 Implementierungssprache(n)

Die Implementierung des Netzstruktur-Lernalgorithmus’ war in der interaktiven vor allem im Bereich der Statistik verbreiteten Programmiersprache S-Plus vorzunehmen bzw. für die Verwendung in S-Plus auszulegen, wobei auch die in S-Plus zur Verfügung stehende C-Schnittstelle genutzt werden durfte.

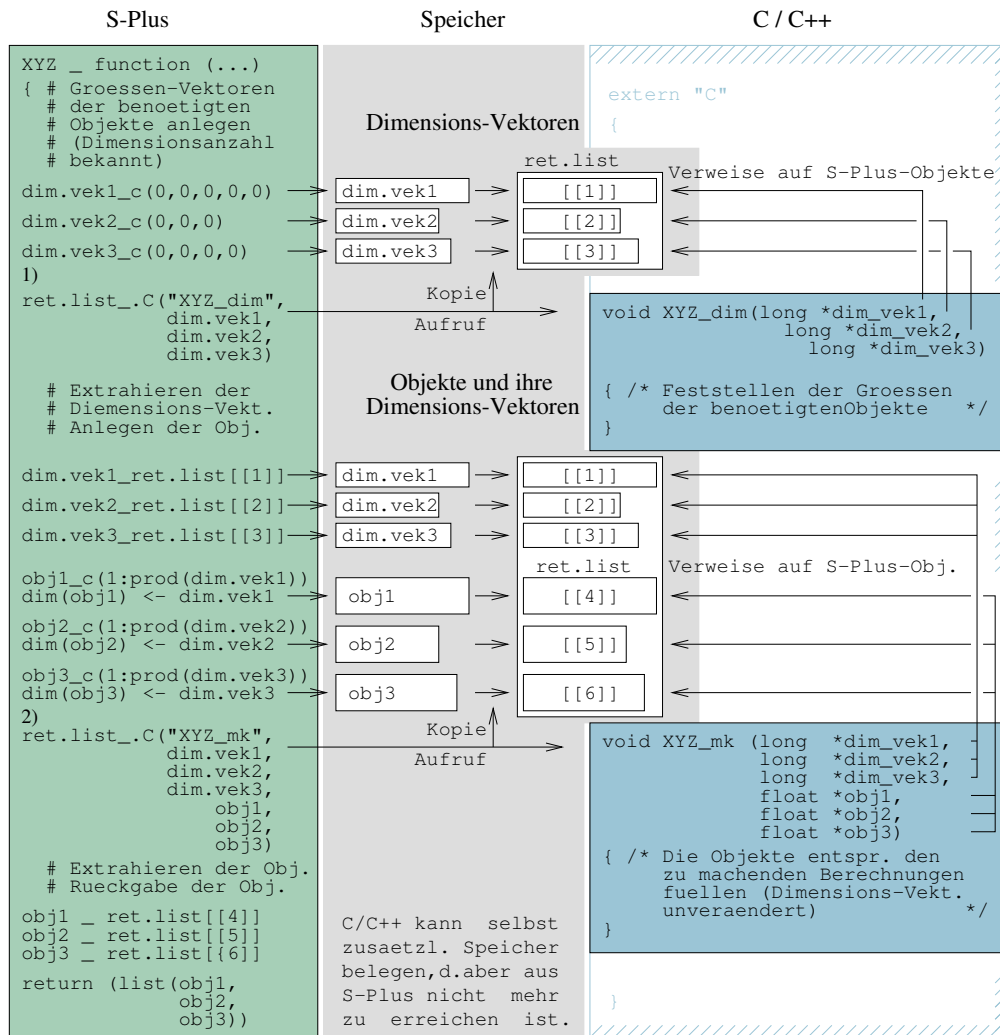


Abbildung 36: **Schematische Darstellung** der verwendeten Struktur bei der Berechnung von S-Plus-Objekten (meißt Matrizen und mehrdimensionale Tafeln) durch C/C++ Funktionen

S-Plus	Speicher	C / C++
1) zusätzliche Typfestlegung nötig	S-Plus erstellt grundsätzlich bei einem C-Funktions-Aufruf eine Kopie der Parameter;	das C/C++-Modul muß vor dem Aufruf der C-Funktionen in S-Plus geladen werden
2) zusätzliche Typfestlegung nötig	übergeben werden die Kopien!	

### 11.1.2 S-Plus

S-Plus besitzt neben der Möglichkeit interaktiv Funktionen aufzurufen, vor allem den Vorteil, daß auch komplexe Datentypen, wie Vektoren, Matrizen, mehrdimensionale Tafeln und (gemischte) Listen verschiedenster Typen nebst grundlegenden Operationen auf entsprechenden Objekten<sup>32</sup> implementiert sind. Ein weiterer Vorteil von S-Plus ist, daß eine Datenbank zur Speicherung solcher Objekte über die Länge einer Rechnersitzung hinaus zur Verfügung steht. Von entscheidendem Nachteil ist aber, daß wenn nicht hauptsächlich auf die erwähnten fest implementierten grundlegenden Rechenoperationen zurückgegriffen werden kann, S-Plus selbst für eine Interpretersprache<sup>33</sup> nicht im eigentlichen Sinne schnell zu nennen ist. Daher die Möglichkeit auf in C (oder auch Fortran) geschriebene Funktionen zuzugreifen und die bekanntermaßen hohe Geschwindigkeit dieser Compiler-Sprache<sup>34</sup> zu nutzen. Da die verwendeten Objekte letztenendes in S-Plus zu benutzen und zu “sehen” sind, bietet sich eine Nutzung der vorhandenen Datentypen (bzw. Klassen, im objektorientierten Sinn) zur Darstellung der benötigten Objekte an. Wir gehen im folgenden Abschnitt darauf ein, wie die einzelnen Objekte sinnvollerweise zu repräsentieren sind.

### 11.1.3 Festlegung der Speicherklassen in S-Plus

Bei der Übergabe komplexer, wie auch einfacher Datentypen an eine C-Funktion ist darauf zu achten, daß der Speichertyp der einzelnen Elemente in S-Plus zuvor festgelegt wird. Dabei entspricht:

S-Plus	C	in C++ auch
<code>storage.mode(ObjectXYZ)&lt;-"integer"</code>	<code>long*</code>	<code>long&amp; (f. Einzelwerte)</code>
<code>storage.mode(ObjectXYZ)&lt;-"single"</code>	<code>float*</code>	<code>float&amp; (f. Einzelwerte)</code>
<code>storage.mode(ObjectXYZ)&lt;-"character"</code>	<code>char**</code>	

oder

<code>as.integer(ObjectXYZ)</code>	<code>long*</code>	<code>long&amp; (f. Einzelwerte)</code>
<code>as.single(ObjectXYZ)</code>	<code>float*</code>	<code>float&amp; (f. Einzelwerte)</code>
<code>as.character(ObjectXYZ)</code>	<code>char**</code>	

Der Unterschied zwischen den beiden Varianten mit

```
storage.mode()<- "<type>"
```

und

```
as.<type>()
```

besteht darin, daß im ersteren Fall die Struktur des (komplexen) Datentyps in S-Plus erhalten bleibt, während die alternative Methode dazu führt, daß die Struktur verloren geht und die Daten in die Form eines Vektor (alle Elemente eindimensional aufgereiht) überführt werden. Diese Unterscheidung bezieht sich jedoch nur auf S-Plus selbst.

### 11.1.4 Aufspaltung von (gemischten) Objekten

Bei der Übergabe an C verliert ein Objekt in jedem Fall die Informationen über seine (komplexe) Struktur. Entscheidend ist bei der Festlegung der Speicherklasse, daß ein Objekt eines komplexen Datentyps “sortenrein” getrennt wird. So läßt sich z.B. eine Matrix mit Zeilen- und/oder Spaltennamen nicht in ein “integer”-Objekt umwandeln. Für die Übergabe an C-Funktionen müssen also komplexe Datentypen evtl. aufgespalten werden. Beispiel: Aufspaltung und Typfestlegung einer Ganzzahl-Matrix

```
a.matrix,
```

die Zeilen und Spaltennamen enthält:

---

<sup>32</sup>S-Plus ist im weiteren Sinne eine objektorientierte Sprache, so daß wir im Zusammenhang mit den erwähnten Datentypen auch von Objekten sprechen. Allerdings unterscheidet sich der S-Plus-Ansatz bezüglich Objektorientiertheit deutlich von anderen Sprachen und wird kaum (auch hier nicht) benutzt.

<sup>33</sup>der Programmcode wird erst während der Ausführung stückweise “interpretiert” (in Maschinencode umgesetzt)

<sup>34</sup>der gesamte Programmcode wird vorab in Maschinencode übersetzt (compiliert)



```

my.matrix _ as.integer(unlist(a.matrix,
                             recursive=TRUE,
                             use.names=FALSE))
dim.of.my.matrix _ as.integer(dim(a.matrix))
names.of.my.matrix _ as.character(names(a.matrix))
row.names.my.matrix _ as.character(row.names(a.matrix))

```

Neben Namensbezeichnungen müssen insbesondere auch Dimensionsvektoren, wie sie in S-Plus durch

```
dim(ObjectXYZ)
```

zu einem Objekt erhalten werden können, gesondert an C-Funktionen übergeben werden, damit sie in diesen bekannt sind, da mit der Aufspaltung und Übergabe der S-Plus-Objekte an C-Funktionen in jedem Fall die Strukturinformationen der Objekte verloren gehen.

### 11.1.5 Erstellung der Objekte in zwei Stufen

In jedem Falle ist es nötig, die Objekte, in denen C-Funktionen ihre Ergebnisse zurückgeben, zuvor in S-Plus (z.B. mit "0" gefüllt) anzulegen, d.h. insbesondere sie in ihrer Größe (Anzahl Elemente) festzulegen. Da oft selbst die Berechnung der Größen, also der Dimensionsvektoren, der verschiedenen benötigten Objekte zu (zeit-) aufwendig ist, um sie in S-Plus durchzuführen, erfolgt die Berechnung, wie in Abbildung 36 dargestellt evtl. in zwei Schritten:

1. Berechnung der Dimensionsvektoren der erforderlichen Objekte durch eine C-Funktion, wobei wir davon ausgehen, daß die Länge - d.h. die Anzahl der Dimensionen des benötigten Objekts - schon in S-Plus bekannt (erreichbar) ist. Andernfalls würde noch ein weiterer, vorangestellter Schritt zur Berechnung dieses Wertes nötig werden.
2. Erstellung der Objekte, entsprechend der im ersten Schritt berechneten Größe und Aufruf einer zweiten C-Funktion, um die Objekte entsprechend den durch die C-Funktion vorzunehmenden Berechnungen zu füllen.

### 11.1.6 Laden von C/C++-Modulen

Bevor eine/mehrere C-Funktion(en) aus S-Plus heraus mit

```
.C("<funktionsname>",<parameterliste>)
```

aufrufbar ist, muß das C-Modul, daß die entsprechende(n) (compilierten) Funktion(en), die üblicherweise in einem ".o"-File vorliegen, geladen werden. Das geschieht üblicherweise mit:

```
dyn.load("<pfad>+<file.o>")
```

### 11.1.7 Nutzung der Vorteile von C++ gegenüber C

Außer in C kann auch in C++ programmiert werden. Lediglich die Schnittstellenfunktionen, im in Abbildung 36 dargestellten Beispiel wären dies:

```
XYZ_dim() und XYZ_mk()
```

müssen nach außen hin C-Format haben. Dieses kann einfach dadurch erreicht werden, daß die Funktionen in einen

```
extern "C" { }
```

Block eingeschlossen werden. Die Verwendung von C++ hat den Vorteil, daß der objektorientierte Ansatz in S-Plus auch bei der Programmierung in C++ fortgeführt werden kann.

Beispiel einer C++-Klasse auch zur Übernahme von aus S-Plus übergebenen (eindimensionalen) Vektoren, (zweidimensionalen) Matritzen und mehrdimensionalen Tafeln (für Gleitkommazahlen), inclusive der Implementierung einer Elementfunktion für den Zugriff auf einzelne Zellen:

```

class complexObject
{
protected:

    // Attribute //
    float *field;           // Zeiger auf Datenfeld
    long n;                 // Anzahl der Dimensionen
    long *dim;              // Ausprägungen pro Dimension
    char **stateLabels[MAXDIM]; // Array der Ausprägungslabls

public:
    // Deklarationen: Konstruktoren
    complexObject (float*, long*, long, // z.B. Konstruktor zur Ueber-
                  char**, char**)      // nahme von S-Plus-Objekten
    /* ... */

    // Deklarationen: Elementfunktionen (Methoden)
    float *elem (long*)           // Zugriff auf einzelne Zellen
    /* ... */
};

float* complexObject::elem (long *Elem)
{
    long N, ElemNr=0, dimProduct=1;
    if (n>0) {
        for(N=0; N<n ; N++) {
            // Wenn Koordinaten ausserhalb der Objekt-Grenzen
            // werden die Koordinaten entsprechend angepasst
            if (Elem[N]<1) Elem[N]=r[N]-((0-Elem[N-1])%r[N]);
            else if (Elem[N]>r[N]) Elem[N-1]=((Elem[N-1]-1)%r[N])+1;
            ElemNr += dimProduct * (Elem[N]-1);
            dimProduct *= dim[N];
        }
        return (&field[ElemNr]); // Zeiger auf das gesuchte Element
    } else return(NULL);         // Objekt leer oder nicht in Ordnung
}

```

### 11.1.8 Objektübergabe / Objekterzeugung (S-Plus/C++)

Durch geeignete Konstruktoren können entweder die Datenfelder für

`*field`, `*dim` und `**stateLabels`

aus C++ heraus mit der aus der Standard-Bibliothek

`stdlib.h` ( `#include <stdlib.h>` )

stammenden Funktion

`calloc()`

angelegt werden, um C++-interne Objekte zu erzeugen, oder die Zeiger können auf die Übergabewerte aus einem S-Plus-Aufruf referenziert werden, so daß ohne Umkopieren direkt die S-Plus-Übergabedaten übernommen werden und somit auch bei der Rückgabe keine Umkopierarbeiten anfallen. Dies gilt in diesem Beispiel jedoch nicht für das Attribut:

`n`,

da die Anzahl der Dimensionen ohnehin vorher in S-Plus zur Erzeugung des Dimensionsvektors bekannt sein muß und daher nie (verändert) zurückgegeben werden kann.

Ein Übernahme-Konstruktor für S-Plus-Objekte in C++-Objekte kann dann beispielsweise so aussehen:

```

complexObject::complexObject (float *S_Object,
                              long  *S_Dim,
                              long   S_N,
                              char **S_ColNames,
                              char **S_RowNames)
{

```

```

long i, j;

if (S_N>0) {
  dim=S_Dim;
  n=S_N;
  field = S_Object;

  for(j=0;j<N;j++) {
    stateLabels[j]=NULL;
    if (j==0 && S_RowNames!=NULL) {
      stateLabels[0]=S_RowNames;
    } else if (j==1 && S_ColNames!=NULL) {
      stateLabels[1]=S_ColNames;
    }
  }
} else n=0; // evtl. Fehler bei Uebergabe
}

```

Damit kann die komplette Übernahme z.B. einer Matrix von S-Plus nach C/C++ wie folgt realisiert werden:

**S-Plus**

```

Uebergabe _ function(a.matrix)
{
  dyn.load("Uebernahme.o")

  my.matrix _ as.integer(unlist(a.matrix,
                                recursive=TRUE,
                                use.names=FALSE))
  dim.of.my.matrix _ as.integer(dim(a.matrix))
  names.of.my.matrix _ as.character(names(a.matrix))
  row.names.my.matrix _ as.character(row.names(a.matrix))

  my.new.matrix.list _ .C("Matrix_Uebernahme",
                          my.matrix,
                          dim.of.my.matrix,
                          as.integer(2),
                          names.of.my.matrix,
                          row.names.my.matrix)

  my.new.matrix _ my.new.matrix.list[[1]]
  dim(my.new.matrix) <- my.new.matrix.list[[2]]
  row.names(my.new.matrix) <- my.new.matrix.list[[4]]
  names(my.new.matrix) <- my.new.matrix.list[[5]]

  return(my.new.matrix)
}

```

**C++**

```

// File: Uebernahme.cc
extern "C"
{
  void Matrix_Uebernahme(float *S_Object,
                          long *S_Dim,
                          long &S_N,
                          char **S_ColNames,
                          char **S_RowNames)
  {
    // Uebernahme der Matrixteile und Zusammenbau zu
    // dem C++-Objekt C_Matrix:
    complexObject C_Matrix(S_Object,S_Dim,S_N,
                           S_ColNames,S_RowNames);

    /* Veraenderung des Matrixinhalts mithilfe
       weiterer (Element-) Funktionen */
  }
}

```

### 11.1.9 Der "Zwei-Stufen-Fall"

Zuletzt haben wir konkret den einfacheren Fall betrachtet, in dem die Größe des zu übergebenden Objekts, der Matrix

```
a.matrix
```

bereits bekannt ist. Dieser Fall ist für den Kern des Lernalgorithmus' ausreichend, da die Größe des DAG-Objekts schon vor dem Aufruf der Lern-Routine bekannt sein muß, zumindest wenn wir den DAG so präsentieren, wie dies im folgenden Abschnitt vorgestellt wird. Die schwierigere Variante findet aber u.U. bei Randproblemen, wie der Erstellung mittels C-Funktionen der Vorwissen-Wahrscheinlichkeits-Tafeln oder auch bei der Erstellung eines kompletten Bayes'schen-Netz-Objekts zur weiteren Verwendung (z.B. Inferenzberechnung) ihre Anwendung. Der Parameter

```
float *S_Object
```

kann dann an eine erste C-Funktion nicht übergeben werden. Der Dimensionsvektor

```
long *S_Dim
```

wird leer (mit "0" gefüllt) übergeben und durch die erste C-Funktion mit den errechneten Werten für die benötigte Objektgröße belegt. Danach kann mit

```
ObjectXYZ _ c(1:prod(dim.of.my.object))
ObjectXYZ <- dim.of.my.object
storage.mode(ObjectXYZ) <- "single"
```

das Objekt selbst in S-Plus angelegt werden und zur Bestimmung der Werte - wie zuvor beschrieben - an eine weitere C-Funktion übergeben werden. Dabei steht

```
dim.of.my.object
```

für den Rückgabewert, des in C berechneten Dimensionsvektors und

```
prod(dim.of.my.object)
```

für eine Funktion, die  $\prod \text{dim.of.my.object}$  - also die Anzahl der Zellen des Objekts - ergibt.

### 11.1.10 Listen Komplexer Objekte

Für die Erstellung mehrerer, in ihrer Anzahl variabler mehrdimensionaler Objekte, z.B. im Falle der Wahrscheinlichkeitstafelerstellung, können u.U. die Dimensionsvektoren selbst zu einem komplexeren Typ zusammengefügt werden, z.B. einer  $n \times m$ -Matrix für  $n$  Tafeln mit je  $m$  Dimensionen:

```
dim.matrix _ matrix(data=0,ncol=n,nrow=m)
storage.mode(dim.matrix) <- "integer"
```

Auch die Tafeln selbst können, um sie an C übergeben zu können, als ein Objekt erzeugt werden, z.B. als ein Mega-Vektor, der später in die einzelnen Tafeln aufzusplitten ist (sowohl in C/C++ zur Berechnung, als auch nach der Rückgabe in S-Plus, um sie in anschaulicher(er) Art zu speichern):

```
all.objects _ c(elements.in.all.objects),
```

wobei

```
elements.in.all.objects
```

für das Ergebnis einer Funktion steht, die  $\sum_n \prod_m \text{dim.matrix}$ , also die Anzahl der Zellen in allen Tafeln zusammen - ergibt.

Die Überführung eines solchen Mega-Vektors in eine Liste mehrdimensionaler Tafeln kann in S-Plus folgendermaßen geschehen:

```

create.table.list _ function(all.objects,dim.matrix)
{
  Table.List _ list(1:dim(dim.matrix)[1])
  Pos _ 1
  for (i in 1:dim(dim.matrix)[2]) {
    Prod _ 1
    for (j in 1:dim(dim.matrix)[1])
      Prod _ Prod * dim.matrix[j,i]
    Table _ all.objects[Pos:(Pos+Prod-1)]
    dim(Table) <- dim.matrix[,i]
    Table.List[[i]] _ Table
    Pos _ Pos + Prod
  }
  return(Table.List)
}

```

in C++ wird zunächst eine Klasse zur Repräsentation einer `complexObject`-Liste benötigt:

```

class complexObjList
{
public:

  // Attribute //
  long nTables;           // Anzahl Tables
  class comlexObject *Tables; // Tafel-Liste

  /* Konstruktoren, Element-Funktionen (Methoden) */
};

```

Durch die Bibliotheks-Funktion

```
calloc() (<stdlib.h>)
```

kann in einem Konstruktor zunächst Speicher für  $n$  `complexObjects` reserviert werden. Die Zeiger-Attribute der einzelnen Objekte

```
ListObjXYZ[i].dim; ListObjXYZ[i].field;
```

können dann auf die entsprechenden Segmente, der aus S-Plus übergebenen Objekte

```
dim.matrix bzw. all.objects
```

referenziert werden.

## 11.2 Repräsentation der Objekte

Für das weitere Vorgehen bei der Umsetzung des Lern-Algorithmus' ist es von Bedeutung, wie die benötigten Objekte repräsentiert werden.

### 11.2.1 Die Wahrscheinlichkeitstafeln

Im Fall etwa der Wahrscheinlichkeitstafeln (Vorwissen) ist die Frage der Umsetzung in S-Plus-Objekte in natürlicher Weise durch die zur Verfügung stehenden Datentypen geregelt. Es bietet sich eine Liste mehrdimensionaler Tafeln an. Da wir eine feste Zuordnung von Merkmalen (Knoten) zu Dimensionen vornehmen, sind alle (zu  $n$ -Knoten gegebenen) Tafeln  $n$ -dimensional. Für den Dimensionsvektor einer Tafel  $i$  gilt daher

```
dim(table.list[[i]])[k] == 1
```

wenn Merkmal  $k$  **nicht** in der Tafel enthalten ist. Wie mit solchen Listen mehrdimensionaler Tafeln zu verfahren ist, hatten wir bereits im vorigen Abschnitt erläutert.

### 11.2.2 Der Datensatz

Auch die Repräsentation eines Datensatzes ist durch S-Plus schon weitgehend vorgegeben. Es handelt sich schlicht um eine Matrix, deren Spalten den Merkmalen und deren Zeilen je einem beobachteten Fall entsprechen.

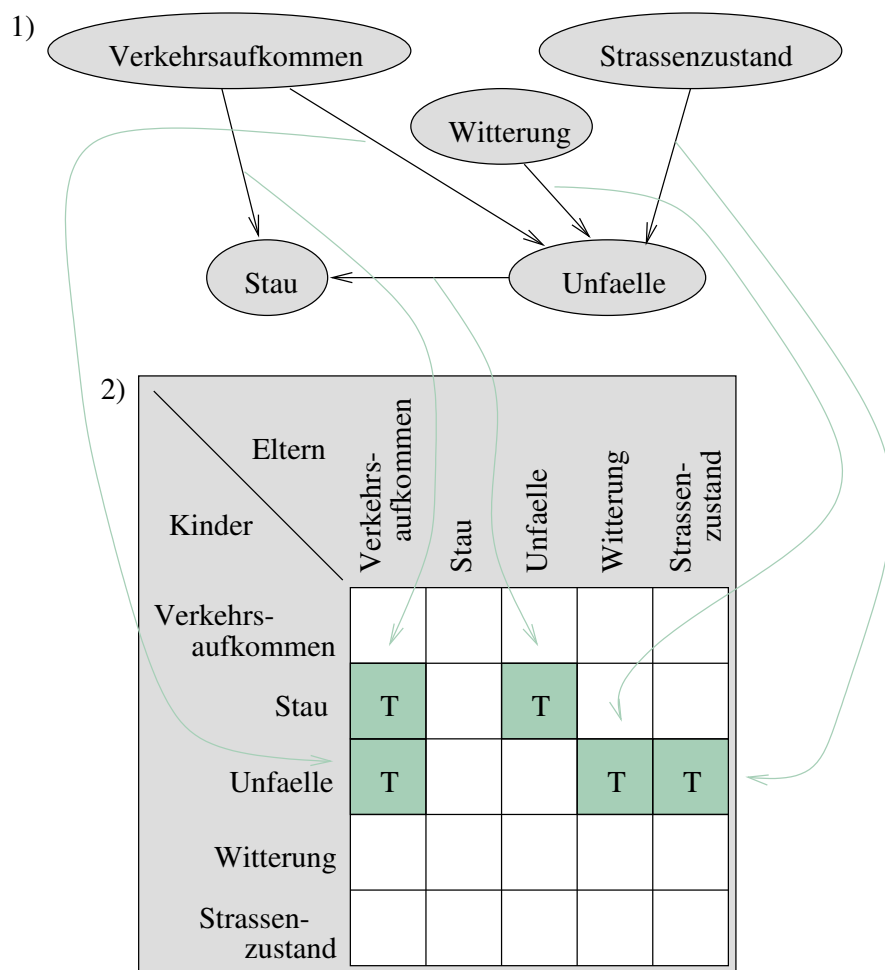


Abbildung 37: Repräsentation eines DAG's durch eine Matrix

### 11.2.3 Der DAG

Für die Umsetzung des DAG's in ein S-Plus-Objekt gäbe es mehrere Möglichkeiten, unter denen gewählt werden könnte. Möglich wäre etwa eine  $n$ -elementige Liste von (unterschiedlich langen) Vektoren, die jeweils die Eltern (oder auch die Kinder) eines Knotens angeben. Diese Variante hat den Vorteil, daß der DAG mit einem Minimum an Speicher dargestellt wird. Dennoch favorisieren wir eine Alternative. Der DAG wird durch eine  $n \times n$ -Matrix wiedergegeben, wobei der Wert in der  $i$ -ten Zeile und der  $l$ -ten Spalte angibt, ob  $X_l$  Elternknoten von  $X_i$  ist.

```
DAG[i,l]==TRUE : l ist Elternknoten von i
DAG[i,j]==FALSE: j ist nicht Elternknoten von i
```

Diese Variante enthält Redundanz, indem sie sowohl gesetzte, als auch nicht gesetzte Kanten darstellt und zu einer gesetzten Kante auch zusätzlich die Information darstellt, daß sie in der anderen Richtung nicht gesetzt ist, was aber in einem DAG per Definition gegeben ist. Sie ist aber für eine gleichförmige Berechnungsstruktur und auch für die Nutzung der C-Schnittstelle leichter zu handhaben. Da DAG's nur eine relativ begrenzte Knotenanzahl beinhalten, kann das Problem der nicht optimalen Speichernutzung und das ebenfalls resultierende Problem der nicht optimalen Berechnungsgeschwindigkeit, beim Durchsuchen der DAG-Matrix nach Kanten, vernachlässigt werden.

## 11.3 (Daten-) Struktur des Lernalgorithmus'

Wir haben nun theoretisch die Gleichungen und Formeln aufgestellt, um eine Netzstruktur und von dieser ausgehend weitere abgeleitete Strukturen zu bewerten und um daraus einen Lernalgorithmus zu entwickeln (Greedy-Search). Außerdem haben wir die grundlegenden Fragen nach der zu verwendenden Implementierungsmethode geklärt und wollen nun beides zusammenführen.

Der Lernalgorithmus geht von folgenden Ausgangsgrößen aus:

1. Einem Datensatz  $D$ , der in der Form einer Matrix gegeben wird.
2. Einer initialen DAG-Matrix, welche das Vorwissen über die Netzstruktur repräsentiert, die aber auch "leer" (kantenlos) sein darf.
3. Einer Matrix in derselben Größe wie die DAG-Matrix, die verbotene, bzw. erlaubte Kantenänderungen in der DAG-Matrix darstellt. Man beachte, daß jede Zelle der DAG-Matrix einer Kante in einer bestimmten Richtung entspricht (die Diagonale bleibt ausgenommen, da hier "TRUE's" Kanten entsprechen, die der Definition des DAG's widersprechen). Wir legen fest, daß der Wert TRUE in dieser zweiten Matrix das Verbot bedeutet, diese Zelle in der DAG-Matrix zu verändern. Diese Veränderungs-Verbots-Matrix nennen wir kurz NOCHANGE-Matrix.
4. U.U. den bedingten Wahrscheinlichkeitstabellen, entsprechend der Struktur des initialen DAG's, die das Vorwissen bezüglich der Wahrscheinlichkeitsverteilungen darstellen. Dieses Vorwissen kann auch in der Form von "Unwissen" im Bayesschen Sinne gegeben werden, also in der Form von mit Gleichverteilung gefüllten Tabellen, wobei die Wahl der *UserSampleSize* (siehe Abschnitt 8.2.1 - 8.2.3) problematisch ist. Alternativ kann dieses Wissen auch gänzlich weggelassen werden, wobei dann das Vorwissen entsprechend Abschnitt 8.3 wie die normalen Kontingenztafeln ebenfalls aus den Daten gewonnen wird.
5. Einem Gewichtsverhältnis des Datensatz'  $D$  zum Vorwissen (**ALPHAWEIGHT**), aus dem bei gegebenem Vorwissen die *UserSampleSize* errechnet wird und bei fehlendem Vorwissen der Anteil der Daten, der für das Vorwissen (bzw. welcher weiterhin für die normalen Kontingenztafeln) Verwendung findet.
6. Einer Angabe, ob mit der " $\alpha + 1$ -Methode" gerechnet werden soll (**ALPHAPLUS1**)
7. Einer Begrenzung der Anzahl der maximal vorzunehmenden Verbesserungsschritte (**MAXITER**)
8. Einer Angabe ob zusätzlich eine Bestimmung der Netz-a-priori-Wahrscheinlichkeit  $p(S^h)$  gemäß Gleichung (40) in die Gesamtbewertung einbezogen werden soll (**USENETPRIOR**)

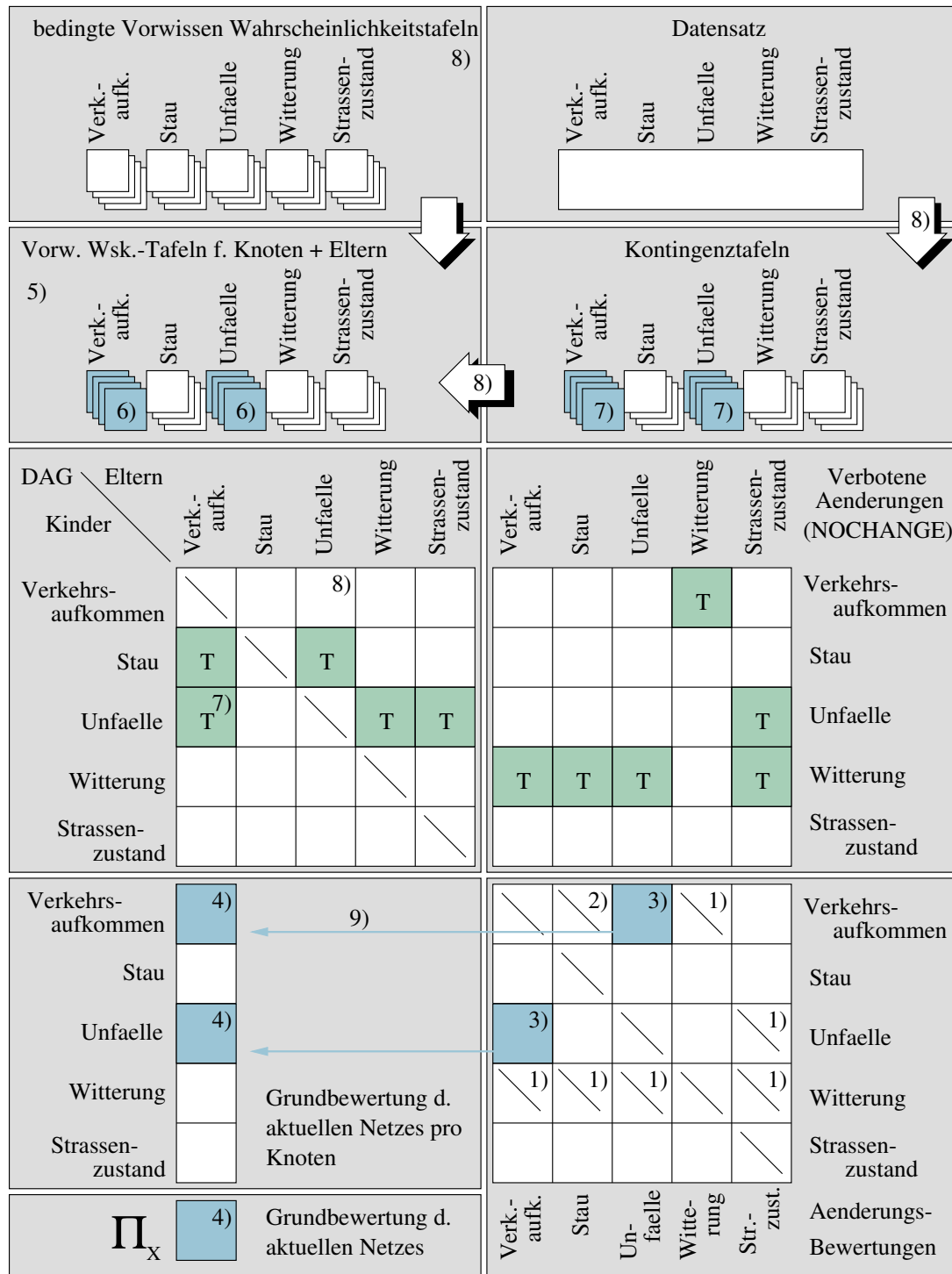


Abbildung 38: Schematische Darstellung der bei einer Netzberechnung verwendeten Objekte

- |                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                                                                                                          |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>1) vom Benutzer ausgeschlossen (NOCHANGE)</p> <p>3) beste Änderung als Bsp.: Kante umdrehen (beide Zeilen im nächsten Schritt neu berechnen)</p> <p>5) Neuberechnung von zwei Wahrscheinlichkeitstafeln</p> <p>8) Berechnung differiert ohne Vorw.-Angabe (Berechnung ähnlich den Kontingenztafeln)</p> | <p>2) Zirkel; evtl. später zu berechnen</p> <p>4) neue Grundbewertung für 2 Knoten i.d.R. := 3) Ausnahme siehe 8)</p> <p>6) neuzuberechnende Wsk.-Tafeln</p> <p>7) neuzuberechnende Kontingenztafeln</p> <p>9) Ohne Vorw.-Angabe evtl. Neuberechnung statt Übernahme</p> |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|



### 11.3.1 Mit Vorwissen

Bei gegebenem Vorwissen können nun aus diesem, entsprechend den in Kapitel 7 vorgestellten Berechnungsschritten, die gemeinsamen Wahrscheinlichkeitsverteilungen jedes Knotens gemeinsam mit seinen Eltern errechnet werden. Man beachte, daß wir für weitere Berechnungen zu abgeleiteten Netzstrukturen diese gemeinsamen Wahrscheinlichkeitstafeln noch benötigen, während wir die  $\alpha$ -“Vorwissen-Kontingenztafeln”, die durch die Multiplikation mit der *UserSampleSize* entstehen, erst beim Einsetzen der Werte in die Berechnungsformel (Gleichung (26)) benutzen. Es ist daher sinnvoll, die gemeinsamen Wahrscheinlichkeitstafeln während eines Laufs des Lernalgorithmus’ zu speichern und die  $\alpha$ -Werte erst bei der Einsetzung in Gleichung (26) zu erzeugen.

$$\alpha_{ijk} = [1+] UserSampleSize \cdot p(X_i = k, \mathbf{Pa}_i = j | S^h) \quad (41)$$

$$\alpha_{ij} = [r_i+] UserSampleSize \cdot \sum_k p(X_i = k, \mathbf{Pa}_i = j | S^h) \quad (42)$$

|  $r_i$  ist die Anzahl der Ausprägungen von  $X_i$  |

(Die Terme in eckigen Klammern sind nur bei der “ $\alpha + 1$ -Methode” zu verwenden.)

Während diese Tafeln jeweils an die konkrete Netzstruktur  $S_{neu}^h$  angepaßt werden, belassen wir die bedingten Wahrscheinlichkeitstafeln unverändert entsprechend  $S_{orig}^h$ , um aus ihnen evtl. das Original-Vorwissen ableiten zu können (siehe Gleichung ???). Daneben benötigen wir zum Vergleich, ob die Original-Tafeln zur Berechnung herangezogen werden müssen<sup>35</sup>, die original DAG-Matrix zu  $S_{orig}^h$ , müssen also zur Darstellung des aktuellen Graphen  $S_{akt}^h$ , zu einem Zeitpunkt des Lernalgorithmus’s anfangs einmalig eine Kopie anlegen, die dann fortlaufend verändert wird. Zur Repräsentation des Test-DAG’s  $S_{neu}^h$  wird dagegen nicht unbedingt eine weitere Kopie benötigt, da die Test-Änderungen in der Matrix zu  $S_{akt}^h$  vorgenommen, bewertet und wieder zurückgenommen werden können, bis sich eine Änderung als deutlichste Verbesserung herausstellt und in  $S_{akt}^h$  übernommen wird. Im Gegensatz dazu sollten zur Darstellung der entsprechend  $S_{neu}^h$  abgeleiteten  $\alpha$ -Vorwissen-Tafeln andere Objekte als die Wahrscheinlichkeitstafeln zu  $S_{akt}^h$  verwendet werden, da eine Rücknahme der Test-Änderungen hier sonst problematisch wäre.

### 11.3.2 Gruppierte Daten

Zu dem vorgegebenen Netz  $S_{orig}^h$ , wie auch zu jedem später zu testenden Netz  $S_{neu}^h$ , werden die Kontingenztafeln ( $N_{ijk}$ ) aus der Datensatz-Matrix zu  $D$  abgeleitet. Da dieser Schritt evtl. sehr oft zu wiederholen ist, wird i.d.R. durch die Gruppierung der Daten<sup>36</sup> eine Geschwindigkeitssteigerung bei der Berechnung erzielt.

### 11.3.3 Ohne Vorwissen

Bei nicht gegebenem Vorwissen werden entsprechend Abschnitt 8.3 auch die  $\alpha$ -Tafeln, wie die Kontingenztafeln aus den Daten erzeugt, wobei im Falle der Bewertung eines Netzes mit einer hinzugekommenen Kante die  $\alpha$ -Tafeln zunächst entsprechend  $S_{akt}^h$  zu erzeugen sind und erst danach, wenn sich für diese Änderung entschieden wurde die  $\alpha$ ’s für eine Grundbewertung noch einmal entsprechend dem neuen  $S_{akt}^h$  erzeugt werden. Wenn ohne Vorwissen gearbeitet wird, werden die  $\alpha$ - und Kontingenztafeln noch mit je einem Gewichtsparameter multipliziert, um eine Verdoppelung des wahren Beobachtungsumfangs und damit eine Überschätzung der mit  $D$  gegebenen Information zu vermeiden:

$$\alpha_{ijk} = [1+] ALPHAWEIGHT \alpha_{ijk}^* \quad (43)$$

$$\alpha_{ij} = [r_i+] ALPHAWEIGHT \alpha_{ij}^* \quad (44)$$

$$N_{ijk} = (1 - ALPHAWEIGHT) N_{ijk}^* \quad (45)$$

$$N_{ij} = (1 - ALPHAWEIGHT) N_{ij}^* \quad (46)$$

<sup>35</sup>Das war der Fall, wenn eine Kante im Laufe mehrerer Verbesserungsschritte wiedereingefügt wird, die im Original-Ausgangs-Graph  $S_{orig}^h$  enthalten war, aber in einem vorhergehenden Schritt gelöscht worden war.

<sup>36</sup>Zusammenfassung der Fälle mit für alle Merkmale identischen Ausprägungen

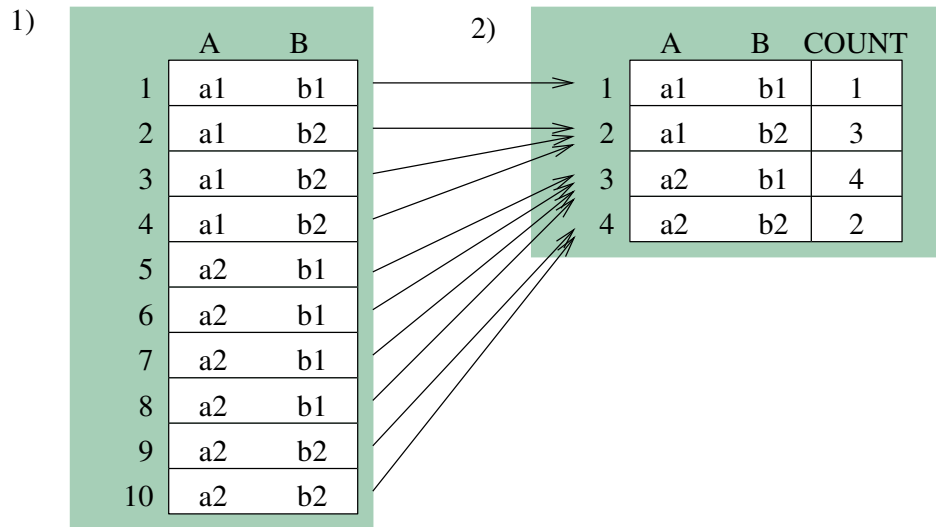


Abbildung 39:

- 1) ungruppierte Daten
- 2) gruppierte Daten

<p>hier seien mit <math>\alpha_{ijk}^*, \alpha_{ij}^*, N_{ijk}^*, N_{ij}^*</math></p> <p style="text-align: center;"><math>r_i</math></p>	<p>die normal erzeugten Kontingenztafeln aus <math>D</math> gemeint, in denen der Wert jeder Zelle mit der Anzahl der korrespondierenden Beobachtungen übereinstimmt d. Anzahl d. Auspräg. v. <math>X_i</math></p>
-------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

(Die Terme in eckigen Klammern sind nur bei der “ $\alpha + 1$ -Methode” zu verwenden.)

### 11.3.4 Der Grundbewertungs-Vektor

Vor dem ersten Netzverbesserungslauf steht einmalig eine Grundbewertung aller Knoten (einzeln). Diese Bewertungen werden in einem Vektor abgelegt, das Produkt aller enthaltenen Werte ergibt die Gesamtbewertung für  $p(D|S_{orig}^h)$  ( $\cdot p(S_{orig}^h)$  bei Einbeziehung der Netz-a-priori )

### 11.3.5 Die Änderungsbewertungs-Matrix

Alle von einem aktuellen Netz  $S_{akt}^h$  (anfänglich  $S_{orig}^h$ ) abgeleiteten Bewertungen aller Knoten (einzeln), bei einzelnen Kantenänderungen (d.h. mit Auswirkung nur auf die Elternmenge einzelner Knoten), können in einer Matrix, in ihrer Größe äquivalent zur DAG-Matrix abgelegt werden, indem die zur geänderten DAG-Zelle korrespondierende Änderungs-Matrix-Zelle mit der neuen Bewertung des Knotens (der entsprechenden Zeile) belegt wird. In einem ersten initialen Suchlauf werden alle Zellen dieser Matrix berechnet, die nicht zu einer Netzstruktur  $S_{neu}^h$  gehören, die Zirkel enthält<sup>37</sup>. Die Zelle (im Falle von Kantenumkehrungen zwei Zellen) dieser Matrix, mit der die größte Verbesserung bezüglich  $p(D|S^h)$  erzielt wird, kann in den Grundbewertungsvektor pro Knoten übernommen werden (Ohne Vorwissen muß bei hinzugefügten Kanten neu berechnet werden). Die Zeilen der betroffenen Zelle(n) in der Änderungs-Bewertungs-Matrix müssen gelöscht werden, so daß sie im nächsten Lauf neu berechnet werden, alle anderen bereits errechneten Werte behalten ihre Gültigkeit. Die im letzten Verbesserungslauf übernommenen Zellen der Änderungs-Bewertungs-Matrix müssen im darauffolgenden Lauf nicht sofort neu berechnet werden, da dadurch das vorhergehende Netz bewertet würde, das aber bereits als schlechter als das aktuelle erkannt wurde (Im Falle der Umkehrung einer Kante war auch das Fehlen dieser Kante zuvor schon bewertet und zumindest als weniger gut als eben die Umkehrung der betreffenden Kante befunden worden). Die Berechnung dieser Zellen kann also für einen Suchlauf ausgesetzt werden.

<sup>37</sup> eine solche Zelle wird erst berechnet, wenn durch vorhergehende Kantenänderungen kein Zirkel mehr entsteht

## 11.4 Struktur des Lernalgorithmus' ( Überblick )

### 11.5 Rechnen mit logarithmierten Werten

Unsere bisherige Berechnungsformel aus Gleichung (26) birgt, für nicht sehr kleine Werte sowohl aus den  $N_{ijk}$ -, als auch aus den  $\alpha_{ijk}$ -Tabellen, das Problem, daß die Werte der  $\Gamma$ -Funktion die Größe jedes üblichen Darstellungsbereichs in Computern sprengen und die Endergebnisse für  $p(D|S^h)$  wiederum zu klein sind um bei einer rechnergestützten Umsetzung noch von "0" unterschieden zu werden. Ein u.a. auch in der Statistik üblicher Ausweg besteht darin, mit logarithmierten Werten zu rechnen. Wir verwenden daher stattdessen die Formel:

$$\begin{aligned}\log(p(D|S^h)) &= \prod_i \prod_j \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \prod_k \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})} \\ &= \sum_i \sum_j \log \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} + \sum_k \log \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})} \\ &= \sum_i \sum_j \log(\Gamma(\alpha_{ij})) - \log(\Gamma(\alpha_{ij} + N_{ij})) \\ &\quad + \sum_k \log(\Gamma(\alpha_{ijk} + N_{ijk})) - \log(\Gamma(\alpha_{ijk}))\end{aligned}\tag{47}$$

Entsprechend berechnen wir bei Einbeziehung der Netz-a-priori statt Gleichung (40):

$$\begin{aligned}\log(p(D|S^h) \cdot p(S^h)) &= \log(p(D|S^h)) + \log(p(S^h)) \\ &= \log(p(D|S^h)) + \log(e^{-\frac{B}{2} \log N}) \\ &= \log(p(D|S^h)) - \frac{B}{2} \log N\end{aligned}\tag{48}$$

Damit würde aber nicht der gewünschte Erfolg zu erzielen sein, nie den Darstellungsbereich von Zahlen im Rechner zu über- / unterschreiten, wenn nicht in den meisten Programmiersprachen statt, oder zusätzlich zur  $\Gamma$ - auch direkt die  $\log \Gamma$ -Funktion implementiert wäre. So ist auch in der Standardbibliothek

```
math.h ( #include <math.h> )
```

in C die Funktion

```
(double) lgamma (double)
```

implementiert.

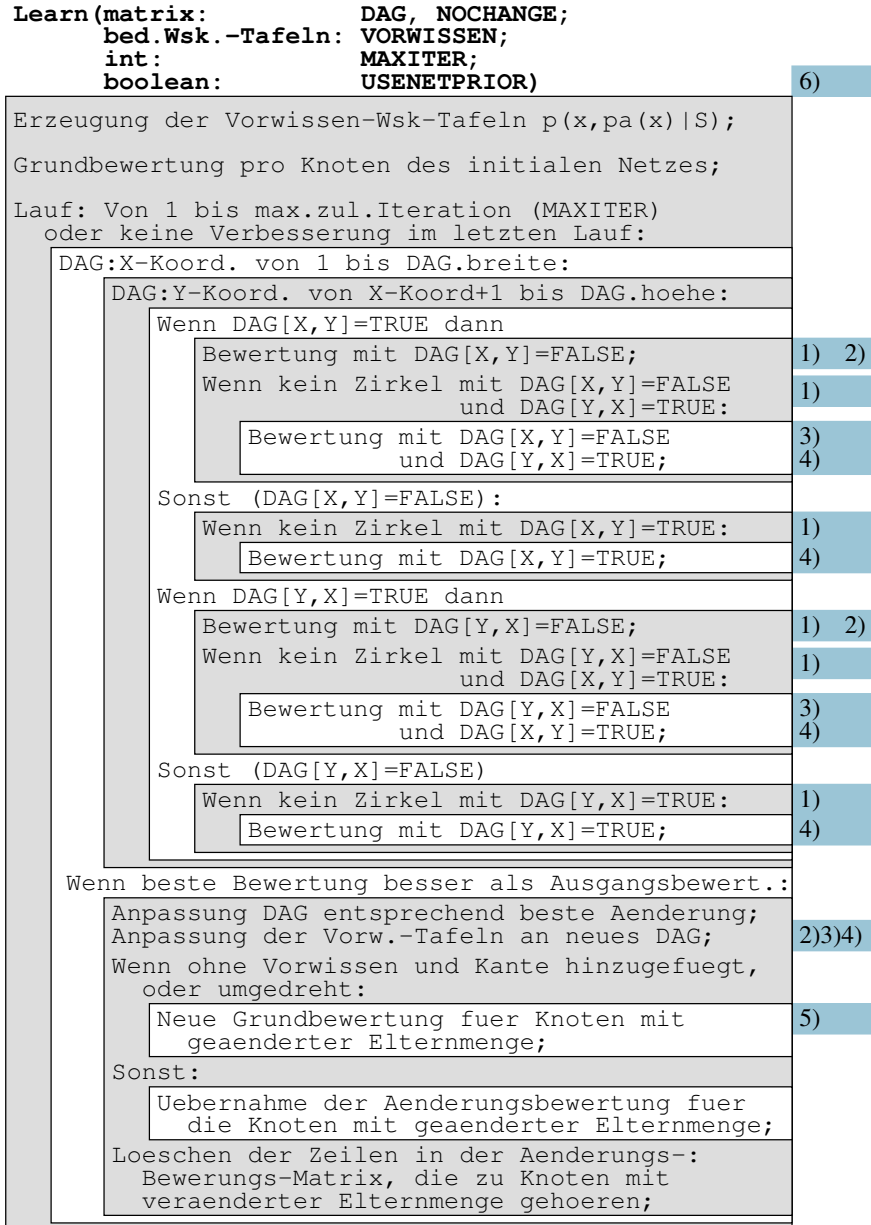


Abbildung 40: Struktur des **Lernalgorithmus**'

- 1) wenn NOCHANGE[X,Y]=FALSE bzw. NOCHANGE[Y,X]=FALSE  
(d.h. Änderung erlaubt)
- 2) testweise Anpassung der Vorwissen-Wsk.-Tafeln entspr.  
Gleichung (35) testweise Neuerzeugung der Kontingenztafeln
- 3) testweise Anpassung der Vorwissen-Wsk.-Tafeln entspr.  
Gleichung (36)/(37) testweise Neuerzeugung der Kontingenztafeln
- 4) testweise Anpassung der Vorwissen-Wsk.-Tafeln entspr.  
Gleichung (36)/(37) testweise Neuerzeugung der Kontingenztafeln
- 5) da das "Vorwissen" bei der Testbewertung zum Netz ohne die  
neue Kante erzeugt wurde
- 6) Berechnung 2,3,4) entspr. Gleichung (40) um die Bestimmung der  
Netz-a-priori  $p(S^h)$  erweitern, wenn USENETPRIOR=TRUE
- 2), 3) u. 4) Berechnung nur, wenn Wert noch nicht ermittelt oder wieder  
gelöscht wurde; sonst kann der Wert aus der Änderungs-  
Bewertungs-Matrix direkt als Ergebnis eingesetzt werden, um  
die Bewertung des betreffenden Knotens in  $S_{neu}^h$  zu erhalten

## 12 Anhang

### 12.1 Mehrdimensionale Tafeln ( formale Definitionen )

#### 12.1.1 Größe Mehrdimensionaler Tafeln

Die Größe mehrdimensionaler Tafeln kann durch einen Dimensions-Ausprägungs-Vektor kurz *dim*-Vektor beschrieben werden:

Tafel	<i>dim</i> -Vektor	Anzahl Zellen
(eindimensionaler) Vektor mit 4 Zellen	$dim = (4)$	4
(zweidimensionale) $3 \times 8$ -Matrix	$dim = (3, 8)$	$3 \cdot 8 = 24$
(dreidimensionale) $3 \times 5 \times 7$ -Tafel	$dim = (3, 5, 7)$	$3 \cdot 5 \cdot 7 = 105$
$4 \times 9 \times 2 \times 14 \times 1 \times 6$ -Tafel	$dim = (4, 9, 2, 14, 1, 6)$	$4 \cdot 9 \cdot 2 \cdot 14 \dots = 6048$
allgemein gilt für eine $n$ -dimensionale Tafel	$dim = (r_1, r_2, r_3, \dots, r_n)$	$\prod_{i \in \{1 \dots n\}} r_i$

Die Länge des Dimensions-Ausprägungs-Vektors entspricht der Anzahl der Dimensionen der Tafel. Der  $i$ -te Wert des Vektors ( $r_i$ ) entspricht der Anzahl der Ausprägungen (Kategorien), in die die  $i$ -te Dimension eingeteilt ist. Für jedes  $r_i$  ( $i \in \{1 \dots n\}$ ) gilt  $r_i > 0$ , da wenn mind. ein  $r_i = 0$  auch  $\prod dim = 0$  wäre. Für eine (eigentlich) nicht in einer Tafel enthaltene Dimension  $i$  setzen wir  $r_i = 1$ . Bei allen unseren Berechnungen gilt für die *dim*-Vektoren  $dim^X$  und  $dim^Y$  zweier Tafeln  $X$  und  $Y$  mit gleicher Dimensionsanzahl  $n = n^X = n^Y$ :

$$\begin{aligned}
 & r_i^X = r_i^Y \\
 \text{oder} \quad & r_i^X = 1 \text{ (nicht dargestellte Dimension)} \\
 \text{oder} \quad & r_i^Y = 1 \text{ (nicht dargestellte Dimension)} \\
 \text{f.a.} \quad & i \in \{1 \dots n\}
 \end{aligned}$$

#### 12.1.2 Vektoren “in den ( mehrdimensionalen ) Raum legen”

Alle verwendeten Tafeln werden bei uns die gleiche Dimensionsanzahl haben, bis auf eine Ausnahme, der Multiplikation einer mehrdimensionalen Tafel  $T$  mit einem “Informations”-Vektor  $I$ , der zu einem bestimmten Merkmal in der Tafel  $T$  gegeben wird und entsprechend “in den Raum gelegt” werden muß. Das geschieht auf folgende Weise:

Der *dim*-Vektor des “Informations”-Vektors  $I$   $dim^I = (r^I)$  wird derart erweitert, daß der neue *dim*-Vektor  $dim^{I*}$  die gleiche Länge ( $n^T$ ) wie der *dim*-Vektor der Tafel  $dim^T$  hat.

Im Einzelnen werden die Werte des Vektors folgendermaßen gesetzt:

$$\begin{aligned}
 dim^{I*} &= (1, 1, 1, \dots, r_l^{I*} := r^I, 1, 1, 1, \dots, r_{n^T}^{I*} := 1), \\
 \left| \begin{array}{l} \text{wobei } l \text{ die Nummer der fraglichen Dimension} \\ \text{in } T \text{ ist zu der } I \text{ gehört} \end{array} \right|
 \end{aligned}$$

Auch hier gilt bei uns immer  $r^I = r_l^T$ . Damit wird der Vektor zu einer  $n^T$ -dimensionalen Tafel “hochtransformiert”, ohne daß sich etwa die Anzahl der Zellen ändern würde.

#### 12.1.3 Größe resultierender Tafeln

Bei jeder Rechenoperation, die auf eine Tafelmeng  $\mathbf{T}$  mit zwei oder mehr  $n$ -dimensionalen Tafeln angewendet wird, entsteht eine neue Tafel mit ebenfalls  $n$  Dimensionen. Für den *dim*-Vektor der resultierenden Tafel  $R$  gilt:

$$r_i^R = \max_{\mathbf{T}} r_i^{\mathbf{T}}$$

Durch z.B. Multiplikation von einer  $3 \times 1 \times 1$ -Tafel mit einer  $3 \times 4 \times 1$ -Tafel und einer  $1 \times 4 \times 9$ -Tafel entstünde also eine  $3 \times 4 \times 9$ -Tafel.

#### 12.1.4 Zugriff auf einzelne Zellen in mehrdimensionalen Tafeln

Um auf einzelne Zellen einer mehrdimensionalen Tafel zugreifen zu können definieren wir einen Zugriffoperator  $elem()$ , der einen Koordinaten-Vektor der betreffenden Zelle als Parameter erhält:  $(e_1, e_2, e_3, \dots, e_n)$ . Dabei sind Koordinaten, die außerhalb der Tafel liegen möglich ( $e_i > r_i^T$ ) in einem solchen Fall gelte<sup>38</sup>:

$$T.elem(e_1, e_2, \dots, e_n) := T.elem(e_1 \bmod r_1^T, e_2 \bmod r_2^T, \dots, e_n \bmod r_n^T)$$

Vereinfacht ausgedrückt werden bei einem kontinuierlichen Durchlaufen der Tafel, die betreffenden Koordinaten, die zu einer Überschreitung der Tafelgrenzen führen, wieder auf 1 zurückgesetzt, um von vorne zu beginnen.

#### 12.1.5 Rechenoperationen auf mehrdimensionalen Tafeln

Jede Rechenoperation, wie Multiplikation, Division, Addition oder Subtraktion von mehrdimensionalen Tafeln  $T^1, T^2, \dots$  sei dann (hier am Beispiel der Addition) beschrieben durch:

$$\begin{aligned} R.elem(e_1, e_2, e_3, \dots, e_n) &= T^1.elem(e_1, e_2, e_3, \dots, e_n) \\ &+ T^2.elem(e_1, e_2, e_3, \dots, e_n) \\ &+ \dots \\ &\text{f.a. } e_i \in \{1 \dots r_i^R\} \text{ und } i \in \{1 \dots n\} \end{aligned}$$

#### 12.1.6 Rechnen mit mehrdimensionalen Tafeln und Einzelwerten

Wird eine Rechenoperation wie Multiplikation, Division, Addition oder Subtraktion auf eine mehrdimensionale Tafel  $T$  und einen einzelnen Wert  $z$  (z.B.  $z = 5$ ) angewendet so entsteht eine Tafel  $R$ , für die gilt:  $dim^R := dim^T$  und (am Beispiel der Addition):

$$\begin{aligned} R.elem(e_1, e_2, e_3, \dots, e_n) &= T^1.elem(e_1, e_2, e_3, \dots, e_n) + Z \\ &\forall e_i \in \{1 \dots r_i^R\} \text{ und } i \in \{1 \dots n\} \end{aligned}$$

Dieses Verfahren ist äquivalent zu einer Überführung von  $z$  in eine einzellige  $n^T$ -dimensionale Tafel  $Z$ , für deren  $dim$ -Vektor gilt:  $r_i^Z = 1 \forall i \in \{1 \dots n^Z\}$  ( $n^Z := n^T$ ).

#### 12.1.7 Marginale Tafeln / Marginalisierung

Sollen eine oder mehrere Dimensionen in einer Tafel nicht mehr auftauchen, weil die enthaltene Information vernachlässigt werden soll, so kann über diese Dimensionen marginalisiert werden. In der resultierenden Tafel  $M$  "fallen" solche Dimensionen "zusammen", d.h. es gilt für den  $dim$ -Vektor  $dim^M$ :  $r_m^M = 1$  für alle zu marginalisierenden Dimensionen  $m$ .

Für die Werte der einzelnen Zellen der marginalen Tafel  $M$  gilt (mit  $m \in \{m_1, m_2\}$ ):

$$\begin{aligned} M.elem(e_1, \dots, 1, \dots, 1, \dots, e_n) &= \sum_{e_{m_1} \in \{1 \dots r_{m_1}^T\}} \sum_{e_{m_2} \in \{1 \dots r_{m_2}^T\}} T.elem(e_1, \dots, e_{m_1}, \dots, e_{m_2}, \dots, e_n) \\ &\equiv M = \sum_{m_1, m_2} T \end{aligned}$$

#### 12.1.8 Gewinnung einzelner Zellen einer Kontingenztafel

Eine Zelle einer Kontingenztafel  $K$  zu dem in Abschnitt 1.6 dargestellten Datensatz, kann wie folgt gewonnen werden:

$$\begin{aligned} &K.elem(21 - 30 \text{ J}, 181 - 190 \text{ cm}, \text{Informatiker}, 0 - 10 \text{ Kamele}, \text{blond}) \\ &:= \text{Anzahl Fälle in } D \text{ mit :} \\ &\quad \text{Alter} = 21 - 30 \text{ Jahre,} \\ &\quad \text{Größe} = 181 - 190 \text{ cm,} \end{aligned}$$

---

<sup>38</sup>  $x \bmod y$  : Rest der Ganzzahldivision  $\frac{x}{y}$

*Beruf = Informatiker,*  
*Vermögen = 0 – 10 Kamele,*  
*Haarfarbe = blond*

Für alle weiteren Zellen in  $K$  ist entsprechend zu verfahren. Dabei werden die Ausprägungen jedes Merkmals normalerweise kodiert, so daß die Zuordnung über Nummern von 1 bis  $r_i^K$  für jedes Merkmal  $i \in \{1 \dots n\}$  erfolgt. Auch hier gilt für den  $dim$ -Vektor nichtdargestellter Merkmale  $m$  einer Kontingenztafel  $K'$ :  $r_m^{K'} = 1$ . Dabei wird eine entsprechende marginale Tafel  $K' = \sum_m K$  der Kontingenztafel aller Merkmale gebildet - allerdings aus Effizienzgründen i.d.R. direkt, ohne über die "Gesamttafel"  $K$  zu gehen.

### 12.1.9 Darstellungsformen verschiedener Verteilungen

## 12.2 Detaillierte Berechnung zur Wahl der *UserSampleSize*

Berechnung der Ergebnisse zur Abbildung 16:

$$\begin{aligned}
 p(D|S^1) &= \frac{\Gamma(\alpha_{i=A,j=()})}{\Gamma(\alpha_{i=A,j=()} + N_{i=A,j=()})} \cdot \frac{\Gamma(\alpha_{i=A,j=(),k=a_1} + N_{i=A,j=(),k=a_1})}{\Gamma(\alpha_{i=A,j=(),k=a_1})} \\
 &\cdot \frac{\Gamma(\alpha_{i=A,j=(),k=a_2} + N_{i=A,j=(),k=a_2})}{\Gamma(\alpha_{i=A,j=(),k=a_2})} \cdot \frac{\Gamma(\alpha_{i=B,j=()})}{\Gamma(\alpha_{i=B,j=()} + N_{i=B,j=()})} \\
 &\cdot \frac{\Gamma(\alpha_{i=B,j=(),k=b_1} + N_{i=B,j=(),k=b_1})}{\Gamma(\alpha_{i=B,j=(),k=b_1})} \cdot \frac{\Gamma(\alpha_{i=B,j=(),k=b_2} + N_{i=B,j=(),k=b_2})}{\Gamma(\alpha_{i=B,j=(),k=b_2})} \\
 \\
 p(D|S^2) &= \frac{\Gamma(\alpha_{i=A,j=()})}{\Gamma(\alpha_{i=A,j=()} + N_{i=A,j=()})} \cdot \frac{\Gamma(\alpha_{i=A,j=(),k=a_1} + N_{i=A,j=(),k=a_1})}{\Gamma(\alpha_{i=A,j=(),k=a_1})} \\
 &\cdot \frac{\Gamma(\alpha_{i=A,j=(),k=a_2} + N_{i=A,j=(),k=a_2})}{\Gamma(\alpha_{i=A,j=(),k=a_2})} \cdot \frac{\Gamma(\alpha_{i=B,j=(a_1)})}{\Gamma(\alpha_{i=B,j=(a_1)} + N_{i=B,j=(a_1)})} \\
 &\cdot \frac{\Gamma(\alpha_{i=B,j=(a_2)})}{\Gamma(\alpha_{i=B,j=(a_2)} + N_{i=B,j=(a_2)})} \cdot \frac{\Gamma(\alpha_{i=B,j=(a_1),k=b_1} + N_{i=B,j=(a_1),k=b_1})}{\Gamma(\alpha_{i=B,j=(a_1),k=b_1})} \\
 &\cdot \frac{\Gamma(\alpha_{i=B,j=(a_2),k=b_1} + N_{i=B,j=(a_2),k=b_1})}{\Gamma(\alpha_{i=B,j=(a_2),k=b_1})} \cdot \frac{\Gamma(\alpha_{i=B,j=(a_1),k=b_2} + N_{i=B,j=(a_1),k=b_2})}{\Gamma(\alpha_{i=B,j=(a_1),k=b_2})} \\
 &\cdot \frac{\Gamma(\alpha_{i=B,j=(a_2),k=b_2} + N_{i=B,j=(a_2),k=b_2})}{\Gamma(\alpha_{i=B,j=(a_2),k=b_2})}
 \end{aligned}$$

*UserSampleSize* = 100

$$\begin{aligned}
 p(D|S^1) &= \frac{\Gamma(100)}{\Gamma(110)} \cdot \frac{\Gamma(54)}{\Gamma(50)} \cdot \frac{\Gamma(56)}{\Gamma(50)} \cdot \frac{\Gamma(100)}{\Gamma(110)} \cdot \frac{\Gamma(55)}{\Gamma(50)} \cdot \frac{\Gamma(55)}{\Gamma(50)} \\
 &= \frac{9.33262 \cdot 10^{155}}{1.44386 \cdot 10^{176}} \cdot \frac{4.27488 \cdot 10^{69}}{6.08282 \cdot 10^{62}} \cdot \frac{1.26964 \cdot 10^{73}}{6.08282 \cdot 10^{62}} \cdot \frac{9.33262 \cdot 10^{155}}{1.44386 \cdot 10^{176}} \cdot \frac{1.30844 \cdot 10^{71}}{6.08282 \cdot 10^{62}} \cdot \frac{1.30844 \cdot 10^{71}}{6.08282 \cdot 10^{62}} \\
 &= 8.82631 \cdot 10^{-7} \\
 \\
 p(D|S^2) &= \frac{\Gamma(100)}{\Gamma(110)} \cdot \frac{\Gamma(54)}{\Gamma(50)} \cdot \frac{\Gamma(56)}{\Gamma(50)} \cdot \frac{\Gamma(50)}{\Gamma(54)} \cdot \frac{\Gamma(50)}{\Gamma(56)} \cdot \frac{\Gamma(26)}{\Gamma(25)} \cdot \frac{\Gamma(29)}{\Gamma(25)} \cdot \frac{\Gamma(28)}{\Gamma(25)} \cdot \frac{\Gamma(27)}{\Gamma(25)} \\
 &= \frac{9.33262 \cdot 10^{155}}{1.44386 \cdot 10^{176}} \cdot \frac{4.27488 \cdot 10^{69}}{6.08282 \cdot 10^{62}} \cdot \frac{1.26964 \cdot 10^{73}}{6.08282 \cdot 10^{62}} \cdot \frac{6.08282 \cdot 10^{62}}{4.27488 \cdot 10^{69}} \cdot \frac{6.08282 \cdot 10^{62}}{1.26964 \cdot 10^{73}} \\
 &\cdot \frac{1.55112 \cdot 10^{25}}{6.20448 \cdot 10^{23}} \cdot \frac{3.04888 \cdot 10^{29}}{6.20448 \cdot 10^{23}} \cdot \frac{1.08889 \cdot 10^{28}}{6.20448 \cdot 10^{23}} \cdot \frac{4.03291 \cdot 10^{26}}{6.20448 \cdot 10^{23}} \\
 &= 9.05825 \cdot 10^{-7}
 \end{aligned}$$

*UserSampleSize* = 10

$$\begin{aligned}
 p(D|S^1) &= \frac{\Gamma(10)}{\Gamma(20)} \cdot \frac{\Gamma(9)}{\Gamma(5)} \cdot \frac{\Gamma(11)}{\Gamma(5)} \cdot \frac{\Gamma(10)}{\Gamma(20)} \cdot \frac{\Gamma(10)}{\Gamma(5)} \cdot \frac{\Gamma(10)}{\Gamma(5)} \\
 &= \frac{632880}{1.21645 \cdot 10^{17}} \cdot \frac{40320}{24} \cdot \frac{6328800}{24} \cdot \frac{632880}{1.21645 \cdot 10^{17}} \cdot \frac{632880}{24} \cdot \frac{632880}{24} \\
 &= 5.16775 \cdot 10^{-7} \\
 \\
 p(D|S^2) &= \frac{\Gamma(10)}{\Gamma(20)} \cdot \frac{\Gamma(9)}{\Gamma(5)} \cdot \frac{\Gamma(11)}{\Gamma(5)} \cdot \frac{\Gamma(5)}{\Gamma(9)} \cdot \frac{\Gamma(5)}{\Gamma(11)} \cdot \frac{\Gamma(3.5)}{\Gamma(2.5)} \cdot \frac{\Gamma(6.5)}{\Gamma(2.5)} \cdot \frac{\Gamma(5.5)}{\Gamma(2.5)} \cdot \frac{\Gamma(4.5)}{\Gamma(2.5)} \\
 &= \frac{632880}{1.21645 \cdot 10^{17}} \cdot \frac{40320}{24} \cdot \frac{6328800}{24} \cdot \frac{24}{40320} \cdot \frac{24}{6328800} \cdot \frac{3.32335}{1.32934} \cdot \frac{287.885}{1.32934} \cdot \frac{52.3428}{1.32934} \cdot \frac{11.6317}{1.32934}
 \end{aligned}$$

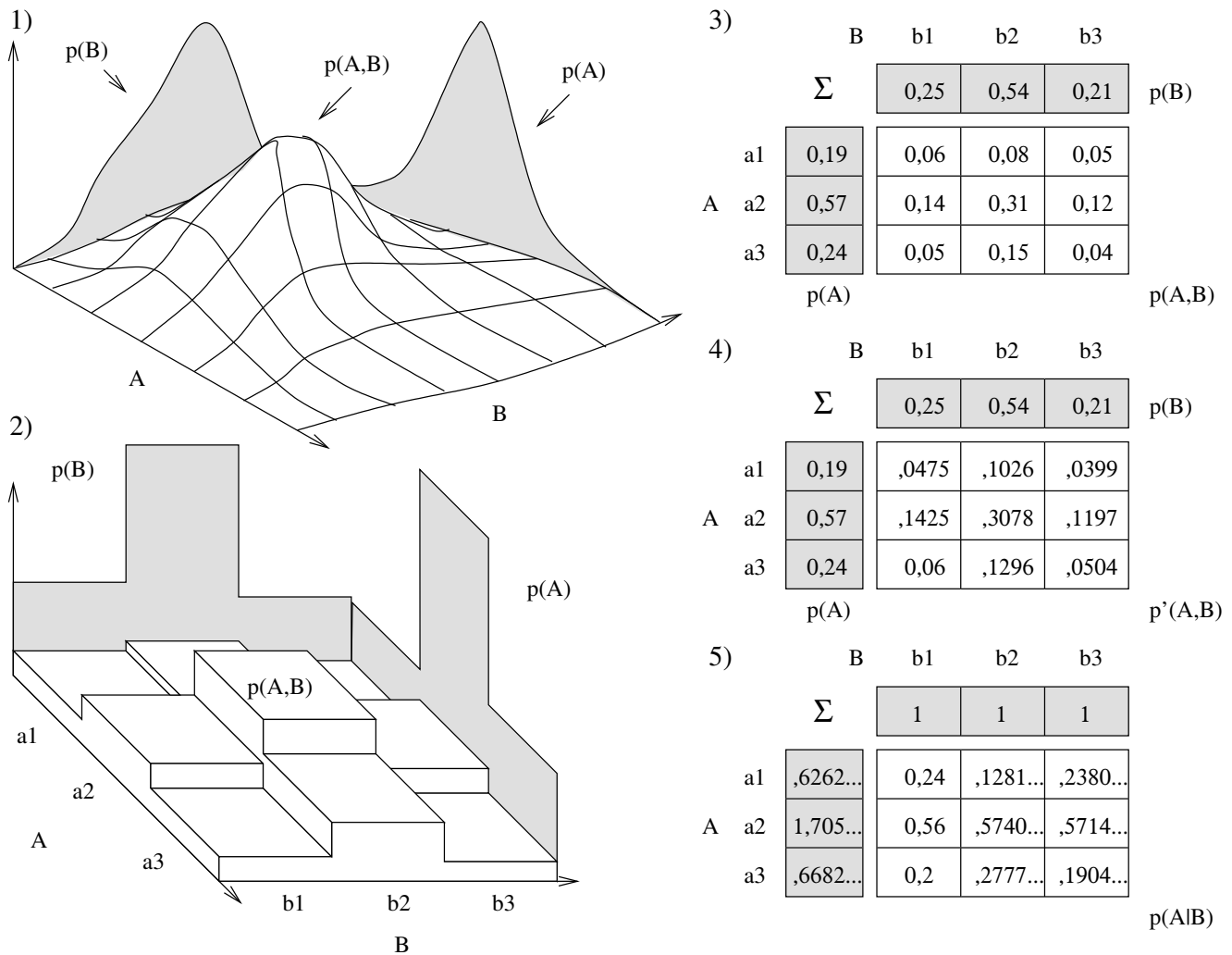


Abbildung 41: Verteilungen, mehrdimensionale Tafeln und Rechenbeispiele

- 1) gemeinsame Verteilung zweier stetiger Merkmale  $A$  u.  $B$  :  
für eine stetige Verteilung gilt :  
sowie die marginalen Verteilungen d. einzelnen Merkmale :
$$p(A, B)$$

$$\int p(A, B) dA, B = 1$$

$$p(A) = \int p(A, B) dB$$

$$p(B) = \int p(A, B) dA$$
- 2) Verteilung zweier diskreter (diskretisierter) Merkmale  $A$  und  $B$  mit je drei Ausprägungen :  
für eine diskrete Verteilung gilt:  
sowie den marginalen Verteilungen:
$$p(A, B), \text{ mit}$$

$$A \in \{a1, a2, a3\} \text{ u. } B \in \{b1, b2, b3\}$$

$$\sum_{A, B} p(A, B) = 1$$

$$p(A) = \sum_{\{b1, b2, b3\}} p(A, B)$$

$$p(B) = \sum_{\{a1, a2, a3\}} p(A, B)$$
- 3) Darstellung von 2) in Tafel-Form
- 4) Beispiel für Matritzenmultiplikation  
Durch die Multiplikation einer  $3 \times 1$ - und einer  $1 \times 3$ -Tafel entsteht eine  $3 \times 3$ -Tafel  
es gilt hier im allgemeinen :  
wenn  $p(A, B) = p(A)p(B)$  sind  $A$  und  $B$  unabhängig
$$p'(A, B) := p(A)p(B)$$

$$p(A, B) \neq p(A)p(B)$$
- 5) Beispiel: bedingte Wahrscheinlichkeitsverteilung für  
diese Tafel kann aus  $p(A, B)$  gewonnen werden durch :  
umgekehrt gilt, wenn  $p(A|B)$  u.  $p(B)$  gegeben :  
die marginalen Randwerte sind nicht informativ
$$p(A|B) \text{ "A gegeben B"}$$

$$p(A|B) = p(A, B)/p(B)$$

$$p(A, B) = p(A|B)p(B)$$



$$= 5.56443 \cdot 10^{-7}$$

$$UserSampleSize = 1$$

$$\begin{aligned} p(D|S^1) &= \frac{\Gamma(1)}{\Gamma(11)} \cdot \frac{\Gamma(4.5)}{\Gamma(0.5)} \cdot \frac{\Gamma(6.5)}{\Gamma(0.5)} \cdot \frac{\Gamma(1)}{\Gamma(11)} \cdot \frac{\Gamma(5.5)}{\Gamma(0.5)} \cdot \frac{\Gamma(5.5)}{\Gamma(0.5)} \\ &= \frac{1}{3628800} \cdot \frac{11.6317}{1.77245} \cdot \frac{187.885}{1.77245} \cdot \frac{1}{3628800} \cdot \frac{52.3428}{1.77245} \cdot \frac{52.3428}{1.77245} \\ &= 7.05913 \cdot 10^{-8} \end{aligned}$$

$$\begin{aligned} p(D|S^2) &= \frac{\Gamma(1)}{\Gamma(11)} \cdot \frac{\Gamma(4.5)}{\Gamma(0.5)} \cdot \frac{\Gamma(6.5)}{\Gamma(0.5)} \cdot \frac{\Gamma(0.5)}{\Gamma(4.5)} \cdot \frac{\Gamma(0.5)}{\Gamma(6.5)} \cdot \frac{\Gamma(1.25)}{\Gamma(0.25)} \cdot \frac{\Gamma(4.25)}{\Gamma(0.25)} \cdot \frac{\Gamma(3.25)}{\Gamma(0.25)} \cdot \frac{\Gamma(2.25)}{\Gamma(0.25)} \\ &= \frac{1}{3628800} \cdot \frac{11.6317}{1.77245} \cdot \frac{187.885}{1.77245} \cdot \frac{1.77245}{11.6317} \cdot \frac{1.77245}{287.885} \cdot \frac{0.906402}{3.62561} \cdot \frac{8.28509}{3.62561} \cdot \frac{2.54926}{3.62561} \cdot \frac{1.133}{3.62561} \\ &= 3.4592 \cdot 10^{-8} \end{aligned}$$

$$UserSampleSize = 0.1$$

$$\begin{aligned} p(D|S^1) &= \frac{\Gamma(0.1)}{\Gamma(10.1)} \cdot \frac{\Gamma(4.05)}{\Gamma(0.05)} \cdot \frac{\Gamma(6.05)}{\Gamma(0.05)} \cdot \frac{\Gamma(0.1)}{\Gamma(10.1)} \cdot \frac{\Gamma(5.05)}{\Gamma(0.05)} \cdot \frac{\Gamma(5.05)}{\Gamma(0.05)} \\ &= \frac{9.51351}{454761} \cdot \frac{11.6317}{19.4701} \cdot \frac{287.885}{19.4701} \cdot \frac{9.51351}{454761} \cdot \frac{25.8843}{19.4701} \cdot \frac{25.8843}{19.4701} \\ &= 1.70461 \cdot 10^{-9} \end{aligned}$$

$$\begin{aligned} p(D|S^2) &= \frac{\Gamma(0.1)}{\Gamma(10.1)} \cdot \frac{\Gamma(4.05)}{\Gamma(0.05)} \cdot \frac{\Gamma(6.05)}{\Gamma(0.05)} \cdot \frac{\Gamma(0.05)}{\Gamma(4.05)} \cdot \frac{\Gamma(0.05)}{\Gamma(6.05)} \cdot \frac{\Gamma(1.025)}{\Gamma(0.025)} \cdot \frac{\Gamma(4.025)}{\Gamma(0.025)} \cdot \frac{\Gamma(3.025)}{\Gamma(0.025)} \cdot \frac{\Gamma(2.025)}{\Gamma(0.025)} \\ &= \frac{9.51351}{454761} \cdot \frac{11.6317}{19.4701} \cdot \frac{287.885}{19.4701} \cdot \frac{19.4701}{6.39118} \cdot \frac{19.4701}{130.716} \cdot \frac{0.986174}{39.447} \cdot \frac{6.19196}{39.447} \cdot \frac{2.04693}{39.447} \cdot \frac{1.01083}{39.447} \\ &= 1.0916 \cdot 10^{-10} \end{aligned}$$

## 12.3 Detaillierte Berechnung zum Beispiel zur “ $\alpha + 1$ -Methode”

Berechnung der Ergebnisse zur Abb 8.2.4:

$$UserSampleSize = 0$$

(Datensatz mit 40

Beobachtungen)

$$\begin{aligned} p(D|S^1) &= \frac{\Gamma(2)}{\Gamma(42)} \cdot \frac{\Gamma(21)}{\Gamma(1)} \cdot \frac{\Gamma(21)}{\Gamma(1)} \cdot \frac{\Gamma(2)}{\Gamma(42)} \cdot \frac{\Gamma(21)}{\Gamma(1)} \cdot \frac{\Gamma(21)}{\Gamma(1)} \\ &= \frac{1}{3.34525 \cdot 10^{49}} \cdot \frac{2.43290 \cdot 10^{18}}{1} \cdot \frac{2.43290 \cdot 10^{18}}{1} \cdot \frac{1}{3.34525 \cdot 10^{49}} \cdot \frac{2.43290 \cdot 10^{18}}{1} \cdot \frac{2.43290 \cdot 10^{18}}{1} \\ &= 3.1307 \cdot 10^{-26} \end{aligned}$$

$$\begin{aligned} p(D|S^2) &= \frac{\Gamma(2)}{\Gamma(42)} \cdot \frac{\Gamma(21)}{\Gamma(1)} \cdot \frac{\Gamma(21)}{\Gamma(1)} \cdot \frac{\Gamma(2)}{\Gamma(22)} \cdot \frac{\Gamma(2)}{\Gamma(22)} \cdot \frac{\Gamma(13)}{\Gamma(1)} \cdot \frac{\Gamma(9)}{\Gamma(1)} \cdot \frac{\Gamma(9)}{\Gamma(1)} \cdot \frac{\Gamma(13)}{\Gamma(1)} \\ &= \frac{1}{3.34525 \cdot 10^{49}} \cdot \frac{2.43290 \cdot 10^{18}}{1} \cdot \frac{2.43290 \cdot 10^{18}}{1} \cdot \frac{1}{5.10909 \cdot 10^{19}} \cdot \frac{1}{5.10909 \cdot 10^{19}} \cdot \frac{479001600}{1} \cdot \frac{40320}{1} \cdot \frac{40320}{1} \cdot \frac{479001600}{1} \\ &= 2.5284 \cdot 10^{-26} \end{aligned}$$

$$UserSampleSize = 20$$

(Halber Datensatz

andere Hälfte zuvor

als Vorwissen bekannt)

$$\begin{aligned} p(D|S^1) &= \frac{\Gamma(22)}{\Gamma(42)} \cdot \frac{\Gamma(21)}{\Gamma(11)} \cdot \frac{\Gamma(21)}{\Gamma(11)} \cdot \frac{\Gamma(22)}{\Gamma(42)} \cdot \frac{\Gamma(21)}{\Gamma(11)} \cdot \frac{\Gamma(21)}{\Gamma(11)} \\ &= \frac{5.10909 \cdot 10^{19}}{3.34525 \cdot 10^{49}} \cdot \frac{2.43290 \cdot 10^{18}}{3628800} \cdot \frac{2.43290 \cdot 10^{18}}{3638800} \cdot \frac{5.10909 \cdot 10^{19}}{3.34525 \cdot 10^{49}} \cdot \frac{2.43290 \cdot 10^{18}}{3628800} \cdot \frac{2.43290 \cdot 10^{18}}{3628800} \\ &= 4.7128 \cdot 10^{-13} \end{aligned}$$

$$\begin{aligned} p(D|S^2) &= \frac{\Gamma(22)}{\Gamma(42)} \cdot \frac{\Gamma(21)}{\Gamma(11)} \cdot \frac{\Gamma(21)}{\Gamma(11)} \cdot \frac{\Gamma(12)}{\Gamma(22)} \cdot \frac{\Gamma(12)}{\Gamma(22)} \cdot \frac{\Gamma(13)}{\Gamma(7)} \cdot \frac{\Gamma(9)}{\Gamma(5)} \cdot \frac{\Gamma(9)}{\Gamma(7)} \cdot \frac{\Gamma(13)}{\Gamma(5)} \\ &= \frac{5.10909 \cdot 10^{19}}{3.34525 \cdot 10^{49}} \cdot \frac{2.43290 \cdot 10^{18}}{3628800} \cdot \frac{2.43290 \cdot 10^{18}}{3638800} \cdot \frac{39916800}{5.10909 \cdot 10^{19}} \cdot \frac{39916800}{5.10909 \cdot 10^{19}} \cdot \frac{479001600}{720} \cdot \frac{40320}{24} \cdot \frac{40320}{720} \cdot \frac{479001600}{24} \\ &= 5.2346 \cdot 10^{-13} \end{aligned}$$

## 12.4 Ausgabe des Simulations-Tools in Kapitel 10

### 12.4.1 DAG in Matrix-Form

Zunächst wird das Original-DAG in der Matrix-Form angezeigt:

```
DAG
      A      B      C      D      E
A
B      TRUE
C      TRUE
D      TRUE      TRUE
E
```

### 12.4.2 Die bedingten Wahrscheinlichkeiten zum DAG

Nach der DAG-Ausgabe werden die per Zufallsgenerator erzeugten bedingten Wahrscheinlichkeiten angezeigt (diese werden immer identisch sein solange `dag=testDAG` und `rnd=1` angegeben wird):

```
LENGTH-VECTOR (states per dimension):
LengthVektor
  A B C D E
[1] 2 2 2 2 2
```

```
MIN-VECTOR (minimal state):
MinVektor
  A B C D E
[1] 1 1 1 1 1
```

TABLE 1 (NODE A):

```
( ... , ... , 1, 1, 1 )
[      1][      2]
[1] 0.525063 0.474937
[2] 0.179936 0.820064

( ... , ... , 1, 2, 1 )
[      1][      2]
[1] 0.0752212 0.924779
[2] 0.02331 0.97669
```

TABLE 3 (NODE C):

```
( ... , ... , 1, 1, 1 )
[      1]
[1] 0.89916
[2] 0.927449

( ... , ... , 2, 1, 1 )
[      1]
[1] 0.10084
[2] 0.0725514
```

TABLE 5 (NODE E):

```
( ... , ... , 1, 1, 1 )
[      1]
[1] 0.614144

( ... , ... , 1, 1, 2 )
[      1]
[1] 0.385856
```

TABLE 2 (NODE B):

```
( ... , ... , 1, 1, 1 )
[      1]
[1] 0.711409
[2] 0.288591
```

TABLE 4 (NODE D):

```
( ... , ... , 1, 1, 1 )
[      1]
[1] 0.716561
[2] 0.747934

( ... , ... , 1, 2, 1 )
[      1]
[1] 0.283439
[2] 0.252066

( ... , ... , 1, 1, 2 )
[      1]
[1] 0.124207
[2] 0.36658

( ... , ... , 1, 2, 2 )
[      1]
[1] 0.875793
[2] 0.63342
```

Das Tafel-Objekt enthält, neben den eigentlichen bedingten Wahrscheinlichkeitstabellen, noch einmal extra die Anzahl der Ausprägungen jedes Knotens, bzw. jeder Dimension (`lengthVector`) sowie die Nummer der kleinsten Ausprägung (`minVector`). Auf diese Weise muß die Nummerierung nicht unbedingt mit 1 beginnen, da die Zustände binomial verteilter Merkmale oft auch mit 0 und 1 statt 1 und 2 kodiert werden.

### 12.4.3 Die “Unwissen”-Tafeln

Die “Unwissen-Tafeln” zum kantenlosen DAG (Ausgangspunkt des Lern-Algorithmus’) haben folgendes Aussehen:

```
> dagzerotabs

$MinVektor:
kredit lfd.konto laufzeit moral nutzung hoehe sparkonto beschzeit
  0      1      1      1      1      1      1      1
ratenhoehe fam.geschl buergen jetzt.wohn vermoegen alter weit.raten wohnung
  1      1      1      1      1      1      1      1
bish.raten beruf personen telefon gastarb
  1      1      1      1      1

$LengthVektor:
kredit lfd.konto laufzeit moral nutzung hoehe sparkonto beschzeit
  2      3      3      3      4      3      3      3
ratenhoehe fam.geschl buergen jetzt.wohn vermoegen alter weit.raten wohnung
  3      4      3      2      4      4      2      3
bish.raten beruf personen telefon gastarb
  2      4      2      2      2

$DagTables:
$DagTables[[1]]:

, , 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
[,1]
[1,] 0.5
[2,] 0.5

$DagTables[[2]]:

, , 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
[,1] [,2] [,3]
[1,] 0.3333333 0.3333333 0.3333333

$DagTables[[3]]:

, , 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
[,1]
[1,] 0.3333333

, , 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
[,1]
[1,] 0.3333333

, , 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
[,1]
[1,] 0.3333333

... Tafeln 4 bis 21 folgen ...
```

Aus Platzgründen sind nur die ersten zwei Tafeln abgedruckt. Da die Tafeln zu einem kantenlosen DAG (`kreditDAG`) gehören, sind sie jeweils faktisch nur eindimensional, liegen aber entsprechend der festen Zuordnung der Merkmale im 21-dimensionalen Raum (siehe Anhang 12.1.2).

## 12.5 Konventionen

Beschreibung	Bsp.	Anmerkung
Merkmale beginnen mit einem großen Buchstaben	<i>Beruf</i> $A, B, C, D, E$ $X, X_1, \dots, X_n$	konkretes Merkmal in Beispielen allgemeine Bezeichnung
Ausprägungen können teilweise auch mit einem Großbuchstaben beginnen aber und	<i>Kopf</i> $a_1, a_2, \dots, a_{r_A}$ $X_i = k$	Ausprägung beim Münzwurf Ausprägungen von $A$ Nr. d. Auspräg. allg. Merkmale
Mengen von Merkmalen werden groß und fett gedruckt	<b>Pa<sub>i</sub></b>	
Einzelne Werte (auch Indizes) werden klein geschrieben außer die Großschreibung bleibt ohne Mißverständnisse	$i, j, k, r_i, \dots, kopf$ <i>UserSampleSize</i>	<i>kopf</i> Zähler v. <i>Wurf</i> = <i>Kopf</i>
Wahrscheinlichkeitsverteilungen ( $\equiv$ mehrdim. Tafel)	$p(\dots)$	
Gemeinsame Wahrscheinlichkeitsverteilung	$p(A, B, C)$	v. $A, B$ und $C$
Bedingte Wahrscheinlichkeitsverteilung	$p(A, B C, D)$	v. $A, B$ gegeben $C$ u. $D$
Sonstige Tafeln	$N_i, \alpha_i$	$N_{ijk}$ = Zelle d. Tafel $N_i$
Technische Werte wie Parameter von Funktionen und Programmbeispiele werden in Schreibmaschinenschrift gedruckt	<b>a.matrix</b>	
“Für alle ...”	$\forall \dots$	mathem. Schreibweise
Mengen (mit konkreter Elementangabe)	$\mathbf{M} = \{\dots\}$	$\mathbf{M}$ ist Mengenbezeichner
Leere Menge	$\emptyset$	
Element einer Menge	$x \in \mathbf{M}$	$x$ ist aus $\mathbf{M}$
Vereinigungsmenge aus $\{1, 2, 3\}$ u. $\{4, 5\}$	$\{1, 2, 3\} \cup \{4, 5\}$	$= \{1, 2, 3, 4, 5\}$
Schnittmenge aus $\{1, 2, 3\}$ u. $\{2, 5\}$	$\{1, 2, 3\} \cap \{2, 5\}$	$= \{2\}$
Vereinigungsmenge der Mengen $\mathbf{M}_1, \dots, \mathbf{M}_n$	$\bigcup_{i=1}^n \mathbf{M}_i$	
Schnittmenge der Mengen $\mathbf{M}_1, \dots, \mathbf{M}_n$	$\bigcap_{i=1}^n \mathbf{M}_i$	
Mengensubtraktion: $\{1, 2, 3\}$ ohne $\{2, 5\}$	$\{1, 2, 3\} \setminus \{2, 5\}$	$= \{1, 3\}$
Absolutwert von $x$	$ x $	$ 5  = 5, \quad  -5  = 5$
äquivalent	$\equiv$	
“Daraus folgt”	$\Rightarrow$	
$x$ ist Element aus der Menge $\{1, 2, 3\}$	$x \in \{1, 2, 3\}$	
Fakultät	$n!$	$= 1 \cdot 2 \cdot \dots \cdot n$
Griechische Buchstaben	$\alpha$ $\theta$ $\Gamma()$	“alpha” (Vorwissen) “theta” (Parameter) “Gamma”-Funktion

## Literatur

- [Heckerman (March 1995)] A Tutorial On Learning With Bayesian Networks, David Heckerman, March 1995 (Revised 1996)
- [Jensen (1996)] An Introdduction to Bayesian Networks, Finn v. Jensen, 1996
- [Cooper & Herskovits (1992)] A Bayesian Method For The Introduction Of Probabilistic Networks From Data, G. Cooper, and E. Herskovits, 1992
- [Jensen et al. (1990)] Bayesian Updating In Recursive Graphical Models By Local Computations. Finn v. Jensen, S. Lauritzen and K. Olsen, 1990
- [Lauritzen & Spiegelhalter (1988)] Local Computations With Probabilities On Graphical Structures And Their Application To Expert Systems. S. Lauritzen and D. Spiegelhalter, 1988
- [Shafer & Shenoy (1990)] Shafer & Shenoy, 1990
- [Shachter (1988)] Probabilistic Inference And Influence Diagrams. R. Shachter 1988
- [Pearl (1986)] Fusion, Propagation And Structuring In Belief Networks J. Pearl 1986
- [Dawid (1992)] Applications Of General Propagation Algorithm For Probabilistic Expert Systems. P. Dawid 1992
- [D'Ambrosio (1991)] Local Expression Languages For Probabilistic Dependence, B. D'Ambrosio 1991
- [Shachter & Kenley (1989)] Gaussian influence diagrams, Shachter & Kenley, 1989
- [Lauritzen (1992)] Propagation Of Probabilities, Means, And Variances In Mixed Graphical Association Models. Lauritzen 1992
- [Chickering et al. (1995)] Learning Bayesian Networks: Search Methods And Experimental Results. D. Chickering, D. Geiger und D. Heckerman, 1995
- [Schwarz (1978)] Estimating The Dimension Of A Model. G. Schwarz 1978
- [Fahrmeir / Hamerle / Tutz (1996)] Multivariate Statistische Verfahren. 2. bearbeitete Auflage, de Gruyter, Berlin (u.a.) Fahrmeir, Ludwig / Hamerle, A. / Tutz, Gerhard (1996)
- [Fahrmeir / Tutz (1994)] Multivariate Statistical Modeling Based On Generalized Linear Models. Springer, New York. Fahrmeir, Ludwig / Tutz, Gerhard (1994)