Bayes Netze

Ludwig-Maximilians-Universität München
Institut:
Statistik

Master Seminar:
Statistische Modellierung latenter Strukturen in den Lebens-, Sozial- und Wirtschaftswissenschaften

Autor:
Thomas Welchowski (3. Semester Master Statistik)

Betreuer:
Paul Fink

15. Januar 2014

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

ARTIVA = Autoregressive time varying models
BIC = Bayes Schwarz Informationskriterium
CPQ = Conditional probability query
DAG = Directed acyclic graph
d-separation = directed graph separation
GS = Grow-Shrink
HC = Hill-Climbing
RMSE = Root mean squared error

1 Ziele und Überblick

Diese Seminararbeit beschäftigt sich mit bayesianischen Netzwerken. Es wird eine Literaturrecherche ausgeführt mit dem Ziel folgende Fragen adequat zu beantworten:

- Was sind bayesianische Netzwerke?
- Wie funktioniert deren Schätzung?
- Wie kann man Informationen aus einem bayesianischen Netzwerk gewinnen?
- Was sind deren Vor- und Nachteile im Vergleich zu etablierten Methoden?

Nach der Erläuterung von Grundbegriffen werden diese Fragen im Theorieteil in Kapitel 2 beantwortet. Im Anschluss daran wird in einer kurzen Simulation untersucht, inwiefern ein statisches, bayesianisches Netzwerk eine vorhandene Struktur in einem einfachen Beispiel erkennen kann.

2 Theorie

In diesem Kapitel werden einerseits die nötigen Grundbegriffe zum Verständnis und andererseits verschiedene Methoden zur Inferenz von bayesianischen Netzwerken aufgeführt.

2.1 Grundbegriffe der Graphentheorie

Ein Graph G=(K, V) besteht aus einer Menge von Knoten K und einer Menge von Verknüpfungen V [Bang-Jensen, 2010, vgl. S. 2]. Es gibt 3 grundlegende Klassen von Verknüpfungen zwischen zwei Knoten: Ungerichtet, gerichtet und gemischt [Bang-Jensen, 2010, vgl. S. 18 ff]. Hierzu werden kurz ein paar Beispiele gegeben: Zuerst ein gerichteter Graph G_1 mit Knoten K_1 und Verknüpfungen V_1:

$$G_1 = (K_1 = \{A, B, C, D\}; V_1 = \{A \to C, B \to C, C \to D\}) \tag{1}$$
$$G_2 = (K_2 = \{A, B, C, D\}, V_2 = \{A \leftrightarrow B, A \leftrightarrow C, B \leftrightarrow C, B \leftrightarrow D\}) \tag{2}$$
$$G_3 = (K_3 = \{A, B, C, D, E\}; V_3 = \{A \to B, B \leftrightarrow C, C \to E, D \to E\}) \tag{3}$$

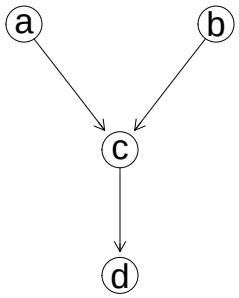

Abbildung 1: Beispiel gerichteter Graph

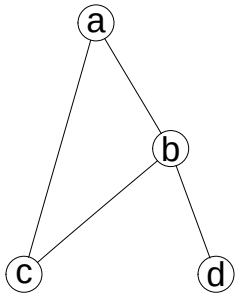

Abbildung 2: Beispiel ungerichteter Graph

G_1 ist ein gerichteter, G_2 ein ungerichteter und G_3 ein gemischter Graph. Des Weiteren gibt es azyklische (z. B. G_1) sowie zyklische Graphen (z. B. G_2 und G_3). Bei einem zyklischen Graphen gibt es mögliche Pfade, bei denen sich die Reihenfolge der besuchten Knoten wiederholt.

Für die statistische Analyse sind vor allem gerichtete, azyklische Graphen (DAG) von zentraler Bedeutung. Diese haben unter anderem folgende Eigenschaften:

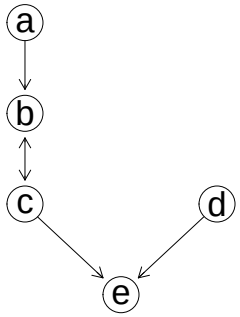

Abbildung 3: Beispiel gemischter Graph

- Jeder DAG hat mindestens einen Anfangs- und Endknoten [Bang-Jensen, 2010, vgl. S. 32]
- Jeder DAG besitzt eine azyklische Reihenfolge an Knoten
- Jeder Graph kann in linearer Zeit O(Anzahl Knoten + Anzahl Verknüpfungen) mit dem DFSA Algorithmus [Bang-Jensen, 2010, s. S. 33] auf Zyklen geprüft werden

Ein Anfangsknoten besitzt nur ausgehende Verknüpfungen und ein Endknoten hat nur eingehende Verknüpfungen. Unter einer azyklischen Reihenfolge versteht man eine Menge an Knoten $x_1, \ldots, x_n$ bei denen für alle Verknüpfungen x_i, x_j i<j gilt. Das heißt es darf kein Knoten in einem Pfad mehrmals vorkommen. Der DFSA Algorithmus hat folgendes Prinzip: Suche alle Anfangsknoten und notiere diese. Im weiteren Schritt gehe von jedem Anfangsknoten zu denjenigen Knoten, welche keine weiteren eingehenden Verbindungen aus dem vorherigen Anfangsknoten haben. Dies wird solange fortgeführt bis alle möglichen Pfade bekannt sind. Man prüft danach, ob diese eine azyklische Reihenfolge aufweisen. Dies bedeutet, dass in der Reihenfolge kein Knoten zweimal vorkommen darf.

Es gibt drei fundamentale Verbindungsarten zwischen drei Knoten [Jensen and Nielsen, 2007, vgl. S. 26-29]: Die konvergierende, die serielle und die divergierende Verbindung 4. Zwei Verbindungen zwischen drei Knoten konvergieren, falls ein Knoten keine ausgehenden Verbindungen hat. Bei einer seriellen Verbindung gibt es eine direkte Reihenfolge, d. h. $V = \{A \to B, B \to C\}$. Die divergierende Verbindung besteht, wenn ein Knoten zwei ausgehende Verbindungen zu den beiden anderen Knoten besitzt.

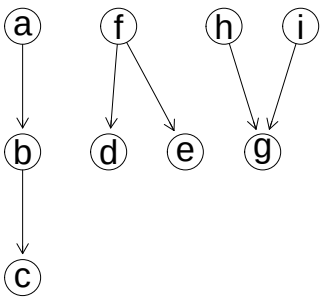

Abbildung 4: Von Links nach Rechts: Serielle, divergierende und konvergierende Verbindung

Bei der seriellen Verbindung ist c bedingt unabhängig von a gegeben b. Im Falle der divergierenden Verknüpfung sind d, e bedingt unabhängig gegeben f. Aber h, i sind im Allgemeinen nicht bedingt unabhängig gegeben g.
Mit diesen Eigenschaften lässt sich die wichtige Eigenschaft der d-Separation (directed graph separation) aus der Graphentheorie beschreiben. Ein Knoten A und ein Knoten B sind d-separated wenn folgende Bedingungen erfüllt sind [Jensen and Nielsen, 2007, s. S. 30]:

- Zwischen allen Pfaden von A nach B existiert ein dazwischenliegender Knoten V in der Form, so dass
- eine serielle oder divergierende Verbindung vorliegt und der Wert von V bekannt ist oder
- die Verbindung konvergiert, allerdings darf weder V selbst noch irgend ein Kinderknoten von V bekannt sein.

Wenn zwei Knoten A, B durch einen dritten Knoten V d-separiert werden, so sind sie bedingt unabhängig gegeben V. So hat man ein relativ einfaches Kriterium, um bedingte Unabhängigkeit von zwei Knoten innerhalb eines Graphen zu prüfen und kann damit die Inferenz vereinfachen.

2.2 Einführung in statische, bayesianische Netzwerke

Bayes Netze wurden ursprünglich mit dem Ziel entwickelt, menschliche Denkprozesse von Experten besser am Computer abbilden zu können. [Pearl, 1985, vgl. S. 3-5]. Da menschliches Denken stets subjektiv, unsicher ist und auf unvollständigen Informationen basiert bildet die Wahrscheinlichkeitstheorie einen guten Startpunkt zur Modellbildung. Nimmt man beispielsweise eine multivariate Verteilung für eine gegebene Anzahl an Variablen an, so ist es aufgrund der kombinatorischen Explosion zu aufwendig und ineffizient alle möglichen Randverteilungen daraus abzuleiten. Des Weiteren widerspricht es menschlichem Verhaltens in komplexen Entscheidungssituationen. Ein Experte schließt für die konkrete Fragestellung irrelevante Inhalte aus und fokussiert sich auf einen kleinen, überschaubaren Teil, um seine Entscheidung zu treffen. Diese Idee macht man sich zu nutze, um die Komplexität des einzuschränken. Man nehme die multivariate Wahrscheinlichkeitsverteilung $P(X_1, \ldots, X_p)$. Von Interesse sind Hypothesen H der Form $P(H_1, \ldots, H_k | E_1, \ldots E_l)$ mit Vorwissen bzw. Evidenzen E. Hierzu verwendet man DAGs, d. h. gibt zwischen zwei Variablen nur jeweils eine Verknüpfung. Dabei soll der modellierte Zusammenhang einerseits vollständig beschrieben werden als auch konsistent sein [Pearl, 1985, vgl. S. 3-5]. Konsistenz ist wichtig, da sonst die gezogenen Schlüsse davon abhängen, welcher Parameter betrachtet wird. Damit beide Kriterien erfüllt sind, stellt man die multivariate Verteilung mit Hilfe des Faktorisierungssatzes aus der Wahrscheinlichkeitstheorie dar: $P(X_1, X_2, \ldots, X_p) = P(X_1 | X_2, \ldots, X_p) P(X_2 | X_3, \ldots, X_p) \cdots P(X_p)$ Die hierfür notwendige Struktur des DAG kann entweder von einem Experten stammen oder anhand der Daten gelernt werden. Wie dies passiert wird im Kapitel 2.3 beschrieben.

2.3 Strukturlernen in statischen, bayesianischen Netzwerken

Strukturlernen bedeutet, die Struktur eines Graphen aus den gegebenen Daten zu schätzen. Jeder Knoten repräsentiert dabei eine Variable des Datensatzes. Grundsätzlich gibt

es drei verschiedene Typen von Verfahren: Constraint based learning, Score based learning oder hybrid learning. Hybrid learning ist eine Verbindung aus den beiden Verfahren. Da in der Simulation der Schwerpunkt auf das hybrid learning gelegt wird, werden die beiden anderen Verfahren nur kurz erläutert. Als Alternative kann man Expertenwissen für die Konstruktion des Graphen verwenden. Statisch bedeutet in diesem Kontext, dass Querschnittsdaten ohne Berücksichtigung der Zeit vorliegen.
Bevor die Struktur eines Netzwerks geschätzt werden kann, muss man einige Annahmen treffen. Es gibt drei verschiedene Typen von Bayesnetzen: Diskret, stetig und gemischt diskret-stetig. Beispielsweise im diskreten Fall werden alle Variablen des Netzwerks als diskret vorausgesetzt. Diese Unterscheidung ist wichtig, da es für jedes dieser Fälle andere Bewertungskriterien im *constraint based learning* als auch im *score based learning* gibt. Außerdem bestimmen diese Unterscheidungen die Likelihood im parameter learning (siehe Kapitel 2.4). Theoretisch ist es schwierig viele Arten von Verteilungen in einem Modell zuzulassen, da für eine Maximum Likelihood Schätzung die bedingten Dichten hergeleitet werden müssen.
Es ist nicht möglich aus allen theoretischen Fällen die beste Alternative zu selektieren, denn Fallanzahl steigt exponentiell mit der Zahl an Variablen an [Santini, 2006]. Deswegen muss man auf iterative Algorithmen zurückgreifen, welche entweder mit einem saturierten oder leeren Graphen zur Selektion beginnen.

2.3.1 Binning als Datenvorbereitung

In dieser Arbeit werden keine stetigen oder diskret-gemischten Bayesnetze betrachtet. Falls in einem Datensatz stetige Variablen auftauchen können diese mittels Binning diskretisiert werden [Nagarajan et al., 2013, vgl. S. 23-24]. Auf der einen Seite verliert man dadurch Informationen, da nicht mehr alle einzelnen Daten verfügbar sind. Andererseits können diskrete Strukturen in der Regel computational effizienter geschätzt werden. Des Weiteren ist eine Multinomialverteilung bei diskreten Strukturen ziemlich generell und bringt weniger Annahmen mit sich, als für den stetigen Fall z. B. eine Normalverteilung anzunehmen. Falls eine höhere Präzision nötig ist, können die Anzahl Splits erhöht werden.
Wenn kein Expertenwissen zur Einteilung der Klassen von stetigen Zufallsvariablen verfügbar ist, gibt es eine Vielzahl von Algorithmen, um einen Ausgleich zwischen Genauigkeit der Datenstruktur und numerischer Effizienz herzustellen. Für eine Übersicht und Referenzen zur entsprechender Fachliteratur wird auf verwiesen [Kotsiantis and Kanellopoulos, 2006].
Als Beispiel sei hier der Ameva Algorithmus gegeben [Gonzalez-Abril et al., 2009]. Das zu optimierende Kriterium setzt sich wie folgt zusammen [Kotsiantis and Kanellopoulos, 2006, s. S. 5328]:

$$\chi^2(k) = N\left(-1 + \sum_{i=1}^{l}\sum_{j=1}^{k}\frac{n_{ij}^2}{n_{i\cdot}n_{\cdot j}}\right) \tag{4}$$

$$\text{Ameva}(k) = \frac{\chi^2(k)}{k(l-1)};\ l \geq 2 \tag{5}$$

Das Ameva-Kriterium basiert auf dem Kontingenzkoeffizienten und einer Adjustierung in Abhängigkeit der Parameteranzahl im Nenner zusammen. Der Kontingenzkoeffizient misst wie stark sich bei zwei diskreten Merkmalen die Anzahl der Fälle, je Zelle einer

Häufigkeitstabelle, von den unter Unabhängigkeit erwarteten Fallanzahl quadratisch unterscheiden. Es lässt sich zeigen [Gonzalez-Abril et al., 2009, s. S. 5329], dass $\chi^2(k)$ mit monoton mit der Anzahl der Klassen ($k \geq 2$) steigt. Bei der Optimierung wird das Maximum gesucht. Somit maximiert das Ameva-Kriterium die Abhängigkeit zwischen C und L und versucht dabei möglichst wenige Intervalle zu verwenden. Für eine stetige, zu diskretisierende Variable wird eine Häufigkeitstabelle erstellt, welche sich folgendermaßen zusammensetzt:

$C_i \vert L_j$	L_1	$\cdots$	L_j	$\cdots$	L_k	$\sum n_{\cdot j}$
C_1	n_{11}	$\cdots$	n_{1j}	$\cdots$	n_{1k}	$n_{1\cdot}$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
C_i	n_{i1}	$\cdots$	n_{ij}	$\cdots$	n_{ik}	$n_{i\cdot}$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
C_l	n_{l1}	$\cdots$	n_{lj}	$\cdots$	n_{lk}	$n_{l\cdot}$
$\sum n_{i\cdot}$	$n_{\cdot 1}$	$\cdots$	$n_{\cdot j}$	$\cdots$	$n_{\cdot k}$	N

Tabelle 1: Kontingenztabelle zur Diskretisierung einer Variable

Zunächst wird die Anzahl der Klassen C_i bestimmt. Dazu formt man alle stetigen Werte in ganzzahlige um. Die Anzahl der ganzzahligen Werte entspricht l. Jede Zelle n_{ij} entspricht der Anzahl der Werte C_i, welche in das entsprechende Intervall L_j fallen. N ist die Anzahl aller Beobachtungen. Die ursprünglichen Werte werden nach aufsteigender Reihenfolge geordnet. Der Algorithmus ist ein Top-Down Verfahren. Somit wird ausgehend von einer Intervallmenge, bei jeder Iteration ein neuer Splitpunkt gesucht. Falls sich das Kriterium verbessert, wird die Iteration fortgesetzt. Wenn es keine Verbesserung gibt, bricht der Algorithmus ab.
Der Vorteil von Ameva ist, dass die Anzahl sowie die Lage der Intervalle automatisch gewählt werden, ohne das vom Benutzer weitere Parameter spezifiziert werden. Im Vergleich zu anderen Algorithmen ist Ameva oft computational effizienter und somit auch für größe Datensätze anwendbar.

2.3.2 Constraint based learning

Dieses Verfahren basiert auf bedingten Unabhängigkeitstests. Für den diskreten Fall wird der klassische, bedingte Pearson X^2 Unabhängigkeitstest verwendet. Dieser lautet wie folgt [Nagarajan et al., 2013, s. S. 21]:

$$H_0 : X \perp Y; H_1 : X \not\perp Y \tag{6}$$

$$X^2\left(X, Y | Z_1, \ldots, Z_p\right) = \sum_{i=1}^{R}\sum_{j=1}^{C}\sum_{k=1}^{L}\frac{\left(n_{ijk} - m_{ijk}\right)^2}{m_{ijk}};\ m_{ijk} = \frac{n_{i+k}n_{+jk}}{n_{++k}} \tag{7}$$

Es wird getestet, ob die zwei diskreten Zufallsvariablen X und Y stochastisch, gegeben einer Menge von weiteren Zufallsvariablen, unabhängig sind. Asymptotisch ist die Statistik unter der Nullhypothese $\chi^2_{(R-1)(C-1)L}$ verteilt. Man kann sich vorstellen, dass hier eine dreidimensionale Kontingenztafel getestet wird. Unter der Nullhypothese erwartet man als Ergebnisse einer inneren Zelle innerhalb einer Kontingenztabelle das Produkt der entsprechenden Zeilen- und Randsummen m_{ijk}. Falls die Summe der quadratischen Abstände zwischen den Zellen zu groß ist, wird die Nullhypothese abgelehnt.

Darauf aufbauend gibt es eine Reihe von Algorithmen, um die Struktur eines Graphen zu schätzen. An dieser Stelle wird kurz vorgestellt, wie man mit dem GS (Grow-Shrink) Algorithmus ein Markov Blanket einer Variablen bestimmen kann. Ein *Markov Blanket* [Nagarajan et al., 2013, vgl. S. 16] einer Variable A gibt die kleinste Menge exklusive A an, welche die Wahrscheinlichkeitsverteilung vollständig beschreibt. Gegeben dieser Menge ist A vom Rest der anderen Zufallsvariablen außerhalb vom Markov Blanket stochastisch unabhängig. Diese Erkenntnis beruht auf der d-Separation (siehe Kapitel 2.1). Denn man kann zeigen, dass ein Knoten A durch sein Markov Blanket von dem Rest des Netzwerks direkt separiert wird. In einem Bayesnetz besteht das Markov Blanket aus den Kinder- und Elternknoten sowie den Elternknoten der Kinderknoten. Elternknoten sind jene Variablen, welche einen direkten Einfluss auf A haben. Kinderknoten werden direkt von A als Vorfolger beeinflusst. Der GS Algorithmus läuft wie folgt ab [Margaritis, vgl. S. 26-32]:

1. Beginne mit einer Menge, welche nur Knoten A enthält
2. In jedem Schritt führe einen Unabhängigkeitstest zwischen dem Anfangsknoten und eines Kandidaten, bedingt auf die Werte aller Variablen außer dem Startknoten innerhalb der Menge, durch
3. Füge so lange neue Variablen zur Menge hinzu bis keine mehr ein signifikantes Testergebnis vorweisen (growing Phase)
4. Es ist möglich, dass in der Wachstumsphase zu viele Variablen hinzugefügt worden sind. Deshalb sind noch jene zu entfernen, welche nicht im Markov-Blanket enthalten sein dürfen
5. Nachdem die 1. Phase abgeschlossen ist, beginnt das shrinking: Entferne solange Variablen aus der Menge, bis alle Variablen entfernt worden sind, bei welchen H_0 nicht verworfen wird

Danach ist das Markov Blanket von A bekannt. GS wird für alle vorhandenen Variablen zur Bestimmung des Markov Blanket eingesetzt. Um einen vollstängigen DAG mit constraint based learning zu erhalten sind noch weitere Anpassungschritte nötig, da bisher noch keine Verknüpfungen zwischen den Knoten vorhanden sind. Für Details wird auf [Margaritis, vgl. S. 34-37] verwiesen.

2.3.3 Score based learning

Score based learning basiert immer auf einem Score, der zur Optimierung entweder maximiert oder minimiert wird. Als Beispiel sei hier der sogenannte Hill-Climbing (HC) Algorithmus gegeben [Nagarajan et al., 2013, s. S. 19]:

1. Beginne mit einer gegebenen Netzwerkstruktur G (gewöhnlich leer)
2. Bewerte die Struktur mit einem geeigneten Kriterium (Score)
3. Bewerte alle möglichen Veränderungen jeder einzelnen Kante im Graphen (Hinzufügen, löschen, Änderung der Richtung von Verknüpfungen)
4. Es werden nur Graphen akzeptiert, welche nicht zu einem zyklischen Netzwerk führen

5. Nehme die beste Alternative als neue Ausgangsbasis

6. Wiederhole so lange die letzten beiden Schritte, bis sich der Score nicht mehr erhöht

Ein gängiges Kriterium zur Evaluation ist der BIC (Bayes Schwarz Informationskriterium). Im Falle von bayesianischen Netzwerken wird dieser wie folgt definiert:
$\sum_{i=1}^{N} \log\left(f_{X_i}(X_i|\prod X_i)\right) - \frac{d}{2}\log(n)$ Hierbei ist $\prod X_i$ die Menge der Eltern von X_i. Das BIC ist ein Kompromiss aus Datenanpassung durch die Likelihood sowie Komplexität des Modells durch den Bestrafungsterm $\frac{d}{2}\log(n)$. Um

2.3.4 Hybrid based learning

Hybrid Algorithmen wie der Sparse Candidate Algorithmus [Friedman et al., 1999b, s. S. 208] versuchen constraint based learning sowie score based learning Methoden zu verbinden:

1. Restriktion des Suchraums mit constraint based learning

2. Maximiere mit Hilfe von score based learning und schätze die Graphenstruktur

3. Wiederhole Schritte 1 und 2 basierend auf dem Netzwerk der vorherigen Iteration bis zur Konvergenz

Hierbei wird zuerst der Suchraum restringiert, um eine passende Menge an Elternknoten zu finden. Danach wird auf Basis der resultierenden Menge ein Score-basierter Algorithmus ausgeführt. In der folgenden Iteration werden im 1. Schritt neue Elternknoten außerhalb der bisher bekannten Elternknoten gesucht. Im 2. Schritt verwendet man die Vereinigungsmenge der Elternknoten aus dem 1. Schritt sowie der neu berechneten Elternknoten (Berechnung aus der Menge ohne Variablen des 1. Schritts) [Friedman et al., 1999b, s. S. 209]. Es hat sich in Simulationen [Friedman et al., 1999b] gezeigt, dass der Sparse Candidate Algorithmus zwei Vorteile mit sich bringt: Erstens wird durch die Einschränkung des Suchraums der Algorithmus computational effizienter und zweitens ist durch die Verwendung von Vorinformation aus der vorherigen Iteration eine genauere Auswahl von Elternknoten möglich, was tendenziell zu Netzwerken mit höheren Scores im 2. Schritt führt. Diese zwei Schritte erfolgen iterativ, bis sich entweder die Struktur des Netzwerks nicht mehr verändert oder der Score identisch bleibt [Friedman et al., 1999b].

2.3.5 Bagged structure learning

Aufbauend auf hybrid based learning gibt es eine weite allgemeine Methode bagging [Breiman, 1996], mit dem die Varianz von Schätzmethoden reduziert werden kann. Die Idee ist dabei $b = 1, \ldots, B$ nichtparametrische Bootstrapstichproben mit Zurücklegen zu ziehen. Für jede Bootstrapstichprobe wird die Struktur des DAG geschätzt [Friedman et al., 1999a]. Danach wird die relative Häufigkeit jeder Verknüpfung zwischen zwei Knoten über alle $b = 1, \ldots, B$ ermittelt. Desto höher die relative Häufigkeit ist, desto plausibler ist eine Verknüpfung zwischen zwei Knoten. Wenn dabei eine Verbindung häufiger als ein bestimmter Grenzwert ist, so nimmt man diese in den finalen DAG auf. Den Grenzwert kann man entweder vorgeben oder automatisch aus den Daten bestimmen lassen.
Zur automatischen Wahl des Grenzwerts hat Marco Scutari einen Ansatz entwickelt [Scutari and Nagarajan, 2013, vgl. S. 3-6]: Ziel ist es signifikante Verbindungen zu identifizieren. Zunächst ordnet man die relativen Häufigkeiten in aufsteigender Reihenfolge und

berechnet daraus die empirische Verteilungsfunktion. Es ist folgendes Kriterium zu optimieren:

$$L_1(t;\hat{p}_{(\cdot)}) = \int \left| F_{\hat{p}_{(\cdot)}}(x) - F_{\tilde{p}_{(\cdot)}}(x) \right| dx \tag{8}$$

$$L_1(t;\hat{p}_{(\cdot)}) = \sum_{x_i \in 0 \cup \hat{p}_{(\cdot)} \cup 1} \left| F_{\hat{p}_{(\cdot)}} - t \right| (x_{i+1} - x_i) \tag{9}$$

$$\Rightarrow \hat{t} = \underset{t \in [0,1]}{\operatorname{argmin}}\ L_1(t;\hat{p}_{(\cdot)}) \tag{10}$$

Man geht davon aus, dass es eine zugrundeliegende wahre Verteilungsfunktion gibt, welche ab einem gewissen Wert auf 1 springt und sonst 0 ist. Es soll der absolute Abstand beider Funktionen minimiert werden. Da das Kriterium nur andere Werte bei unterschiedlichen Werten der empirischen Verteilungsfunktion annimmt, kann sie mit Hilfe linearer Programmierung optimiert werden.

2.4 Parameter learning in statischen, bayesianischen Netzwerken

Nachdem die Struktur des DAG bekannt ist, muss der Zusammenhang noch quantifiziert werden. Wie bereits am Anfang des Kapitels 2.3 erläutert wurde, werden nur diskrete Zufallsvariablen angenommen. Durch Anwendung der Kettenregel kann die gemeinsame Wahrscheinlichkeitsverteilung als Produkt jeder einzelnen Variablen gegeben seiner Elternknoten dargestellt werden. Es wird eine Multinomialverteilung angenommen, dessen marginale Dichte wie folgt gegeben ist [Evans et al., 2000, s. S. 135]:

$$f(x_1, \ldots, x_k) = n! \prod_{j=1}^{k} \left(\frac{p_j}{x_j!} \right) \tag{11}$$

$$j = 1, \ldots, k;\ x_j \in \mathbb{N}_0;\ p_j \in (0,1);\ \sum_{i=1}^{k} p_j \tag{12}$$

Mit p_j als Wahrscheinlichkeiten der j-ten Kategorie. Diese stellt eine Verallgemeinerung der Binomialverteilung dar, denn die Randverteilungen einer Kategorie sind binomialverteilt. Es wird eine Maximum Likelihood Schätzung der Parameter verwendet. Man kann zeigen, dass dieser Schätzer der relativen Häufigkeit der jeweiligen Klasse im Datensatz entspricht. Wenn man nun jede Variable auf dessen Elternknoten bedingt, lassen sich die Parameterschätzungen in einer bedingten Wahrscheinlichkeitstabelle darstellen. Dies wird in einem einfachen Beispiel illustriert. Man nehme folgende konvergierende Verbindung zwischen 2 Elternknoten **V=Vorbereitung** sowie **W=Wissen** und einer interessierenden Variable **L=Leistung in einer Prüfung**. Der Wertebereich der Variablen sei wie folgt:

$$\begin{aligned} V &= \{\text{Ja}, \text{Nein}\} \\ W &= \{\text{Kein Vorwissen}, \text{Grundlagen}, \text{Fortgeschritten}\} \\ L &= \{\text{Note 1}, \text{Note 2}, \text{Note 3}, \text{Note 4}, \text{Note 5}\} \end{aligned}$$

Gegebene Daten lassen sich in folgender, fiktiver Tabelle darstellen:

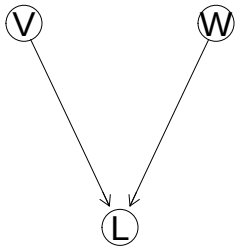

Abbildung 5: Beispiel zum Parameterlernen

Bedingung/Leistung	1	2	3	4	5
V=Ja, W=Kein Vorwissen	0.1	0.15	0.2	0.4	0.15
V=Ja, W=Grundlagen	0.15	0.2	0.45	0.1	0.1
V=Ja, W=Fortgeschritten	0.65	0.15	0.1	0.05	0.05
V=Nein, W=Kein Vorwissen	0.05	0.1	0.05	0.3	0.5
V=Nein, W=Grundlagen	0.15	0.2	0.25	0.2	0.2
V=Nein, W=Fortgeschritten	0.45	0.2	0.15	0.15	0.1

Tabelle 2: Bedingte Wahrscheinlichkeitstabelle als Beispiel

In jeder Zeile der Tabelle steht eine bedingte, diskrete Wahrscheinlichkeitsverteilung, dessen Wahrscheinlichkeiten sich zu 1 addieren. Die Zellen geben die relative Häufigkeit der jeweiligen Klasse an. Als Alternative bietet sich eine bayesianische Schätzung an [Friedman and Singer, 1999, vgl. S. 1-3]. Hierzu wird als priori die Dirichletverteilung

$$f(\theta_i, \ldots, \theta_k | \alpha_1, \ldots \alpha_k) \propto \prod_i^k \theta_i^{\alpha_i - 1};\ \sum_i \theta_i = 1;\ \theta_i \geq 0;\ \forall i = 1, \ldots k \tag{13}$$

angenommen. Die Dirichletverteilung ist die konjugierte priori zur Multinomialverteilung, d. h. die posteriori ist dirichletverteilt. Es kann gezeigt werden, dass der posteriori Erwartungswert sich als gewichtete Summe des priori Erwartungswerts sowie des ML-Schätzers darstellen lässt:

$$E(Y = i|\mathbf{x}, \alpha) = \frac{\sum_j \alpha_j}{\sum_j (\alpha_j + N_j)} \underbrace{\frac{\alpha_i}{\sum_j \alpha_j}}_{\text{priori EW}} + \frac{\sum_j N_j}{\sum_j (\alpha_j + N_j)} \underbrace{\frac{N_i}{\sum_j N_j}}_{\text{ML-Schätzer}} \tag{14}$$

Dabei ist Y die multinomialverteilte Zufallsvariable, α_i die priori Parameter aus der Dirichletverteilung für die i-te Klasse, und N_i die Anzahl der Beobachtungen in der Klasse i. Die priori Parameter kann man als Vorwissen über die priori Wahrscheinlichkeiten der i-ten Klasse interpretieren. Die bayesianische Schätzung hat unter anderem einige Vorteile im Vergleich gegenüber der ML-Schätzung:

- Schätzung ist robuster
- Bereichsschätzungen leichter interpretierbar
- Vorwissen in Schätzung verwendbar

2.5 Inferenz in statischen, bayesianischen Netzwerken

Nachdem die Struktur des DAG und die Parameter bekannt sind, will man Prognosen auf Basis von gegebenen Daten für unterschiedliche Szenarien erstellen. Man will die Wahrscheinlichkeitsverteilung des CPQ (Conditional probability query) $P(X_1, \ldots, X_n | Y_1, \ldots, Y_m)$ der unbekannten Zufallsvariablen X_i mit gegebenen Daten Y_i bestimmen. Im Unterschied beispielsweise zur Regressionsanalyse es ist zulässig mehr als eine Zielgröße zu betrachten. Es gibt einerseits exakte und andererseits approximative Algorithmen diese Verteilung zu bestimmen. In dieser Arbeit wird der approximative Algorithmus logic sampling vorgestellt, welcher methodisch auf rejection sampling beruht [Korb and Nicholson, 2011, s. S. 75-76]. Die Grundidee ist einfach und effektiv: Zuerst ordnet man die Variablen in eine topologische Reihenfolge in Abhängigkeit des Graphen, d.h. man beginnt bei den Knoten, welche keine Elternknoten haben und endet bei jenen, die keine Kinderknoten besitzen. In jeder Iteration zieht man nun eine Zufallszahl über alle Knoten hinweg. Die Ziehungen eines Knotens hängen von den Zufallszahlen der Elternknoten ab. Beispielsweise bei der Struktur des Graphen (Abbildung 6im Kapitel 3) geht der Algorithmus je Ziehung wie folgt vor: Zuerst wird aus den Wahrscheinlichkeitsverteilungen der Knoten A, C und F eine Zufallszahl gezogen. Gegeben dem gezogenen Wert aus A wird aus der bedingten Verteilung von B gezogen. Danach zieht man aus der bedingten Verteilunng von D gegeben A, C sowie für den Knoten E gegeben der Werte aus B und F. Damit erhält man einen Zustand des gesamten Netzwerks. Dies wird solange wiederholt bis die vorher spezifizierte Stichprobenumfang erreicht ist.

Nachdem eine genügend große Stichprobe gezogen ist, zählt man die Anzahl der Stichproben, bei welchen die vorher spezifizierten Bedingungen (Evidenzen) $Y_1 = b_1, \ldots Y_m = b_m$ zutreffen. Innerhalb dieser Menge wird die Anzahl einer interessierenden Kombinationen der diskreten Ausprägungen $X_1 = x_{i1}, \ldots X_n = x_{in}$ gezählt. x_{ik} bezeichnet dabei die Wahrscheinlichkeit der i-ten Kategorie der k-ten Zufallsvariable. Insgesamt erhählt man damit folgenden Schätzer:

$$\hat{P}(X_1 = x_{i1}, \ldots X_n = x_{in} | Y_1 = b_1, \ldots, Y_m = b_m) \tag{15}$$

$$= \frac{\hat{P}(X_1 = x_{i1}, \ldots X_n = x_{in}, Y_1 = b_1, \ldots, Y_m = b_m)}{\hat{P}(Y_1 = b_1, \ldots Y_m = b_m)} \tag{16}$$

$$= \frac{\#(X_1 = x_{i1}, \ldots, X_n = x_{in}, Y_1 = b_1, \ldots, Y_m = b_m)}{\#(Y_1 = b_1, \ldots, Y_m = b_m)} \tag{17}$$

Nach dem Gesetz der großen Zahlen konvergiert die relative Häufigkeit $\#(\cdot)$ gegen die wahre bedingte Wahrscheinlichkeit. Ein Vorteil des Algorithmus ist seine Einfachheit. Zudem sind die Ziehungen voneinander unabhängig und können parallelisiert werden. Ein Nachteil ist, dass falls eine Kombination an Evidenzen sehr unwahrscheinlich ist, eine relativ große Stichprobe gezogen werden muss, um valide Schätzungen zu bekommen.
Falls eine harte Prognose, d.h. eine exakte Zuordnung zu einer Klasse, notwendig ist, nimmt man die Klasse mit der höchsten prognostizierten Wahrscheinlichkeit.

3 Simulation

In diesem Kapitel wird anhand einer kurzen Simulation eines Datensatzes überprüft, wie gut bayesianische Netzwerke einen Zusammenhang erkennen können. Hierbei werden alle 3 Phasen zur Schätzung (Strukturlernen, Parameterlernen, Inferenz) getrennt untersucht. Im folgenden Kapitel 3.1 wird das Design des zu simulierenden Datensatzes erläutert. Im anschließenden Kapitel werden zu den verwendeten Methoden weitere Einstellungen und Annahmen kurz dargestellt. Gegeben dieser Daten soll nun untersucht werden inwiefern die theoretisch vorgestellten Methoden die Struktur, die Parameter

3.1 Design

Es wird folgende Graphenstruktur 6 für ein wahres Modell mit fiktiven sechs Knoten $\{A, B, \dots F\}$ angenommen:

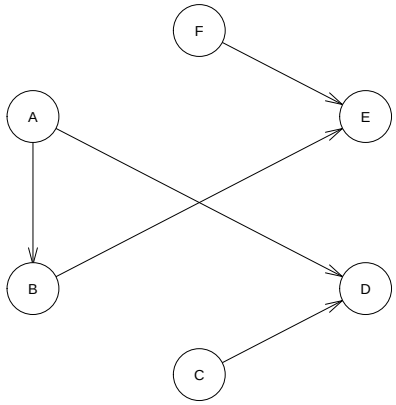

Abbildung 6: Wahrer DAG

Bei allen Knoten nimmt man an, dass die multinomialverteilt sind. Die Zufallsvariablen $A, \dots, E$ haben dabei jeweils drei mögliche Ausprägungen $\{a, b, c\}$ und Variable F zwei $\{a, b\}$. Die bedingten Wahrscheinlichkeitsverteilungen aller Zufallsvariablen lassen sich in Tabellen zusammenfassen:

	A=a	A=b	A=c
P(A)	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$

Tabelle 3: Wahrscheinlichkeitsverteilung für A

	B=a	B=b	B=c
P(B|A=a)	0.8	0.1	0.1
P(B|A=b)	0.4	0.2	0.4
P(B|A=c)	0.1	0.1	0.8

Tabelle 4: Bedingte Wahrscheinlichkeitsverteilung für B

	C=a	C=b	C=c
P(C)	0.75	0.2	0.05

Tabelle 5: Wahrscheinlichkeitsverteilung für C

	D=a	D=b	D=c
P(D|A=a, C=a)	0.8	0.1	0.1
P(D|A=a, C=b)	0.2	0.1	0.7
P(D|A=a, C=c)	0.4	0.2	0.4
P(D|A=b, C=a)	0.1	0.8	0.1
P(D|A=b, C=b)	0.9	0.05	0.05
P(D|A=b, C=c)	0.3	0.4	0.3
P(D|A=c, C=a)	0.1	0.1	0.8
P(D|A=c, C=b)	0.25	0.5	0.25
P(D|A=c, C=c)	0.15	0.45	0.4

Tabelle 6: Bedingte Wahrscheinlichkeitsverteilung für D

	E=a	E=b	E=c
P(E|B=a, F=a)	0.8	0.1	0.1
P(E|B=a, F=b)	0.4	0.5	0.1
P(E|B=b, F=a)	0.2	0.2	0.6
P(E|B=b, F=b)	0.3	0.4	0.3
P(E|B=c, F=a)	0.1	0.1	0.8
P(E|B=c, F=b)	0.25	0.5	0.25

Tabelle 7: Bedingte Wahrscheinlichkeitsverteilung für E

	F=a	F=b
P(F)	0.5	0.5

Tabelle 8: Wahrscheinlichkeitsverteilung für F

Daraus wurden unabhängig, identisch verteilt 5000 Beobachtungen gezogen. Eine Beobachtung enthält einen Zustand des vollständigen Netzwerks inklusive aller Variablen.

3.2 Angewandte Methoden, Einstellungen und CPQ Herleitungen

Zum Strukturlernen wird als Grundmethode im bagging (Kapitel 2.3.5) der Sparse Candidate Algorithmus (Kapitel 2.3.4) mit bedingtem Pearson X^2 Unabhängigkeitstest nach dem GS Algorithmus im 1. Schritt und HC im 2. Schritt angewandt. Dabei werden 1000 Bootstrapstichproben gezogen. Die Parameter werden bayesianisch (Kapitel 2.4) geschätzt. Hierbei nimmt man als priori für alle Parameter der Dirichletverteilung $\alpha = 10$ an. Damit sind die priori Erwartungswerte aller Klassen einer Zufallsvariable identisch, d.h. es wird eine nichtinformative priori verwendet. Bei der Inferenz findet das logic sampling (Kapitel 2.5) Verwendung. Die Anzahl der gezogenen Stichproben beim logic sampling ergibt sich aus 5000 je Parameter, d.h. in diesem Fall 205000. Je nachdem welche bedingte Wahrscheinlichkeiten untersucht werden sollen, variiert der tatsächliche Stichprobenumfang. Dabei sollen drei verschiedene bedingte Wahrscheinlichkeitsverteilungen berechnet werden: $P(D|A = c, C = b)$; $P(E|A = a)$; $P(A, B|E = c)$ Die erste Verteilung davon lässt sich direkt aus der bedingten Wahrscheinlichkeitstabelle 3.1von D ablesen $P(D|A = c, C = b) = (a = 0.25, b = 0.5, c = 0.25)$. Die zweite Wahrscheinlichkeitsverteilung ist nicht direkt aus den Tabellen ersichtlich sondern muss theoretisch hergeleitet werden. Dazu geht man den DAG schrittweise mit gegebenen Bedingungen durch und berechnet für die entsprechenden Pfade die Wahrscheinlichkeiten:

$$
\begin{aligned}
P(E=a|A=a) = {} & P(B=a|A=a)P(F=a)P(E=a|B=a,F=a)+ \\
& P(B=a|A=a)P(F=b)P(E=a|B=a,F=b)+ \\
& P(B=b|A=a)P(F=a)P(E=a|B=b,F=a)+ \\
& P(B=b|A=a)P(F=b)P(E=a|B=b,F=b)+ \\
& P(B=c|A=a)P(F=a)P(E=a|B=c,F=a)+ \\
& P(B=c|A=a)P(F=b)P(E=a|B=c,F=b) = \\
& 0.32+0.16+0.01+0.015+0.005+0.0125 = 0.5225 \\
P(E=b|A=a) = {} & P(B=a|A=a)P(F=a)P(E=b|B=a,F=a)+ \\
& P(B=a|A=a)P(F=b)P(E=b|B=a,F=b)+ \\
& P(B=b|A=a)P(F=a)P(E=b|B=b,F=a)+ \\
& P(B=b|A=a)P(F=b)P(E=b|B=b,F=b)+ \\
& P(B=c|A=a)P(F=a)P(E=b|B=c,F=a)+ \\
& P(B=c|A=a)P(F=b)P(E=b|B=c,F=b) = \\
& 0.04+0.2+0.01+0.02+0.005+0.025 = 0.3 \\
P(E=c|A=a) = {} & P(B=a|A=a)P(F=a)P(E=c|B=a,F=a)+ \\
& P(B=a|A=a)P(F=b)P(E=c|B=a,F=b)+ \\
& P(B=b|A=a)P(F=a)P(E=c|B=b,F=a)+ \\
& P(B=b|A=a)P(F=b)P(E=c|B=b,F=b)+ \\
& P(B=c|A=a)P(F=a)P(E=c|B=c,F=a)+ \\
& P(B=c|A=a)P(F=b)P(E=c|B=c,F=b) = \\
& 0.04+0.04+0.03+0.015+0.04+0.0125 = 0.1775
\end{aligned}
$$

Die dritte Wahrscheinlichkeitsverteilung ist in zwei Aspekten schwieriger zu berechnen: Erstens wird der kausale Zusammenhang umgedreht, d.h. man will wissen welcher Input am plausibelsten für einen gegebenen Endknoten ist. Des Weiteren wird hier nicht nur eine Zielgröße sondern die gemeinsame Wahrscheinlichkeitsverteilung von zwei Variablen betrachtet. Um diese Wahrscheinlichkeiten theoretisch bestimmen zu können wendet man den Satz von Bayes an:

$$P(A=a,B=a|E=c) = \frac{P(E=c|A=a,B=a)\ P(A=a,B=a)}{P(E=c)} =$$
$$\frac{P(E=c|B=a)\ P(B=a|A=a)\ P(A=a)}{P(E=c)} =$$
$$\frac{P(E=c,F=a|B=a)\ P(B=a|A=a)\ P(A=a)}{P(E=c)} +$$
$$\frac{P(E=c,F=b|B=a)\ P(B=a|A=a)\ P(A=a)}{P(E=c)} =$$
$$\frac{P(E=c|B=a,F=a)\ P(F=a)\ P(B=a|A=a)\ P(A=a)}{P(E=c)} +$$
$$\frac{P(E=c|B=a,F=b)\ P(F=b)\ P(B=a|A=a)\ P(A=a)}{P(E=c)} \overbrace{=}^{\text{NR 1–4}}$$
$$\frac{(0.1\cdot 0.5\cdot 0.8\cdot \frac{1}{3})1200}{397} + \frac{(0.1\cdot 0.5\cdot 0.8\cdot \frac{1}{3})1200}{397} = \frac{32}{397}$$
$$P(A=a,B=b|E=c) = \ldots =$$
$$\frac{P(E=c|B=b,F=a)\ P(F=a)\ P(B=b|A=a)\ P(A=a)}{P(E=c)} +$$
$$\frac{P(E=c|B=b,F=b)\ P(F=b)\ P(B=b|A=a)\ P(A=a)}{P(E=c)} \overbrace{=}^{\text{NR 1–4}}$$
$$\frac{(0.6\cdot 0.5\cdot 0.1\cdot \frac{1}{3})1200}{397} + \frac{(0.3\cdot 0.5\cdot 0.1\cdot \frac{1}{3})1200}{397} = \frac{18}{397}$$
$$P(A=a,B=c|E=c) = \ldots =$$
$$\frac{P(E=c|B=c,F=a)\ P(F=a)\ P(B=c|A=a)\ P(A=a)}{P(E=c)} +$$
$$\frac{P(E=c|B=c,F=b)\ P(F=b)\ P(B=c|A=a)\ P(A=a)}{P(E=c)} \overbrace{=}^{\text{NR 1–4}}$$
$$\frac{(0.8\cdot 0.5\cdot 0.1\cdot \frac{1}{3})1200}{397} + \frac{(0.25\cdot 0.5\cdot 0.1\cdot \frac{1}{3})1200}{397} = \frac{21}{397}$$
$$P(A=b,B=a|E=c) = \ldots =$$
$$\frac{P(E=c|B=a,F=a)\ P(F=a)\ P(B=a|A=b)\ P(A=b)}{P(E=c)} +$$
$$\frac{P(E=c|B=a,F=b)\ P(F=b)\ P(B=a|A=b)\ P(A=b)}{P(E=c)} \overbrace{=}^{\text{NR 1–4}}$$
$$\frac{(0.1\cdot 0.5\cdot 0.4\cdot \frac{1}{3})1200}{397} + \frac{(0.1\cdot 0.5\cdot 0.4\cdot \frac{1}{3})1200}{397} = \frac{16}{397}$$
$$P(A=b,B=b|E=c) = \ldots =$$
$$\frac{P(E=c|B=b,F=a)\ P(F=a)\ P(B=b|A=b)\ P(A=b)}{P(E=c)} +$$
$$\frac{P(E=c|B=b,F=b)\ P(F=b)\ P(B=b|A=b)\ P(A=b)}{P(E=c)} \overbrace{=}^{\text{NR 1–4}}$$
$$\frac{(0.6\cdot 0.5\cdot 0.2\cdot \frac{1}{3})1200}{397} + \frac{(0.3\cdot 0.5\cdot 0.2\cdot \frac{1}{3})1200}{397} = \frac{36}{397}$$

$$P(A=b,B=c|E=c) = \ldots =$$
$$\frac{P(E=c|B=c,F=a)\ P(F=a)\ P(B=c|A=b)\ P(A=b)}{P(E=c)} +$$
$$\frac{P(E=c|B=c,F=b)\ P(F=b)\ P(B=c|A=b)\ P(A=b)}{P(E=c)} \overbrace{=}^{\text{NR 1–4}}$$
$$\frac{(0.8\cdot 0.5\cdot 0.4\cdot \frac{1}{3})1200}{397} + \frac{(0.25\cdot 0.5\cdot 0.4\cdot \frac{1}{3})1200}{397} = \frac{84}{397}$$
$$P(A=c,B=a|E=c) = \ldots =$$
$$\frac{P(E=c|B=a,F=a)\ P(F=a)\ P(B=a|A=c)\ P(A=c)}{P(E=c)} +$$
$$\frac{P(E=c|B=a,F=b)\ P(F=b)\ P(B=a|A=c)\ P(A=c)}{P(E=c)} \overbrace{=}^{\text{NR 1–4}}$$
$$\frac{(0.1\cdot 0.5\cdot 0.1\cdot \frac{1}{3})1200}{397} + \frac{(0.1\cdot 0.5\cdot 0.1\cdot \frac{1}{3})1200}{397} = \frac{4}{397}$$
$$P(A=c,B=b|E=c) = \ldots =$$
$$\frac{P(E=c|B=b,F=a)\ P(F=a)\ P(B=b|A=c)\ P(A=c)}{P(E=c)} +$$
$$\frac{P(E=c|B=b,F=b)\ P(F=b)\ P(B=b|A=c)\ P(A=c)}{P(E=c)} \overbrace{=}^{\text{NR 1–4}}$$
$$\frac{(0.6\cdot 0.5\cdot 0.1\cdot \frac{1}{3})1200}{397} + \frac{(0.3\cdot 0.5\cdot 0.1\cdot \frac{1}{3})1200}{397} = \frac{18}{397}$$
$$P(A=c,B=c|E=c) = \ldots =$$
$$\frac{P(E=c|B=c,F=a)\ P(F=a)\ P(B=c|A=c)\ P(A=c)}{P(E=c)} +$$
$$\frac{P(E=c|B=c,F=b)\ P(F=b)\ P(B=c|A=c)\ P(A=c)}{P(E=c)} \overbrace{=}^{\text{NR 1–4}}$$
$$\frac{(0.8\cdot 0.5\cdot 0.8\cdot \frac{1}{3})1200}{397} + \frac{(0.25\cdot 0.5\cdot 0.8\cdot \frac{1}{3})1200}{397} = \frac{168}{397}$$

NR1: $P(E=c) = P(E=c|B=a,F=a)\ P(B=a)\ P(F=a) +$
$P(E=c|B=a,F=b)\ P(B=a)\ P(F=b) +$
$P(E=c|B=b,F=a)\ P(B=b)\ P(F=a) +$
$P(E=c|B=b,F=b)\ P(B=b)\ P(F=b) +$
$P(E=c|B=c,F=a)\ P(B=c)\ P(F=a) +$
$P(E=c|B=c,F=b)\ P(B=c)\ P(F=b)$

NR2: $P(B=a) = P(B=a|A=a)\ P(A=a) +$
$P(B=a|A=b)\ P(A=b) + P(B=a|A=c)\ P(A=c) =$
$0.8\cdot\frac{1}{3} + 0.4\cdot\frac{1}{3} + 0.1\cdot\frac{1}{3} = \frac{13}{30}$

NR3: $P(B=b) = P(B=b|A=a)\ P(A=a) +$
$P(B=b|A=b)\ P(A=b) + P(B=b|A=c)\ P(A=c) =$
$0.1\cdot\frac{1}{3} + 0.2\cdot\frac{1}{3} + 0.1\cdot\frac{1}{3} = \frac{4}{30}$

NR4: $P(B = c) = P(B = c|A = a)\ P(A = a) +$
$P(B = c|A = b)\ P(A = b) + P(B = c|A = c)\ P(A = c) =$
$0.1 \cdot \frac{1}{3} + 0.4 \cdot \frac{1}{3} + 0.8 \cdot \frac{1}{3} = \frac{13}{30}$
$\Rightarrow P(E = c) = 0.1 \cdot \frac{13}{30} \cdot 0.5 + 0.1 \cdot \frac{13}{30} \cdot 0.5 + 0.6 \cdot \frac{4}{30} \cdot 0.5 +$
$0.3 \cdot \frac{4}{30} \cdot 0.5 + 0.8 \cdot \frac{13}{30} \cdot 0.5 + 0.25 \cdot \frac{13}{30} \cdot 0.5 = \frac{397}{1200}$

3.3 Ergebnisse der Simulation

Beim Lernen der Strukter des DAG ergab sich exakt diejenige Struktur, welche dem wahren Graphen zugrunde liegt. Dies ist eine essentielle Voraussetzung, damit die Parameter bzw. die darauf aufbauende Inferenz nahe am wahren Modell sein kann. Bei der Betrachtung der Bootstrapstichprobe, fällt auf, dass der Sparse Candidate Algorithmus mit den verwendeten Einstellungen ziemlich robust ist. Denn die wahren Verbindungen zwischen den einzelnen Knoten sind in jeder Bootstrapstichprobe vorhanden gewesen. Bei diesem Beispiel zeigt sich, dass man mit einem ausreichenden Stichprobenumfang sowie genügend Bootstrapstichproben die wahre, den Daten zugrundeliegende Struktur erkennen kann.

Da in dieser Simulation das Verfahren die wahre, zugrundeliegende Graphenstruktur erkannt hat, bietet sich für den Vergleich der Parameter der RMSE (Root mean squared error) an, denn für jede Zufallsvariable steht die gleiche Anzahl an Parametern zur Verfügung. Der RMSE mit den wahren Parametern θ_j innerhalb eines Knotens ist folgendermaßen definiert: $\sqrt{\frac{1}{p}\sum_{j=1}^{p}\left(\hat{\theta}_j - \theta_j\right)^2}$ Insgesamt gesehen hat das Parameterlernen relativ gut funktioniert. Es ergab sich ein RMSE von 0.0267, d.h. im Mittel unterscheiden sich die geschätzten und wahren Wahrscheinlichkeiten um 2.67. Wenn man sich den RMSE je Zufallsvariable anschaut, zeigt sich, dass die Wahrscheinlichkeitsverteilung der Anfangsknoten genauer geschätzt werden kann als abhängige Knoten:

	A	B	C	D	E	F
RMSE	0.0007	0.0422	0.0054	0.0159	0.0018	0.0159

Tabelle 9: RMSE für Parameterlernen je Zufallsvariable

Bei der Inferenz der drei Wahrscheinlichkeitsverteilungen (Kapitel 3.2) war der RMSE bei allen Fällen unter 0.02 und ist damit praktisch vernachlässigbar. In der Simulation zeigte sich, dass für die bedingte Verteilung $P(D|A = c, C = b)$ nur ca. 15000 Beobachtungen verfügbar waren. Bei den anderen beiden CPQ ergaben sich ca. 65000 Fälle bei denen die Bedingungen zutrafen. Somit kann es bei großen Netzwerken und restriktiven Bedingungen nötig sein eine wesentlich höhere Anzahl an Stichproben zu ziehen, wovon der Großteil nicht mehr verwendet wird. Auf der anderen Seite kann man auf Basis einer Stichprobe viele unterschiedliche CPQ auswerten, ohne neu ziehen zu müssen.

4 Diskussion

In dieser Arbeit wurde ein Überblick zu statischen Bayesnetzen und deren Theorie dargestellt. Bayesnetze sind eine komplexe Methode, welche sich aus etablierten, stochastischen Bausteinen zusammensetzt. Einerseits gibt es eine Vielzahl von Publikationen zu diesem Thema. Andererseits es ist nicht trivial sich für einen konkreten Algorithmus aus der Vielzahl zur Verfügung stehenden zu entscheiden.
Bayesianische Netzwerke haben unter anderem folgende Vorteile [Margaritis, vgl. S. 2]:

- Intuitive, leicht interpretierbare Darstellung als graphisches Modell
- Vereinfachung des Schätzverfahrens durch binning
- Aufdeckung von Strukturen zwischen allen Variablen ohne Vorwissen
- Darstellung gerichteter Zusammenhänge als Indikatoren für kausale Effekte
- Modellierung indirekter Zusammenhänge zwischen Variablen
- Verwendung von etablierter Inferenzmethoden zur Schätzung
- Konzept der lokal, bedingten Unabhängigkeit reduziert Komplexität bei der Inferenz (Vermeidung des Fluchs der Dimensionen)
- Weniger Annahmen als klassische Regression, da man sich nicht auf spezielle Zielgrößen a priori festlegt
- Computational effizient, da Verfahren leicht parallelisierbar sind

Als Nachteile lassen sich folgende Punkte nennen:

- Strukturlernen kritisch für Schätzung der Parameter, Inferenz
- Keine Modellierung von latenten Variablen
- Zyklische Zusammenhänge nicht erfasst
- Keine flexiblen Verteilungen spezifizierbar

Bayesnetze bilden eine sinnvolle Alternative und können unterstützend oder ersatzweise für klassische Regression verwendet werden.
Als Anregung sei noch auf folgendes, englisches Sprichwort verwiesen: *There ain't no such thing as a free lunch.* Mathematisch hat Wolpert sich mit diesem Thema im Bezug zur Optimierung auseinander gesetzt: Bei der Betrachtung aller potenziellen Optimierungsprobleme über alle potenziellen Datensätze kein Algorithmus besser als ein anderer ist [Wolpert and Macready, 1996]. Allerdings kann bei einer spezifischen Fragestellung, d.h. wenn die Problemstellung eingeschränkt ist, durchaus ein Inferenzkonzept wesentlich besser sein als andere. Aber dadurch muss dieses zwangsläufig bei irgend einer anderen Problemstellung Nachteile in Kauf nehmen.

5 Ausblick

In diesem Kapitel wird noch kurz auf erweiterte Konzepte eingegangen, welche über statistische Bayesnetzwerke hinaus gehen. Bisher wurden in dieser Arbeit statische Netzwerke betrachtet, d.h. ohne die Berücksichtigung von Zeitreihen, bzw. longitudinalen Daten. In der Literatur gibt es einige Ansätze die zeitliche Komponente mit im Modell zu berücksichtigen z. B. in [Nagarajan et al., 2013, vgl. Kapitel 3]. Ein Ansatz greift hierzu Methoden der Zeitreihenanalyse auf. Dabei werden 4 wesentliche Annahmen getroffen:

- Es wird unterstellt, dass für den stochastischen Prozess X(t) die Markov Eigenschaft gilt
- Somit folgt, dass alle Ausprägungen von $X(t) = (X_1(t), X_2(t), \dots X_k(t))$ mit t>0 bedingt unabhängig gegeben X (t-1) sind
- Das zeitliche Profil $(X_i(1), \dots X_i(n))$ jeder Variable X_i kann nicht als Linearkombination der andere Profile geschrieben werden
- Der stochastische Prozess ist **homogen** über die Zeit, d.h. die Struktur des Graphen ist invariant gegenüber der Zeit

Mit diesen Annahmen lässt sich ein homogenes, dynamisches Bayesnetz als multivariater, autoregressiver Prozess eindeutig spezifizieren und schätzen. Falls eine große Menge an Daten vorliegt, gibt es Ansätze wie z. B. ARTIVA (Autoregressive Time Varying models) welche ohne die letzte Annahme der Homogenität auskommen. Aufgrund der hohen Komplexität wird auf entsprechende Fachliteratur verwiesen [Lebre et al., 2010].

Eine andere Möglichkeit Bayesnetze zu verallgemeinern liefert das Konzept der unpräzisen Wahrscheinlichkeit. In gewissen Situationen, wie z. B. bei Modellierung von Expertensystemen, ist es schwierig für eine vielseitig abhängigen Zufallsprozess exakte Wahrscheinlichkeiten anzugeben. Beispielsweise hat Ellsberg anhand eines einfachen Urnenmodells demonstriert, dass Unsicherheit viele Dimensionen hat [Ellsberg, 1961]. Die Idee dahinter ist es den klassischen Wahrscheinlichkeitsbegriff auf Intervalle zu verallgemeinern. Es wird eine Menge aus Wahrscheinlichkeitsmaßen anstatt einer einzigen Verteilung angenommen. Ein gängiges Maß zur Bewertung ist eine Untergrenze bzw. Obergrenze einer Wahrscheinlichkeit als Intervall anzugeben. Im Grenzfall ergibt sich ein präzises Einpunktintervall als Spezialfall. Für die Umsetzung auf bayesianische Netzwerke wird auf entsprechende Fachartikel wie z. B. [Corani et al., 2010] verwiesen.

Literatur

Jorgen Bang-Jensen. *Digraphs: Theory, Algorithms and Applications.* Springer-Verlag London Limited, Great Britain, 2010.

Leo Breiman. Bagging predictors. *Kluwer Academic Publishers, University of California, Berkeley*, pages 123–140, 1996.

G. Corani, A. Antonucci, and M. Zaffalon. Bayesian networks with imprecise probabilities: Theory and application to classification. *IDSIA, Switzerland*, pages 1–45, 2010.

Daniel Ellsberg. Risk, ambiguity and savage axioms. *MIT Press: The Quarterly Journal of Economics, California*, pages 643–669, 1961.

Merran Evans, Nicolas Hastings, and Brian Peacock. *Statistical Distributions.* Wiley-Interscience Publication, New York, 2000.

Nir Friedman and Yoram Singer. Efficient bayesian parameter estimation in large discrete domains. *University of California, 387 Soda Hall, Berkeley, CA 94720*, page 7, 1999.

Nir Friedman, Moises Goldszmidt, and AbrahamWyner. Data analysis with bayesian networks: A bootstrap approach. *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pages 1–10, 1999a.

Nir Friedman, Iftach Nachman, and Dana Peer. Learning bayesian network structure from massive datasets. *UAI'99 Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence, Israel*, pages 206–215, 1999b.

L. Gonzalez-Abril, F. Cuberos, F. Velasco, and J. Ortega. Ameva: An autonomous discretization algorithm. *Elsevier, Seville University, 41018 Spain*, pages 5327–5332, 2009.

Finn V. Jensen and Thomas D. Nielsen. *Bayesian Networks and Decision Graphs.* Denmark, 2007.

Keven B. Korb and Ann E. Nicholson. *Bayesian Arfificial Intelligence.* CRC Press Taylor and Francis Group, USA, 2011.

Sotiris Kotsiantis and Dimitris Kanellopoulos. Discretization techniques: A recent survey. *GESTS International Transactions on Computer Science and Engineering, University of Patras, Greece*, pages 47–58, 2006.

Sophie Lebre, Jennifer Becq, Frederic Devaux, Michael PH Stumpf, and Geälle Lelandais. Statistical inference of the time-varying structure of gene-regulation networks. *BMC Systems Biology, UK, France*, pages 1–16, 2010.

Dimitris Margaritis. Learning bayesian network model structure from data.

Radhakrishnan Nagarajan, Marco Scutari, and Sophie Lebre. *Bayesian Networks in R with Applications in Systems Biology.* Springer Science+Business Media, New York, 2013.

Judea Pearl. Bayesian networks: A model of self activated memory for evidential reasoning. *University of California*, page 22, 1985.

Simone Santini. How many dags are there?, 2006.

Marco Scutari and Radhakrishnan Nagarajan. On identifying significant edges in graphical models. *Elsevier, UK, USA*, pages 1–13, 2013.

David Wolpert and William Macready. No free lunch theorems for optimization. *IBM Almaden Research Center, San Jose, Santa Fe*, pages 1–32, 1996.

Eidesstattliche Erklärung

Ich versichere, dass ich beim Anfertigen dieser Arbeit keine Versuche unternommen habe, die (schriftliche) geistige Arbeit anderer unbelegt in meine Arbeit zu übernehmen und als meine eigene auszugeben. Ich versichere darüber hinaus, dass ich meine Arbeit niemandem anderen mit der Absicht zur Verfügung gestellt habe, dass diese/r meine Arbeit oder Teile daraus kopiert (auch in nicht technischen Sinne), um diese als seine/ihre eigene Arbeit/ Leistung auszugeben.

Ort, Datum	Name
München, den 15. Januar 2014	Thomas Welchowski