

Statistische Verfahren in der Künstlichen Intelligenz, Bayes'sche Netze

Erich Schubert

erich.schubert@vitavonni.de

<http://www.vitavonni.de/erich/studies/cs/>

6. Juli 2003

Inhaltsverzeichnis

1	Statistische Verfahren	2
1.1	Formel von Bayes	2
1.2	Bedeutung der Formel von Bayes	3
1.3	Beispiel für die Bayes'sche Formel	3
2	Bayes'sche Netze	4
2.1	Anwendungen für Bayes-Netze:	5
2.2	Komplexität der Anfragen:	5
2.3	Inferenz in Polytrees	5
2.4	Naive Bayes	6
2.5	Leistungsfähigkeit Bayes'scher Netze:	6
3	Lernen von Bayes'schen Netzen	7
3.1	Das EM-Verfahren	7
3.2	Lernen von Strukturen	8
4	Weiterführende Informationen	8

1 Statistische Verfahren

Der Begriff „statistische Verfahren“ wird leicht falsch interpretiert: diese Verfahren machen nicht „automatisch“ mehr Fehler als andere.

So würde kaum jemand erwarten, dass schon bei 23 Leuten die Wahrscheinlichkeit über 50% liegt, dass zwei am selben Tag Geburtstag haben. Das 5-Richter-Problem (siehe Vortrag von Maximilian Hirner) ist ein weiteres Beispiel für einen Fall, wo die Stochastik überraschende Ergebnisse liefert.

Statistik zu verwenden heisst hier vor allem:

1. unvollständige Datenspeicherung (um die Komplexität zu beschränken)
2. „uncertain reasoning“ (Schließen unter Ungewissheit)
3. Fehlerabschätzung
4. Optimalitätsuntersuchung

Angeblich konnten Statistiker im zweiten Weltkrieg die Anzahl der deutschen Panzer aufgrund der Seriennummern von zerstörten Panzern genauer schätzen als die Geheimdienste; beim Knacken von Verschlüsselungen war auch schon immer Statistik ein wesentliches Werkzeug.

Richard P. Feynman (Nobelpreisträger in Physik):

“Chance” is a word which is in common use in everyday living. By chance, we mean something like a guess. Why do we make guesses? We make guesses when we wish to make a judgment but have incomplete information or uncertain knowledge... Sometimes we make guesses because we wish, with our limited knowledge, to say *as much as we can* about some situation... There are good guesses and there are bad guesses. *The theory of probability is a system for making better guesses.* The language of probability allows us to speak quantitatively about some situation which may be highly variable, but which does have some consistent average behavior.

1.1 Formel von Bayes

Eine Formel spielt hier eine besondere Rolle: die *Formel von Bayes*.

Diese ist direkt herleitbar aus der Formel für bedingte Wahrscheinlichkeit: $P(B|A) = \frac{P(A \cap B)}{P(A)}$ ist die Formel für „ B tritt ein, wenn A bereits eingetreten ist“, der Produktregel $P(A \cap B) = P(A) \cdot P(B|A) = P(B) \cdot P(A|B)$ und der disjunkten Zerlegung:

Haben wir nun paarweise disjunkte Ereignisse A_i mit $B \subset \bigcup_i A_i$, so gilt $P(B) = \sum_i P(A_i) \cdot P(B|A_i)$ – man kann B „zerlegen“ in die Fälle der verschiedenen A_i .

Nun betrachten wir $P(A_j|B)$ (j beliebig) – wir wollen also die Rollen von A_j und B vertauschen, die Formel sozusagen umdrehen:

$$P(A_j|B) = \frac{P(B \cap A_j)}{P(B)} = \frac{P(A_j) \cdot P(B|A_j)}{P(B)} = \frac{P(A_j) \cdot P(B|A_j)}{\sum_i P(A_i) \cdot P(B|A_i)} = \alpha P(A_j)P(B|A_j)$$

Dies ist die bekannte „BAYES'SCHE FORMEL“.

α hat hier die Rolle einer „Normierungskonstanten“.

1.2 Bedeutung der Formel von Bayes

Um die Bedeutung der Formel von Bayes zu verstehen kann man sich die A_j und B folgendermaßen vorstellen:

- B ist ein Effekt (Wirkung, Symptom – z.B. „Fieber“),
- A_j sind verschiedene Ursachen (z.B. „Schnupfen“, „Grippe“, „Malaria“)

Dann erlaubt uns die Regel von Bayes bei

- bekannten Abhängigkeiten der Wirkung von den Ursachen – also $P(B|A_j)$
- bekannten Wahrscheinlichkeiten („Häufigkeiten“) der Ursachen – $P(A_j)$

auf die Wahrscheinlichkeit zu schließen, *dass der Effekt von einer bestimmten Ursache ausgelöst wird*. (Hier also z.B. abschätzen, ob das Fieber von einem Schnupfen oder einer Grippe kommt.)

Auf diese Weise können Wahrscheinlichkeiten nicht nur „von oben nach unten“, sondern auch „von unten nach oben“ berechnet werden. Diese Ähnlichkeit zum Vorwärts- und Rückwärtsschließen in der Logik ist nicht ganz zufällig.

1.3 Beispiel für die Bayes'sche Formel

Wir haben in einer E-Mail das Wort „free“ gefunden („Ereignis B “).

Wir wissen, dass das Wort „free“ in $P(B|A_s) = 80\%$ aller unerwünschten und $P(B|A_h) = 5\%$ der erwünschten E-Mails vorkommt, insgesamt in $P(B) = 35\%$ der E-Mails.

$P(A_s) = 40\%$ sind unerwünschte E-Mails, $P(A_h) = 60\%$ sind erwünscht.

Nach der Formel von Bayes gilt nun:

$$P(A_s|B) = \frac{0.40 \cdot 0.80}{0.40 \cdot 0.80 + 0.60 \cdot 0.05} \approx 91.4\%$$

$$P(A_h|B) = \frac{0.60 \cdot 0.05}{0.40 \cdot 0.80 + 0.60 \cdot 0.05} \approx 8.6\%$$

Die E-Mail ist also mit einer Wahrscheinlichkeit von etwa 91.4% Spam.

Man sieht hier auch, dass die „Normierungskonstante“ $1/0.40 \cdot 0.80 + 0.60 \cdot 0.05 = 1/0.35$ dafür sorgt dass sich die beiden Fälle zu 100% ergänzen.

2 Bayes'sche Netze

Die klassische Methode Wahrscheinlichkeiten zwischen Variablen anzugeben wäre, für jede Variable in Abhängigkeit *aller* anderen eine Verteilungsfunktion anzugeben. Bei n Variablen wären also n Funktionen in je $n-1$ Variablen, oder im Falle von diskreten Verteilungen mit 2 Zuständen („Boolsche Variablen“) eine Wahrscheinlichkeitstabelle mit 2^n Einträgen (die sogenannte *full joint probability distribution*). Aus dieser Tabelle können alle Anfragen nach einer Wahrscheinlichkeit beantwortet werden. Diese Werte alle zu speichern, zu ermitteln und zu verwalten ist natürlich sehr ineffizient.

Nun macht man sich zu nutze, dass Variablen unabhängig sein können (und oft sind). Dann ist es nicht notwendig, die vollständige Tabelle zu speichern. Dies führt zu der Datenstruktur des Bayes'schen Netzes.

Man bezeichnet als „Bayes'sches Netz“ (auch: belief network, causal network, probabilistic network, knowledge map) einen *gerichteten, zyklensfreien Graphen* (DAG), wobei für jeden Knoten eine Wahrscheinlichkeitsverteilung in Abhängigkeit seiner Elternknoten (die sogenannte *Conditional Probability Table*, CPT) gegeben ist.

Eine Variable ist nur von ihren Eltern „direkt abhängig“; Knoten zwischen denen es keine (ungerichtet betrachtet) Kanten gibt sind unabhängig. Zwei Knoten mit einem gemeinsamen „Vorfahren“ aber ohne direktere Verbindung sind unabhängig, wenn der Wert des Vorfahrens bekannt ist. (Im folgenden Beispiel sind *JohnCalls* und *MaryCalls* unabhängig, wenn der Wert von *Alarm* bekannt ist)

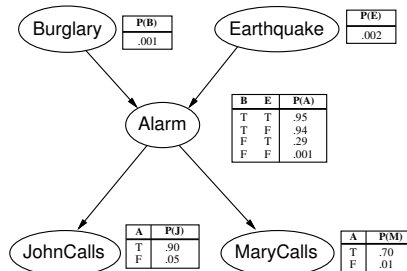


Abbildung 1: Das „Burglary“-Beispiel ist eines der klassischen Beispiele für Bayes'sche-Netze

Mit der Multiplikationsregel können in diesem Netz Kanten „nach unten verfolgt“ werden, mit der Regel von Bayes können umgekehrt Kanten (unter den genannten Voraussetzungen) nach oben verfolgt werden.

Um jetzt die Wahrscheinlichkeit eines bestimmten Zustandes auszurechnen, müssen also nur Werte aus den CPT der Knoten multipliziert werden. So ist hier beispielsweise die Wahrscheinlichkeit dass John und Mary anrufen, weil der Alarm lief aber weder ein Einbruch noch ein Erdbeben war $P(\neg B \wedge \neg E \wedge A \wedge J \wedge M) = P(\neg B) \cdot P(\neg E) \cdot P(A|\neg B \wedge \neg E) \cdot P(J|A) \cdot P(M|A) = 0.999 \cdot 0.998 \cdot 0.001 \cdot 0.90 \cdot 0.70 = 0.00062$.

Um nun eine beliebige Anfrage $P(X|\vec{e})$ beantworten zu können, können alle solchen Möglichkeiten berechnet und summiert werden: $P(X|\vec{e}) = \alpha P(X, \vec{e}) = \alpha \sum_{\vec{y}} P(X, \vec{e}, \vec{y})$. Dies ist der ENUMERATE-JOINT-ASK Algorithmus. Die Laufzeit wächst exponentiell. Mehr dazu im Abschnitt 2.2

2.1 Anwendungen für Bayes-Netze:

Typische Anwendungen für solche Bayes-Netze:

1. *Entscheidungsfindung*: Ermitteln der Wahrscheinlichkeiten bestimmter Ereignisse z.B. für die Wertungsfunktion eines Agenten
2. *Gezielte weitere Abfragen*: Ermitteln, welche zusätzlichen Informationen notwendig sind, eine bestimmte Entscheidung „sicher“ treffen zu können
3. *Sensitivitätsanalyse*: Welche Elemente im Modell haben den größten Einfluss (sollten also möglichst genau gemessen bzw. möglichst präzise durchgeführt werden)
4. *Erklärung von Entscheidungen*: Was ist die wahrscheinlichste Erklärung für das beobachtete Ereignis

2.2 Komplexität der Anfragen:

Ist der Graph ein sogenannter Polytree (single-connected, d.h. es gibt zwischen zwei Knoten maximal einen ungerichteten Weg), so ist die *exakte Inferenz* in linearer Zeit (bzgl. Größe des Netzes) möglich.

Im Allgemeinen ist das Problem aber *nur exponentiell lösbar*. Es ist nicht schwer zu sehen dass sich das SAT-Problem hier abbilden lässt; aber das Problem ist sogar *#P-hard*, also echt schwerer als NP-vollständig („number-P hard“: so schwer wie die Berechnung der *Anzahl* der Lösungen von SAT).

Ein beliebiges Netz kann in exponentieller Zeit in einen solchen Polytree umgewandelt werden. Dabei werden „Schleifen“ zu einem Knoten (mit einer entsprechend größeren CPT!) zusammengezogen.

In der Praxis spielen deswegen approximative Verfahren (Monte-Carlo-Methoden) eine immer größere Rolle, bei denen das Netz sozusagen „simuliert“ wird. Hier werden in jedem Durchlauf die Variablen entsprechend der CPT zufällig belegt und danach die Laplace-Wahrscheinlichkeit berechnet (oder eine Likelihood-Gewichtung vorgenommen).

Auch hier sollte man wieder bedenken dass man ja selten exakte Wahrscheinlichkeiten kennt, eine exakte Inferenz also ebenso Fehler macht.

2.3 Inferenz in Polytrees

Als Beispiel für ein Inferenzverfahren sei hier die Inferenz in Polytrees mittels „message-passing“ beschrieben (nach Pearl). Dies ist auch das Verfahren, das in dem vorgeführten Programm verwendet wurde.

Eine Kante zwischen zwei Knoten kann in zwei Richtungen gelesen werden: Von oben nach unten als „verursachend“ (causal) oder von unten nach oben als „erklärend“ (evidential). Die Wahrscheinlichkeiten werden entlang der Kanten in beide Richtungen weitergegeben und als „causal support message“ bzw. „evidential support message“ bezeichnet.

Aus den „causal support messages“ π_j der Elternknoten kann mit der Wahrscheinlichkeitsfunktion bzw. der CPT die („verursachte“) Wahrscheinlichkeit des Knotens π berechnet werden. Das (punktweise) Produkt der von den Kindern erhaltenen „evidential support messages“ λ_i liefert den „likelihood“ Vektor λ (die Wahrscheinlichkeiten, mit der die Zustände der Kinderknoten erklärbar sind). Das punktweise Produkt dieser beiden Vektoren ist (normalisiert) die gesuchte Wahrscheinlichkeit.

λ' -Nachrichten an den Elternknoten werden mit der Regel von Bayes aus der CPT berechnet (wobei der λ -Vektor als Zustand verwendet wird), π' -Nachrichten an Kinderknoten werden als Produkt der anderen λ_i und π berechnet.

Die festgelegten Knoten („evidence nodes“) stellen dabei Randbedingungen dar (nur der festgelegte Zustand ist möglich und „plausibel“, die anderen nicht), ebenso die Knoten ohne Eltern (Wahrscheinlichkeit π wie in der Tabelle angegeben) sowie die Knoten ohne Kinder ($\lambda = \vec{1}$, alle Zustände sind gleich plausibel).

Hat ein Knoten von allen Eltern „causal support messages“ π_i erhalten und von allen Kindern „evidencial support messages“ λ_j , so kann seine endgültige Wahrscheinlichkeit berechnet werden. Fehlt genau eine dieser Informationen, so kann immernoch eine entgegengesetzte Nachricht berechnet werden, die an den entsprechenden Nachbarknoten weitergegeben wird.

Ein Knoten kann also „Nachrichten“ weitergeben, sobald alle bis auf eine Nachbarkanten in seine Richtung berechnet sind. Damit ist klar, dass dieses Verfahren in linearer Zeit läuft (an jeder Kante wird genau eine Nachricht in jede Richtung weitergegeben) und man sieht auch dass es nur in Polytrees funktioniert. Die Berechnung erfolgt anschaulich „Outside-In“.

2.4 Naive Bayes

„Naive Bayes“-Netze sind Bayes-Netze mit nur einer „Ursache“ und (bei festem Wert der Ursache) unabhängigen „Wirkungen“. Dadurch erhält man eine äußerst einfache Struktur.

Naive Bayes hat einen linearen Speicherbedarf, ist also hervorragend für große Probleme geeignet und gilt als einer der besten Allzweck-Lern-Algorithmen. Er wird in der Praxis sehr häufig verwendet und oft als „Bayesian Classifier“ bezeichnet.

Das Ignorieren eventueller vorhandener Abhängigkeiten verursacht natürlich Fehler (deswegen die Bezeichnung „Idiot Bayes“), macht aber das Verfahren wesentlich einfacher und schneller. Insbesondere sind solche Netze auch sehr einfach zu lernen.

Verwendet wird dies beispielsweise in Spam-Filtern (Mozilla, Spamassassin, bogofilter, ...). Auffallend ist hier, dass diese trotz der offensichtlich bestehenden Abhängigkeiten zwischen den Wortvorkommen noch hervorragende Ergebnisse liefern.

2.5 Leistungsfähigkeit Bayes'scher Netze:

Bayes'sche Netze werden derzeit vor allem als Expertensysteme eingesetzt.

Das Bekannteste ist das PATHFINDER-System zur Diagnose von Lymphknotenerkrankungen. Dieses umfasst 100 Symptome und 14.000 Wahrscheinlichkeiten um 100 Krankheiten zu diagnostizieren. Dieses System ist angeblich mittlerweile besser in der Diagnose als die besten Experten.

Aber: dieses Netz wurde von Experten erstellt und von diesen mit Wahrscheinlichkeitstabellen gefüllt.

— Geht so etwas nicht auch automatisch?

3 Lernen von Bayes'schen Netzen

Beim Lernen von Bayes'schen Netzen muss man im wesentlichen vier Fälle unterscheiden:

Struktur	Beobachtbarkeit	Verfahren
vorgegeben	vollständig	Maximum-Likelihood-Schätzer
vorgegeben	unvollständig	Expectation-Maximization
unbekannt	vollständig	Suchen im Modellraum
unbekannt	unvollständig	EM+Suchen im Modellraum

Bei einer vorgegebenen Struktur müssen letztlich nur die Wahrscheinlichkeitstabellen (CPT: Conditional Probability Table) ausgefüllt werden. Bei vollständiger Beobachtbarkeit können hier alle Werte direkt ausgerechnet werden; im unvollständig beobachtbaren Fall (z.B. ist nicht bekannt ob der Alarm lief, nur dass John angerufen hat, weil er glaubt den Alarm gehört zu haben) müssen die CPTs der „versteckten“ Variablen komplizierter berechnet werden, z.B. mit dem folgenden EM-Verfahren.

3.1 Das EM-Verfahren

Der EM-Algorithmus ist ein allgemeines Verfahren für Lernprobleme.

Zunächst werden die Parameter in irgendeiner Weise vorgelegt („erraten“, z.B. zufällig gewählt).

Danach mit einem aus zwei Schritten bestehenden – dem *expectation step* und dem *maximization step* – Iterationsverfahren bis zu einem lokalen Optimum verbessert.

- Im E-Step werden mittels der aktuellen Parameterwerte die Erwartungswerte für die versteckten Variablen berechnet (also Gewichte berechnet, wie wahrscheinlich die einzelnen Zustände sind),
- Im M-Step werden diese Erwartungswerte als „beobachtet“ angenommen (mit der im E-Step gewonnenen Gewichtung), und die Parameterwerte ermittelt, die *am wahrscheinlichsten dieses Ergebnis liefern*.

Seien \vec{x} die Beobachtungen, \vec{Z} die versteckten Variablen und $\vec{\theta}$ die Parameter. Dann lässt sich der EM-Algorithmus abstrakt beschreiben als

$$\vec{\theta}_{i+1} = \operatorname{argmax}_{\vec{\theta}} \sum_{\vec{z}} \underbrace{P(\vec{Z} = \vec{z} | \vec{x}, \vec{\theta}_i)}_{\text{Gewichtung mit altem Parameter}} \underbrace{L(\vec{x}, \vec{Z} = \vec{z} | \vec{\theta})}_{\text{„Erklärbarkeit“ mit neuem Parameter}}$$

Es wird also diejenigen $\vec{\theta}$ gesucht, dass die möglichen Belegungen der versteckten Variablen ($\vec{Z} = \vec{z}$), gewichtet mit ihrer jeweiligen Plausibilität (Likelihood, $L(\dots)$) mit dem vorherigen Parameter möglichst wahrscheinlich macht ($P(\dots|\vec{\theta}_i)$).

Die Betrachtung der log-likelihood (hier nicht erklärt, aus der Statistik) zeigt, dass das Verfahren gegen ein lokales Minimum konvergiert. Das EM-Verfahren ist ein Hill-Climbing-Algorithmus.

3.2 Lernen von Strukturen

Zum automatischen Lernen von Strukturen gibt es derzeit keine herausragenden Verfahren.

Das Lernen von Strukturen ist sehr aufwendig, es müssen hier Abhängigkeiten untersucht werden (nicht nur statistische Abhängigkeit, sondern Ursache-Wirkungs-Zusammenhänge), versteckte Variablen eingefügt und ein großer Raum von Möglichkeiten durchsucht werden mit geeigneten Wertungsfunktionen. Diese hier darzustellen würde umfangreichere Kenntnisse in der Statistik verlangen und den engen Rahmen des Vortrags sprengen.

Wie stark sich die fehlerhafte Feststellung der Ursache-Wirkungs-Zusammenhänge auf die Struktur des Netzes auswirken kann sieht man in der folgenden Abbildung:

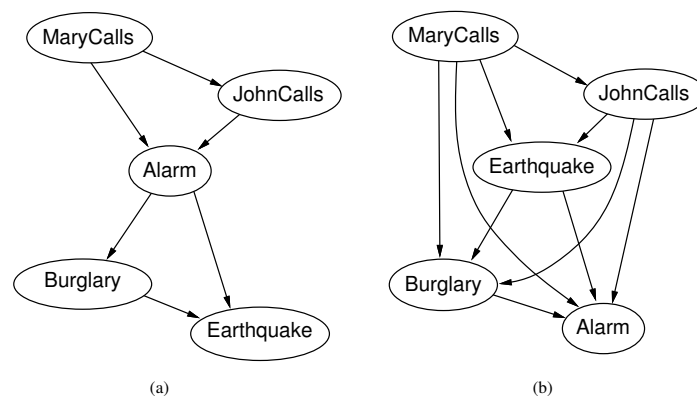


Abbildung 2: Im Fall a) wurden die Knoten in der richtigen Reihenfolge eingefügt, in b) wurden Alarm und Earthquake in der Reihenfolge vertauscht. Dadurch sind kompliziertere CPTs notwendig.

4 Weiterführende Informationen

Als weiterführende Lektüre (insbesondere mit geringen Statistik-Kenntnissen) ist das Tutorial von Heckerman empfohlen.

<http://research.microsoft.com/~heckerman/>

Unterlagen zur Vorlesung „Lectures for Probabilistic Reasoning in AI“ an der McGill Universität Montreal mit detaillierter Beschreibung der Inferenz- und Lernverfahren:

<http://www.cs.mcgill.ca/~dprecup/courses/Winter2002/526/lectures.html>

Unterlagen (Beispielcode etc.) zu diesem Vortrag:

<http://www.vitavonni.de/erich/studies/cs/baynet2003/>