

UseR Bochum/Ruhr

rstan: Bayesianische Modelle in R

Daniel Weller, Ruhr-Universität Bochum

Donnerstag, 11. Mai 2017

Theorie: Bayesianische Statistik

Kurze Suche bei GoogleScholar™ in akad. Publikationen (z.B. Artikel, Reviews, Bänden, etc.) zu Stichwörtern der bayesianischen Statistik:

Ergebnisse bei GoogleScholar™

2000 :~ 23.000//2005 :~ 40.000//2010 :~ 85.000//2015 :~ 97.000

Kurze Suche bei GoogleScholar™ in akad. Publikationen (z.B. Artikel, Reviews, Bänden, etc.) zu Stichwörtern der bayesianischen Statistik:

Ergebnisse bei GoogleScholar™

2000 :~ 23.000//2005 :~ 40.000//2010 :~ 85.000//2015 :~ 97.000

Allerdings finden sich auch einige Beispiele für Erläuterungen und Darstellungen der Bayes-Statistik durch Kunst, Medien und Presse:

Liste von Beispielen

- A Speck in the Sea (The New York Times Magazine, 02.01.2014)
- The Signal and the Noise (N. Silver, Penguin Press, 2012)
- The Theory that would not die (S. B. McGrayne, Yale Press, 2012)
- The Imitation Game (Morten Tyldum, Black Bear Pictures, 2014)
- The Big Bang Theory (Columbia Broadcasting System, 2007)

Häufig wird die Bayes-Statistik in einer Weise dargestellt, die den Eindruck vermittelt, dass es sich um eine neue Methode zur Analyse von Daten handelt.

Häufig wird die Bayes-Statistik in einer Weise dargestellt, die den Eindruck vermittelt, dass es sich um eine neue Methode zur Analyse von Daten handelt.

Aber ist das angebracht? Sinnvoll? Oder nur irreführend?

Häufig wird die Bayes-Statistik in einer Weise dargestellt, die den Eindruck vermittelt, dass es sich um eine **neue Methode zur Analyse von Daten** handelt.

Aber ist das angebracht? Sinnvoll? Oder nur irreführend?

Sicherlich sind gerade durch die starke Verbreitung von Computern sowie die massive **Steigerung von Rechenkapazitäten** seit den 1990er Jahren grundlegende Probleme bei der praktischen Anwendung der bayesianischen Statistik (**Bayesian Computational Problem**) behoben worden,...

Häufig wird die Bayes-Statistik in einer Weise dargestellt, die den Eindruck vermittelt, dass es sich um eine **neue Methode zur Analyse von Daten** handelt.

Aber ist das angebracht? Sinnvoll? Oder nur irreführend?

Sicherlich sind gerade durch die starke Verbreitung von Computern sowie die massive **Steigerung von Rechenkapazitäten** seit den 1990er Jahren grundlegende Probleme bei der praktischen Anwendung der bayesianischen Statistik (**Bayesian Computational Problem**) behoben worden,...

..., jedoch ist die Bayes-Statistik deutlich älter als die **frequentistische Vorgehensweise** und war in der Zeit von **1764 bis 1925** der prägende, wenn auch lediglich schlecht skalierbare Ansatz.

Bayesianische Statistik ist folglich **keine neue Methode, sondern eine andere Perspektive** auf die Art und Weise wie statistische Verfahren und Modelle konzipiert und angewendet werden.

Bayesianische Statistik ist folglich **keine neue Methode, sondern eine andere Perspektive** auf die Art und Weise wie statistische Verfahren und Modelle konzipiert und angewendet werden.

Persönliche Definition

Eine Vorgehensweise zur **Konstruktion und Anwendung** von Modellen bei der man davon ausgeht, dass es diverse Quellen für Information und Unsicherheit gibt.

Bayesianische Statistik ist folglich **keine neue Methode, sondern eine andere Perspektive** auf die Art und Weise wie statistische Verfahren und Modelle konzipiert und angewendet werden.

Persönliche Definition

Eine Vorgehensweise zur **Konstruktion und Anwendung** von Modellen bei der man davon ausgeht, dass es diverse Quellen für Information und Unsicherheit gibt.

Einerseits liegen **Informationen aus Daten**, andererseits manchmal auch **Angaben aus der Umwelt** vor, die durch Verwendung von **Wahrscheinlichkeitsverteilungen** ausgedrückt werden können.

Bayesianische Statistik ist folglich **keine neue Methode, sondern eine andere Perspektive** auf die Art und Weise wie statistische Verfahren und Modelle konzipiert und angewendet werden.

Persönliche Definition

Eine Vorgehensweise zur **Konstruktion und Anwendung** von Modellen bei der man davon ausgeht, dass es diverse Quellen für Information und Unsicherheit gibt.

Einerseits liegen **Informationen aus Daten**, andererseits manchmal auch **Angaben aus der Umwelt** vor, die durch Verwendung von **Wahrscheinlichkeitsverteilungen** ausgedrückt werden können.

Durch die gezielte Verbindung dieser **beiden Quellen** ist es möglich Parameter und Intervalle präzise zu bestimmen.

Bayesianische Statistik ist folglich **keine neue Methode, sondern eine andere Perspektive** auf die Art und Weise wie statistische Verfahren und Modelle konzipiert und angewendet werden.

Persönliche Definition

Eine Vorgehensweise zur **Konstruktion und Anwendung** von Modellen bei der man davon ausgeht, dass es diverse Quellen für Information und Unsicherheit gibt.

Einerseits liegen **Informationen aus Daten**, andererseits manchmal auch **Angaben aus der Umwelt** vor, die durch Verwendung von **Wahrscheinlichkeitsverteilungen** ausgedrückt werden können.

Durch die gezielte Verbindung dieser **beiden Quellen** ist es möglich Parameter und Intervalle präzise zu bestimmen.

Insofern ist es nicht verwunderlich, dass man in Publikationen auch von **probabilistischer Statistik oder Modellierung** spricht.

Die Grundlage bildet die strikte
Anwendung des Mathematikers
und presbyterianischen Pastors
Thomas Bayes (*1701 - †1761).



Abbildung 1: Rev. Thomas Bayes

Die Grundlage bildet die strikte Anwendung des Mathematikers und presbyterianischen Pastors **Thomas Bayes** (*1701 - †1761).

Darin wird eine direkte Beziehung zwischen den konditionalen Wahrscheinlichkeit $\Pr(A|B)$ und $\Pr(B|A)$ herstellt.



Abbildung 1: Rev. Thomas Bayes

Die Grundlage bildet die strikte Anwendung des Mathematikers und presbyterianischen Pastors **Thomas Bayes** (*1701 - †1761).

Darin wird eine direkte Beziehung zwischen den konditionalen Wahrscheinlichkeit $\Pr(A|B)$ und $\Pr(B|A)$ hergestellt.

Insofern erlaubt das Theorem von Bayes eine **Umkehr von Schlussfolgerungen**:



Abbildung 1: Rev. Thomas Bayes

Die Grundlage bildet die strikte Anwendung des Mathematikers und presbyterianischen Pastors **Thomas Bayes** (*1701 - †1761).

Darin wird eine direkte Beziehung zwischen den konditionalen Wahrscheinlichkeit $\Pr(A|B)$ und $\Pr(B|A)$ herstellt.

Insofern erlaubt das Theorem von Bayes eine **Umkehr von Schlussfolgerungen**:

Man geht von einem **vorliegenden Ergebnis** (z.B. **Diagnose**) aus und versucht auf eine **Ursache** (z.B. **Erkrankung**) zu schließen.



Abbildung 1: Rev. Thomas Bayes

Das **Theorem von Bayes** geht von der Formel für konditionale Wahrscheinlichkeit aus und erweitert diese in der folgenden Form:

$$Pr(A \mid B) = \frac{Pr(A \cap B)}{Pr(B)}$$

Das **Theorem von Bayes** geht von der Formel für konditionale Wahrscheinlichkeit aus und erweitert diese in der folgenden Form:

$$Pr(A | B) = \frac{Pr(A \cap B)}{Pr(B)}$$

Wenn keine Angaben über die gemeinsame Wahrscheinlichkeit von A und B vorliegen, lässt sich der **Multiplikationssatz** anwenden. Dieser Satz ergibt sich aus der Umstellung der vorherigen Formel:

$$Pr(A \cap B) = Pr(A | B) * Pr(B)$$

Das **Theorem von Bayes** geht von der Formel für konditionale Wahrscheinlichkeit aus und erweitert diese in der folgenden Form:

$$Pr(A | B) = \frac{Pr(A \cap B)}{Pr(B)}$$

Wenn keine Angaben über die gemeinsame Wahrscheinlichkeit von A und B vorliegen, lässt sich der **Multiplikationssatz** anwenden. Dieser Satz ergibt sich aus der Umstellung der vorherigen Formel:

$$Pr(A \cap B) = Pr(A | B) * Pr(B)$$

Allerdings ist die **Reihenfolge der Ereignisse** bei gemeinsamen Wahrscheinlichkeiten **nicht relevant**, sodass man die Ereignisse vertauschen dürfte:

$$Pr(A \cap B) = Pr(B | A) * Pr(A)$$

Wenn man diese Umstellung in die erste Formel einsetzt, dann erhält man den Sonderfall des Theorems von Bayes:

$$Pr(A | B) = \frac{Pr(B | A) * Pr(A)}{Pr(B)}$$

Wenn man diese Umstellung in die erste Formel einsetzt, dann erhält man den **Sonderfall des Theorems von Bayes**:

$$Pr(A | B) = \frac{Pr(B | A) * Pr(A)}{Pr(B)}$$

Die **Generalisierung des Theorems** erhält man durch die Einbindung des **Satzes der totalen Wahrscheinlichkeit**:

$$P(B) = \sum_{j=1}^N Pr(A_j | B) = \sum_{j=1}^N P(B | A_j) * P(A_j)$$

Wenn man diese Umstellung in die erste Formel einsetzt, dann erhält man den **Sonderfall des Theorems von Bayes**:

$$Pr(A | B) = \frac{Pr(B | A) * Pr(A)}{Pr(B)}$$

Die **Generalisierung des Theorems** erhält man durch die Einbindung des **Satzes der totalen Wahrscheinlichkeit**:

$$P(B) = \sum_{j=1}^N Pr(A_j | B) = \sum_{j=1}^N P(B | A_j) * P(A_j)$$

Folglich gilt für zwei gültige Ereignisse, d.h $Pr(A) > 0$ und $Pr(B) > 0$:

$$Pr(A | B) = \frac{Pr(B | A) * Pr(A)}{\sum_{j=1}^N P(B | A_j) * P(A_j)}$$

Ein Beispiel für die Anwendung des Theorems von Bayes wäre der Versuch festzustellen, wie groß die Wahrscheinlichkeit dafür ist, dass eine Person an Krebs erkrankt (A) ist, wenn lediglich ein positiver Biopsie-Befund (B) vorliegt:

$$Pr(A | B) = \frac{Pr(B | A) * Pr(A)}{Pr(B)}$$

Erläuterung

Die Wahrscheinlichkeit für das **Vorliegen der Tatsache A** (z.B. Wahrscheinlichkeit für eine Krebserkrankung). In der Regel liegt sie vor einem Nachweis (z.B. Diagnose Krebs durch Biopsie) vor und wird daher auch als **Prior-Wahrscheinlichkeit** bezeichnet.

Ein Beispiel für die Anwendung des Theorems von Bayes wäre der Versuch festzustellen, wie groß die Wahrscheinlichkeit dafür ist, dass eine Person an Krebs erkrankt (A) ist, wenn lediglich ein positiver Biopsie-Befund (B) vorliegt:

$$Pr(A | B) = \frac{Pr(B | A) * Pr(A)}{Pr(B)}$$

Erläuterung

Der Ausdruck für den **Nachweis B** (z.B. Diagnose Krebserkrankung durch Biopsie), die eine **Normalisierung** des Produkts im Zähler auf einen gültigen Wertebereich von [0;1] bewirkt.

Ein Beispiel für die Anwendung des Theorems von Bayes wäre der Versuch festzustellen, wie groß die Wahrscheinlichkeit dafür ist, dass eine Person an Krebs erkrankt (A) ist, wenn lediglich ein positiver Biopsie-Befund (B) vorliegt:

$$Pr(A | B) = \frac{Pr(B | A) * Pr(A)}{Pr(B)}$$

Erläuterung

Die konditionale **Wahrscheinlichkeit für den Nachweis B unter der Voraussetzung A** (z.B. die Wahrscheinlichkeit für einen positiven Biopsie-Befund unter der Bedingung, dass eine Krebserkrankung vorliegt). Deshalb trägt diese Wahrscheinlichkeit auch die Bezeichnungen **Likelihood** oder **Evidenz**.

Ein Beispiel für die Anwendung des Theorems von Bayes wäre der Versuch festzustellen, wie groß die Wahrscheinlichkeit dafür ist, dass eine Person an Krebs erkrankt (A) ist, wenn lediglich ein positiver Biopsie-Befund (B) vorliegt:

$$Pr(A | B) = \frac{Pr(B | A) * Pr(A)}{Pr(B)}$$

Erläuterung

Die konditionale Wahrscheinlichkeit für die Tatsache A unter der Bedingung B, z.B. die Wahrscheinlichkeit für eine Krebserkrankung unter der Bedingung, dass ein positiver Biopsie-Befund vorliegt. Aufgrund der Tatsache, dass sie die Wahrscheinlichkeit nach Beobachtung des Nachweises beschreibt, wird sie auch als Posterior-Wahrscheinlichkeit bezeichnet.

Praxis: Bayesianische Modelle

Bisher bezog sich die Anwendung des Theorems von Bayes lediglich auf **diskrete Ereignisse**, aber bereits Pierre-Simon Laplace (*1749 - †1827) zeigte, dass das Theorem auf **Wahrscheinlichkeitsverteilungen** übertragen werden kann:

$$f(\theta | y) = \frac{f(y | \theta) * f(\theta)}{f(y)}$$

Bisher bezog sich die Anwendung des Theorems von Bayes lediglich auf **diskrete Ereignisse**, aber bereits Pierre-Simon Laplace (*1749 - †1827) zeigte, dass das Theorem auf **Wahrscheinlichkeitsverteilungen** übertragen werden kann:

$$f(\theta | y) = \frac{f(y | \theta) * f(\theta)}{f(y)}$$

Dieser Schritt stellte eine **wichtige Voraussetzung** für alle Aspekte der statistischen Modellierung dar, führte jedoch auch dazu, dass **komplexe Integrale** nötig werden, um Berechnungen zu realisieren (**Bayesian Computation Problem**).

Insbesondere bei **komplexen Modellen** - und dazu zählen bereits Regressionen - wird eine **direkte Analyse der Posterior-Verteilung** sehr schwer, wodurch andere Methoden genutzt werden müssen.

Dabei gibt es eine Unterscheidung zwischen approximativen Methoden und simulativen Methoden:

Approximative Methoden

Diese Methoden bieten schnelle Prozeduren zur Approximation der Posterior-Verteilung (Tierney & Kadane 1986; Rue & Martino 2009). Allerdings bestehen aktuell noch Schwierigkeiten bei komplexeren Verfahren (z.B. Multilevel- oder Mixture-Modelle).

Insbesondere bei **komplexen Modellen** - und dazu zählen bereits Regressionen - wird eine **direkte Analyse der Posterior-Verteilung** sehr schwer, wodurch andere Methoden genutzt werden müssen.

Dabei gibt es eine Unterscheidung zwischen approximativen Methoden und simulativen Methoden:

Simulative Methoden

Zur Simulation der Posterior-Verteilung wird bei diesen Methoden auf den Einsatz von Markov-Ketten gesetzt, wobei unterschiedliche Sampling-Verfahren (z.B. Metropolis-Hastings (Metropolis et. al 1953), Gibbs (Geman & Geman 1984), Adaptive Rejection (Gilks & Wild 1995) oder Slice (Neal 1997)) zur Generierung von Stichproben verwendet werden.

Stan ist eine neue C++-Bibliothek, die einen nächsten Schritt in der Entwicklung bayesianischer Inferenz (Carpenter et. al 2016) darstellt.



Stan ist eine neue C++-Bibliothek, die einen nächsten Schritt in der Entwicklung bayesianischer Inferenz (Carpenter et. al 2016) darstellt.

Es folgt der gleichen **Entkopplung von Modellierung und Schätzung**, die bereits seine „Verwandten“ BUGS (Lunn et al. 2000) und JAGS (Plummer 2003) populär gemacht hat:



Stan ist eine neue C++-Bibliothek, die einen nächsten Schritt in der Entwicklung bayesianischer Inferenz (Carpenter et. al 2016) darstellt.

Es folgt der gleichen **Entkopplung von Modellierung und Schätzung**, die bereits seine „Verwandten“ BUGS (Lunn et al. 2000) und JAGS (Plummer 2003) populär gemacht hat:

Modellierung $\Rightarrow$ Exportierung $\Rightarrow$ Kompilierung $\Rightarrow$ Auswertung



Stan ist eine neue C++-Bibliothek, die einen nächsten Schritt in der Entwicklung bayesianischer Inferenz (Carpenter et. al 2016) darstellt.

Es folgt der gleichen Entkopplung von Modellierung und Schätzung, die bereits seine „Verwandten“ BUGS (Lunn et al. 2000) und JAGS (Plummer 2003) populär gemacht hat:

Modellierung $\Rightarrow$ Exportierung $\Rightarrow$ Kompilierung $\Rightarrow$ Auswertung

Dabei setzt der Compiler eine Imperativ-Syntax zur Modellierung ein, die sich stark an der formalen Schreibweise der Statistik ausrichtet (z.B. Matrizen- und Vektoren-Notation für Modellgleichungen, etc.) und direkt aus externen Analysesoftwaren abgeschickt werden kann.



Ein Vorteil von Stan besteht darin, dass es eine Reihe von algorithmischen Regulierungs- und Optimierungsprozeduren bereitstellt, die deutliche Zuwächse an Effizienz und Effektivität bei Markov-Ketten ermöglichen.



Ein Vorteil von Stan besteht darin, dass es eine Reihe von algorithmischen Regulierungs- und Optimierungsprozeduren bereitstellt, die deutliche Zuwächse an Effizienz und Effektivität bei Markov-Ketten ermöglichen.

Außerdem bietet der Compiler eine Prozedur zur Parallelisierung von Ketten, wodurch die Rechenleistung von Mehrkernprozessoren effizienter genutzt werden kann.



Ein Vorteil von Stan besteht darin, dass es eine Reihe von algorithmischen Regulierungs- und Optimierungsprozeduren bereitstellt, die deutliche Zuwächse an Effizienz und Effektivität bei Markov-Ketten ermöglichen.

Außerdem bietet der Compiler eine Prozedur zur Parallelisierung von Ketten, wodurch die Rechenleistung von Mehrkernprozessoren effizienter genutzt werden kann.

Im Prinzip ist Stan durch das No-U-Turn-Sampling (Hoffmann & Gelman 2014), das eine adaptive Variation des Hamiltonian Monte Carlo darstellt, populär geworden.



Ein Vorteil von Stan besteht darin, dass es eine Reihe von algorithmischen Regulierungs- und Optimierungsprozeduren bereitstellt, die deutliche Zuwächse an Effizienz und Effektivität bei Markov-Ketten ermöglichen.

Außerdem bietet der Compiler eine Prozedur zur Parallelisierung von Ketten, wodurch die Rechenleistung von Mehrkernprozessoren effizienter genutzt werden kann.

Im Prinzip ist Stan durch das No-U-Turn-Sampling (Hoffmann & Gelman 2014), das eine adaptive Variation des Hamiltonian Monte Carlo darstellt, populär geworden.

Dadurch ist Stan bei gleicher Kettenlänge deutlich effizienter, allerdings auch langsamer pro Iteration als BUGS oder JAGS.



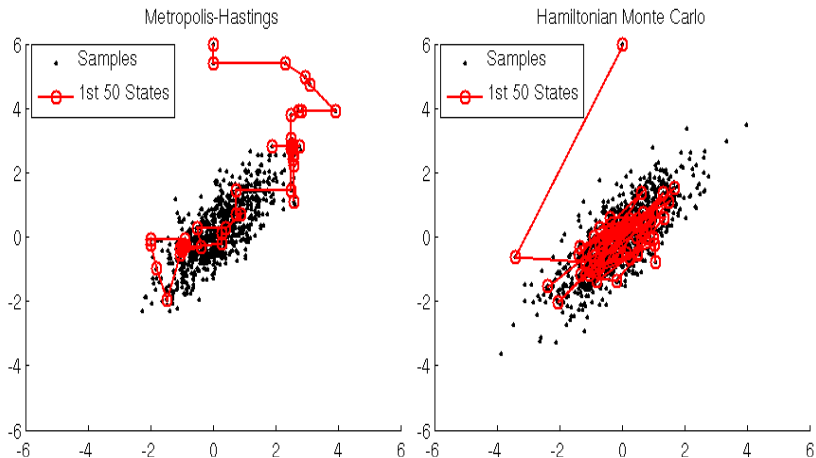


Abbildung 2: The Clever Machine Blog by Dustin Stansbury (2012)

In der Regel besteht ein Modell aus 6 Blöcken, die spezielle Funktionen bei der Kompilierung und Berechnung übernehmen. Allerdings sind nur drei Blöcke verpflichtend:

In der Regel besteht ein Modell aus 6 Blöcken, die spezielle Funktionen bei der Kompilierung und Berechnung übernehmen. Allerdings sind nur drei Blöcke verpflichtend:

data

Zur Berechnung von Modellen werden Werte benötigt, die an dieser Stelle deklariert werden müssen. Dabei haben die Namen den Elementen im Speicher der Analysesoftware zu entsprechen.

In der Regel besteht ein Modell aus 6 Blöcken, die spezielle Funktionen bei der Kompilierung und Berechnung übernehmen. Allerdings sind nur drei Blöcke verpflichtend:

data

Zur Berechnung von Modellen werden Werte benötigt, die an dieser Stelle deklariert werden müssen. Dabei haben die Namen den Elementen im Speicher der Analysesoftware zu entsprechen.

parameters

Dieser Block gibt alle Arten von Parametern an, die ein Modell benötigt. Dies schließt auch Nuisance-Werte und Constraints ein.

In der Regel besteht ein Modell aus 6 Blöcken, die spezielle Funktionen bei der Kompilierung und Berechnung übernehmen. Allerdings sind nur drei Blöcke verpflichtend:

data

Zur Berechnung von Modellen werden Werte benötigt, die an dieser Stelle deklariert werden müssen. Dabei haben die Namen den Elementen im Speicher der Analysesoftware zu entsprechen.

parameters

Dieser Block gibt alle Arten von Parametern an, die ein Modell benötigt. Dies schließt auch Nuisance-Werte und Constraints ein.

model

Zur Berechnung der Log-Posterior-Verteilung wird in diesem Block neben der Likelihood auch die Gruppe der Prior-Verteilungen angegeben.

Sobald diese Blöcke in einem Modell vorliegen, kann eine Berechnung ausgeführt werden. Allerdings kann ein Nutzer die folgenden Blöcke nutzen, um eine deutliche Steigerung der Effizienz zu erreichen:

Sobald diese Blöcke in einem Modell vorliegen, kann eine Berechnung ausgeführt werden. Allerdings kann ein Nutzer die folgenden Blöcke nutzen, um eine deutliche Steigerung der Effizienz zu erreichen:

transformed data

Wenn Daten zur Berechnung des Modells vor der Berechnung vorprozessiert werden sollen, dann erlaubt dieser Block die Anwendung von funktionalen Beziehungen anzugeben.

Sobald diese Blöcke in einem Modell vorliegen, kann eine Berechnung ausgeführt werden. Allerdings kann ein Nutzer die folgenden Blöcke nutzen, um eine deutliche Steigerung der Effizienz zu erreichen:

transformed data

Wenn Daten zur Berechnung des Modells vor der Berechnung vorprozessiert werden sollen, dann erlaubt dieser Block die Anwendung von funktionalen Beziehungen anzugeben.

transformed parameters

Analog dazu, gibt es diesen Block für die Transformierung von Parametern. Dies ist insbesondere für effiziente Modelle relevant.

Sobald diese Blöcke in einem Modell vorliegen, kann eine Berechnung ausgeführt werden. Allerdings kann ein Nutzer die folgenden Blöcke nutzen, um eine deutliche Steigerung der Effizienz zu erreichen:

transformed data

Wenn Daten zur Berechnung des Modells vor der Berechnung vorprozessiert werden sollen, dann erlaubt dieser Block die Anwendung von funktionalen Beziehungen anzugeben.

transformed parameters

Analog dazu, gibt es diesen Block für die Transformierung von Parametern. Dies ist insbesondere für effiziente Modelle relevant.

generated quantities

Im Anschluss an das Modell bündeln sich an dieser Stelle alle Angaben zur Nachbearbeitung, wie z.B. Predicted Values.

```

data {int N;
      int y[N];
      vector[N] x1;
      vector[N] x2;}
transformed data {vector[N] x1_r;
                  x1_r <- x1/10.0;}
parameters {real alpha;
            real sigma;
            vector[2] beta;}
model{alpha~normal(0,10);
      beta[1]~normal(0,10);
      beta[2]~normal(0,10);
      mu = alpha + beta[1]*x1_r + beta[2]*x2;
      y ~ bernoulli_logit(mu);}
generated quantities {vector[N] log_lik;
                     for (n in 1:N) {
                       log_lik[n] = bernoulli_logit_lpdf(y[n],mu);}}

```

```

data {int<lower=0,upper=500> N;
      int<lower=1,upper=2> I;
      int<lower=0,upper=1> y[N];
      matrix[N,I] X;}
transformed data {X[,1] <- X[,1]/10.0;}
parameters {real<lower=-5,upper=5> alpha;
             vector[I] beta;
             real<lower=0> sigma;}
model {alpha~cauchy(0,5);
        beta~cauchy(0,2.5);
        mu = alpha + X*beta
        y ~ bernoulli_logit(mu);}
generated quantities {vector[N] log_lik;
                      for (n in 1:N) {
                        log_lik[n] = bernoulli_logit_lpdf(y[n],mu);}}

```

Es gibt eine große Sammlung von Funktionen, die das Stan Development Team bereitstellt, um Modelle durch Zugriff auf die Stan-Library zu strukturieren, kompilieren, berechnen und analysieren.

Es gibt eine große Sammlung von Funktionen, die das Stan Development Team bereitstellt, um Modelle durch Zugriff auf die Stan-Library zu strukturieren, kompilieren, berechnen und analysieren.

Neben dem Zugpferd Hamiltonian Markov Chain Monte Carlo bietet das Paket auch Approximationen über Variational Bayes-Prozeduren sowie Penalized Maximum Likelihood.

Es gibt eine große Sammlung von Funktionen, die das **Stan Development Team** bereitstellt, um Modelle durch Zugriff auf die Stan-Library zu strukturieren, kompilieren, berechnen und analysieren.

Neben dem Zugpferd **Hamiltonian Markov Chain Monte Carlo** bietet das Paket auch Approximationen über **Variational Bayes-Prozeduren** sowie **Penalized Maximum Likelihood**.

In jedem Fall werden automatische Differenzierungen genutzt, um schnell und zuverlässig Gradienten zu evaluieren.

Es gibt eine große Sammlung von Funktionen, die das **Stan Development Team** bereitstellt, um Modelle durch Zugriff auf die Stan-Library zu strukturieren, kompilieren, berechnen und analysieren.

Neben dem Zugpferd **Hamiltonian Markov Chain Monte Carlo** bietet das Paket auch Approximationen über **Variational Bayes-Prozeduren** sowie **Penalized Maximum Likelihood**.

In jedem Fall werden automatische Differenzierungen genutzt, um schnell und zuverlässig Gradienten zu evaluieren.

Maintainer: Ben Goodrich, Columbia University

Akt. Version: 2.14.2

Abhängigkeit: R ($\geq 3.0.2$), ggplot2 ($\geq 2.2.1$), StanHeaders ($\geq 2.14.0$)

Publikation: 2017-03-19

Ergänzungen: bayesplot, brms, coda, MCMCvis, rstantools, rstudioapi

```
stan(file, model_code= ..., model_name= ..., data= list(),  
     init= ..., algorithm= ..., pars= ..., chains= ...,  
     warmup= ..., iter= ..., thin= ..., ...)
```

file	= String-Vektor für Modell-Datei oder Objektnamen
model_code	= String-Vektor für Modell-Datei oder Connection
model_name	= String-Vektor für den Namen des Modells
data	= Werteliste oder Workspace-Objekt der Modell-Daten
init	= Startwerte der Koeffizienten (0, Random oder Liste)
algorithm	= Auswahl des Sampling (NUTS, HMC oder Fixed)
pars	= String-Vektor von relevanten Sampling-Parametern
chains	= Anzahl der Markov-Ketten
warmup	= Adaptions- und Warmup-Länge der Markov-Ketten
iter	= Gesamtlänge der Markov-Ketten
thin	= Länge des Speicher-Lags von Samples

```
library(rstan)

# Top-Level Optimierung:
options(mc.cores=parallel::detectCores())
rstan_options(boost_lib= "Path_Boost", eigen_lib= "Path_Eigen")

# Low-Level Optimierung:
rstan_options(auto_write= TRUE)
options(scipen=10)
```

Generell ist eine **Top-Level Optimierung** zu empfehlen, da interne Prozeduren von Stan deutlich beschleunigt werden (z.B. **Parallelisierung** und **Maximalauslastung** des Systems; Registrierung von C++-Bibliotheken).

```
library(rstan)

# Top-Level Optimierung:
options(mc.cores=parallel::detectCores())
rstan_options(boost_lib= "Path_Boost", eigen_lib= "Path_Eigen")

# Low-Level Optimierung:
rstan_options(auto_write= TRUE)
options(scipen=10)
```

Generell ist eine **Top-Level Optimierung** zu empfehlen, da interne Prozeduren von Stan deutlich beschleunigt werden (z.B. **Parallelisierung** und **Maximalauslastung** des Systems; Registrierung von C++-Bibliotheken).

Eine **Low-Level Optimierung** reduziert lediglich **potentielle Speicherlasten durch Auslagerung** von Workspace-Objekten auf die Festplatte und Präzisierung von Ausgaben.

In diesem Beispiel werden Daten aus dem **Television School and Family Smoking Prevention and Cessation Project** von Fray et. al (1998) verwendet.

Es soll eine logistische Regression von bestimmten Aspekten des Schulunterrichts zum Thema „Tabak und Gesundheit“ auf das Ergebnis eines Post-Intervention-Test berechnet werden.

```
THKS <- read.csv("thks.csv")
THKS <- subset(THKS, select=c("THKSbin", "PreTHKS",
                             "CC", "TV", "CC.TV"))

Y <- THKS[,1]
X <- THKS[,2:5]
N <- nrow(X)
D <- ncol(X)
Input.Data <- list(Y=Y, X=X, N=N, D=D)
```

```

Logit_Model= "
data {int<lower=0,upper=500> N;
      int<lower=1,upper=4> I;
      int<lower=0,upper=1> y[N];
      matrix[N,I] X;}
transformed data {X[,1] <- X[,1]/10.0;}
parameters {real<lower=-5,upper=5> alpha;
            vector[I] beta;
            real<lower=0> sigma;}
model {alpha~cauchy(0,5);
       beta~cauchy(0,2.5);
       mu = alpha + X*beta
       y ~ bernoulli_logit(mu);}
generated quantities {vector[N] log_lik;
                     for (n in 1:N) {
                         log_lik[n] = bernoulli_logit_lpdf(y[n],mu);}}
writeLines(Logit_Model, con= "Logit_Model.stan")
parameters <- c("alpha", "beta")

```

```
Result_Logit <- stan(model_code= Logit_Model.stan",  
                     model_name= "Logit Model Example",  
                     data= Input.Data, algorithm= "NUTS",  
                     init= "random", pars= parameters,  
                     chains= 5, iter= 5000, warmup= 4000)
```

Value	Mean	SD	2.5%	50%	97.5%	N.Eff	Rhat
Intercept	-1.22	0.14	-1.50	-1.22	-0.94	3116	1
PreTHKS	0.40	0.04	0.32	0.40	0.49	3716	1
CC	0.96	0.15	0.68	0.96	1.26	3158	1
TV	0.31	0.14	0.03	0.31	0.59	3511	1
CCTV	-0.40	0.21	-0.80	-0.40	0.01	3226	1

Trace- und Density-Plots

```
stan_trace(stan_object)  
stan_hist(stan_object)
```

Coefficient- und Scatterplots

```
stan_plot(stan_object)  
stan_scat(stan_object, pars=c("beta[1]", "beta[2]"))
```

Modell-Diagnosen

```
stan_diag(stan_object)
```

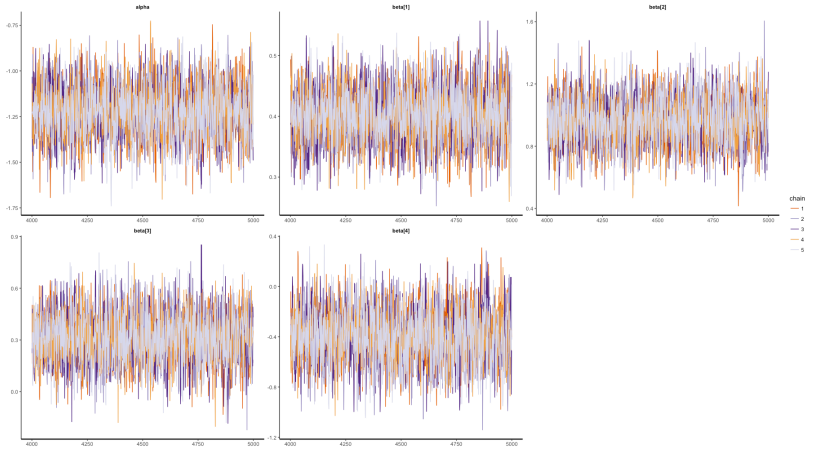


Abbildung 3: Traceplot

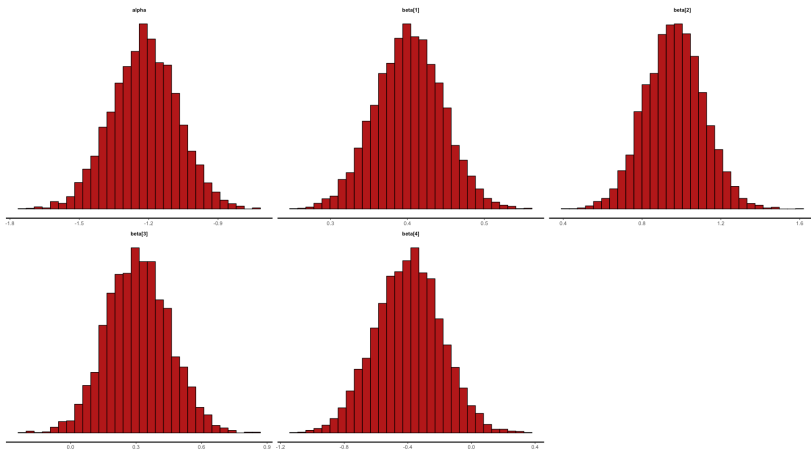


Abbildung 4: Density Plot

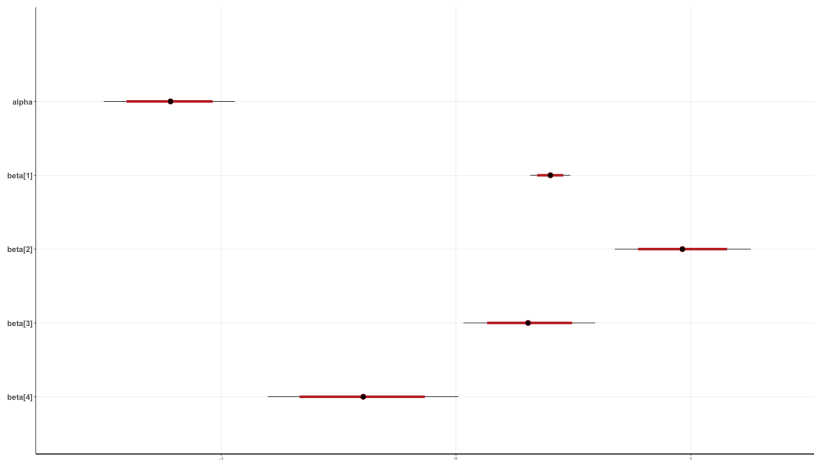


Abbildung 5: Caterpillar Plot

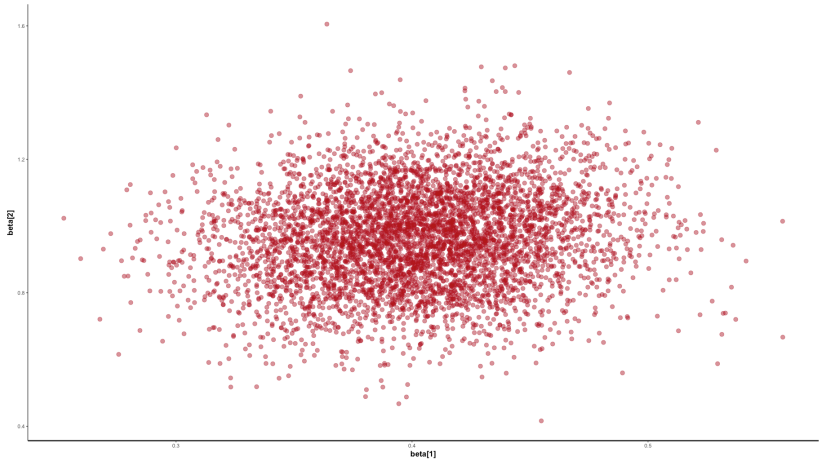


Abbildung 6: Scatterplot

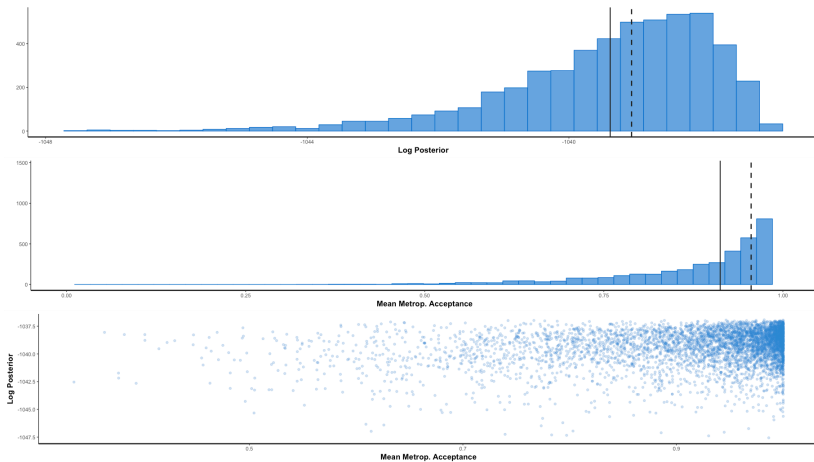
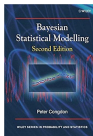
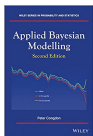


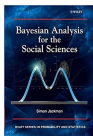
Abbildung 7: Modell-Diagnose



Peter Congdon (2006):
Bayesian Statistical Modelling
John Wiley & Sons



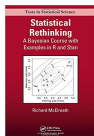
Peter Congdon (2014):
Applied Bayesian Modelling
John Wiley & Sons



Simon Jackman (2009):
Bayesian Analysis for the Social Sciences
John Wiley & Sons



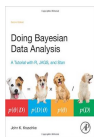
Andrew Gelman, John Carlin, Donald Rubin, u.a. (2013):
Bayesian Data Analysis
Chapman & Hall/CRC Press



Richard McElreath (2016):

Statistical Rethinking: A Bayesian Course with Examples in R and Stan

Chapman & Hall/CRC Press



John Kruschke (2014):

Doing Bayesian Data Analysis: A Tutorial with R, JAGS and Stan

Academic Press



Scott Lynch (2007):

Introduction to Applied Bayesian Statistics and Estimation for Social Scientists

Springer Press

Literatur



Bayes, T. und R. Price (1764): „An Essay towards Solving a Problem in the Doctrine of Chances. By the Late Rev. Mr. Bayes, FRS Communicated by Mr. Price, in a Letter to John Canton, AMFRS“. In: *Philosophical Transactions* 53, S. 370–418.



Carpenter, B. u. a. (2017): „Stan: A probabilistic programming language“. In: *Journal of Statistical Software* 76.1, S. 1–32.



Flay, B. u. a. (1995): „The Television, School, and Family Smoking Prevention and Cessation Project“. In: *PREVENTIVE MEDICINE* 24, S. 29–40.



Geman, S. und D. Geman (1984): „Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 6, S. 721–741.



Gilks, W. und P. Wild (1995): „Adaptive rejection Metropolis sampling within Gibbs sampling“. In: *Applied Statistics*, S. 455–472.

-  Hastings, K. (1970): „Monte Carlo sampling methods using Markov chains and their applications“. In: *Biometrika* 57.1, S. 97–109.
-  Hoffman, M. und A. Gelman (2014): „The No-U-Turn Sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo“. In: *Journal of Machine Learning Research* 15.1, S. 1593–1623.
-  Lunn, D. u. a. (2000): „WinBUGS - A Bayesian modelling framework: Concepts, structure, and extensibility“. In: *Statistics and Computing* 10.4, S. 325–337.
-  Metropolis, N., A. Rosenbluth u. a. (1953): „Equation of state calculations by fast computing machines“. In: *The Journal of Chemical Physics* 21.6, S. 1087–1092.
-  Metropolis, N. und S. Ulam (1949): „The Monte Carlo Method“. In: *Journal of the American statistical association* 44.247, S. 335–341.
-  Neal, R. (1997): „Markov Chain Monte Carlo methods based on slicing the density function“. Diss.
-  – (2011): „MCMC using Hamiltonian Dynamics“. In: *Handbook of Markov Chain Monte Carlo*. Hrsg. von S. Brooks u. a.



Plummer, M. (2003): „JAGS: A program for analysis of Bayesian graphical models using Gibbs sampling“. In: *Proceedings of the 3rd International Workshop on Distributed Statistical Computing*.



R Development Core Team (2017): *R: A Language and Environment for Statistical Computing*. Software Version 3.3.2. URL: <http://www.R-project.org>.



Stan Development Team (2016): *RStan: The R interface to Stan*. R Package Version 2.14.1. URL: <http://mc-stan.org/>.



– (2017): *Stan Modeling Language Users Guide and Reference Manual*. Software Version 2.14.0. URL: <http://mc-stan.org/>.