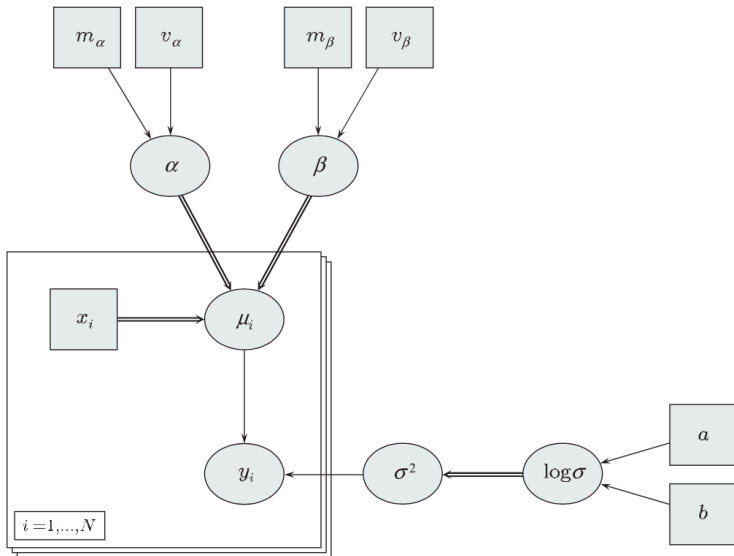


# JAGS, Convergence, STAN

Volker Schmid

# BUGS

# Directed Acyclic Graphs



# Bayesian inference using Gibbs sampling (BUGS)

- ▶ Idee: Ziehen aus der Grundlage der vollständigen bedingten Verteilung eines Knotens, diese hängen nur von den Parameter/Daten ab, die durch Kanten verbunden sind (Markov blanket)
- ▶ Erste Versuche in den frühen 1980er Jahren
- ▶ Erste BUGS-Implementierung 1989
- ▶ WinBUGS Mitte der 1990er Jahre
- ▶ Selbstoptimierender Metropolis-Updater in PKBugs
- ▶ OpenBUGS im Jahr 2004
- ▶ Just Another Gibbs Sampler (JAGS): Klon in C++

# JAGS distributions

Distribution	Distribution	Quantile	
Bernoulli	dbern	pbern	qbern
Beta	dbeta	pbeta	qbeta
Binomial	dbin	pbin	qbin
Chi-squared	dchisqr	pchisqr	qchisqr
Double exponential	ddexp	pdexp	qdexp
Exponential	dexp	pexp	qexp
F	df	pf	qf
Gamma	dgamma	pgamma	qgamma
Generalized gamma	dgen.gamma	pgen.gamma	qgen.gamma
Noncentral hypergeometric	dhyper	phyper	qhyper
Logistic	dlogis	plogis	qlogis
Log-normal	dlnorm	plnorm	qlnorm
Negative binomial	dnegbin	pnegbin	qnegbin
Noncentral Chi-squared	dncchisqr	pnchisqr	qnchisqr
Normal	dnorm	pnorm	qnorm
Pareto	dpar	ppar	qpar
Poisson	dpois	ppois	qpois
Student t	dt	pt	qt
Weibull	dweib	pweib	qweib

Table 6.2: Functions to calculate the probability density, probability function, and quantiles of some of the distributions provided by the `bugs` module.

# Konvergenz-Diagnostik

# Gelman Rubin I

- ▶ Idee: Vergleich der Varianz mehrerer parallel laufender Ketten (alternativ Aufteilung einer Kette in mehrere Teile)
- ▶ Bei Konvergenz sollte die Varianz in den Ketten der Varianz zwischen den Ketten entsprechen
- ▶ Varianz innerhalb der Kette (Within Chain Variance) unterschätzt die wahre Varianz, wenn die Ketten noch nicht konvergiert haben:

$$W = \frac{1}{m} \sum_{j=1}^m s_j^2, s_j^2 = \frac{1}{n-1} \sum_{i=1}^n (\theta_{ij} - \bar{\theta}_j)^2$$

- ▶ Varianz zwischen den Ketten (Between Chain Variance)

$$B = \frac{n}{m-1} \sum_{j=1}^m (\bar{\theta}_j - \bar{\bar{\theta}})^2; \bar{\bar{\theta}} = \frac{1}{m} \sum_{j=1}^m \bar{\theta}_j$$

- ▶ Geschätzte Variance

$$\hat{Var}(\theta) = \left(1 - \frac{1}{n}W + \frac{1}{n}B\right)$$

- ▶ Potential scale reduction factor (psrf) für jeden (!) Parameter

$$\hat{R} = \sqrt{\frac{\hat{Var}(\theta)}{W}}$$

- ▶ Großes  $\hat{R} \rightsquigarrow$  mehr Iterationen nötig
- ▶ Faustformel:  $R < 1.1$
- ▶ Multivariate Version

- ▶ Idee: Test, ob zwei Teile einer Kette die gleiche Verteilung (Mittelwert) haben
- ▶ In der Regel: Erste 10% gegen letzte 50%
- ▶ coda-Paket: `geweke.diag()` liefert Z-Score

# Raftery and Lewis

- ▶ Wir wollen das  $q$ -Quantil der Posteriori schätzen.
- ▶ Wie viele Iterationen  $N$  und welchen burn-in  $M$  brauchen wir, um das  $q$ -quantile mit Wahrscheinlichkeit  $s$  mit Toleranz  $r$  schätzen zu können.
- ▶ Kurze Kette, um die Länge der notwendigen Kette abzuschätzen.

# Heidelberg and Welch

- ▶ Test auf Stationarität der Kette
- ▶ Erster Schritt: Cramer-von-Mises-Test mit Niveau  $\alpha$  auf ganze Kette
- ▶ Falls Nullhypothese abgelehnt, verwirfe erste 10%, 20%, ..., bis zu 50%
- ▶ Falls Nullhypothese nicht abgelehnt: Half-width-test

# Hamiltonian Monte Carlo

# Hamiltonian mechanics

(Radon-Nikodým-)Dichte  $f$  entspricht der physikalischen (Gibbs-)Energie  $\phi$ :

$$p(x) \propto \exp(-\phi(x))$$

**Hamiltonian mechanics:** Die Bewegung eines Objekts durch einen Raum kann durch Differentialgleichungen mit Position  $x$  (potentielle Energie  $U(x)$ ) und Impuls  $v$  (kinetische Energie  $K(v)$ ) beschrieben werden

$$E(x, v) = U(x) + K(v)$$

# Hamiltonian MC (Hybrid MC)

- ▶ Hilfsvariable  $v$ , aus deren Priori (Normalverteilung) gezogen wird
- ▶ Veranschaulichung: Bewegung eines Hockey-Pucks auf der Parameterfläche, der wiederholt in zufällige Richtungen geschoben wird.
- ▶ Für MCMC müssen die Hamilton-Gleichungen zeitlich diskretisiert werden, mit Schrittweite  $\epsilon$  (einfach: Euler-Methode, besser: Leapfrog).

# Euler/Leapfrog

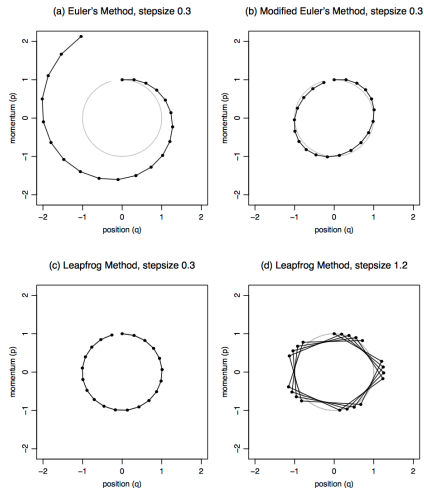


Figure 3: 20 iterations Euler vs. Leapfrog

# NUTS

- ▶ Der No-U-Turn Sampler (NUTS) wählt automatisch eine angemessene Anzahl von Sprungschritten in jeder Iteration aus, um den Vorschlägen zu ermöglichen, den Posterior ohne unnötige Arbeit zu durchlaufen.
- ▶ Maximierung der erwarteten quadratischen Sprungdistanz bei jedem Schritt
- ▶ Vermeidung des „Random-Walk“-Verhaltens, das auftritt, wenn es eine Korrelation im Posterior gibt

# Vorteile und Nachteile von HMC

- ▶ Funktioniert nur mit differenzierbaren Funktionen
- ▶ Individuelle Iteration ist viel komplexer als Metropolis-Hasting oder Gibbs
- ▶ Generell geringere Abhängigkeit, großer Abstand zwischen den Ziehungen
- ▶ Hohe Akzeptanzrate
- ▶ Probleme mit isolierten lokalen Minima (oder Maxima)
- ▶ Abstimmung ist notwendig ( $\epsilon$ )
- ▶ STAN in Entwicklung

Radford M. Neal: MCMC using Hamiltonian dynamics, in: Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng. Handbook of Markov Chain Monte Carlo. Chapman & Hall / CRC Press, 2011

# STAN (RStan)

## Lineare Regression

$$y_i \sim N(x'_i \beta, \sigma^2)$$

$$\beta_j \sim N(0, 1000)$$

$$\sigma^2 \sim \chi^2(2)$$

# STAN Model

```
data {  
  int<lower=0> N;  
  int<lower=0> M;  
  matrix[N, M] x;  
  vector[N] y;  
}  
parameters {  
  vector[M] beta;  
  real<lower=0> sigma;  
}  
model {  
  y ~ normal(x * beta, sigma);  
  for(i in 1:M)  
    beta[i] ~ normal(0, 1000);  
  sigma ~ chi_square(2);  
}
```