

# Bayesianische Mischverteilungsmodelle

Volker Schmid

17. Juli 2017

## Bayesianische Mischverteilungsmodelle

## Bayesianische Mischverteilungsmodelle

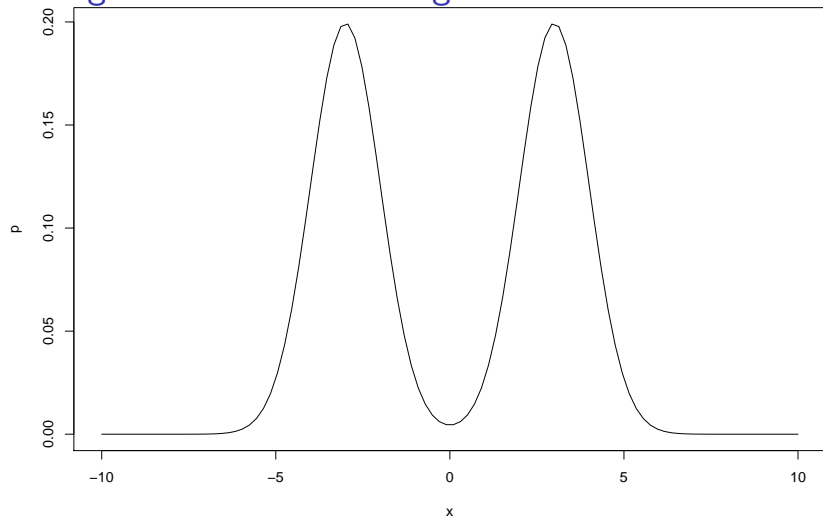
# Mischverteilungen

- ▶ (Normal-)Verteilungsannahme basiert in der Regel auf der Annahme, dass die Beobachtungen identisch verteilt sind, es also eine Verteilung der Grundgesamtheit gibt
- ▶ Alternative Annahme: Es gibt mehrere Gruppen in der Grundgesamtheit, die alle die selbe Verteilungsklasse haben, aber unterschiedliche Parameter
- ▶ Einfachster Fall: Mischung von Normalverteilungen

# Mischung von Normalverteilungen I

```
x <- seq(-10, 10, length=100)
p <- 0.5*dnorm(x,-3,1) + 0.5*dnorm(x,3,1)
plot(x,p, type="l")
```

## Mischung von Normalverteilungen II

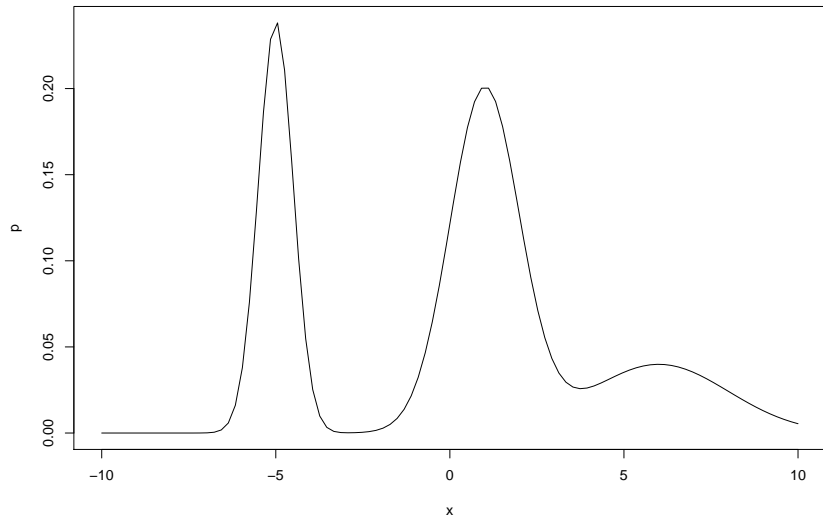


- Natürlich sind auch verschiedene Varianzen und verschiedene Gewichte möglich

## Mischung von Normalverteilungen III

```
x <- seq(-10, 10, length=100)
p <- 0.3*dnorm(x,-5,0.5) + 0.5*dnorm(x,1,1) + 0.2*dnorm(x,6,0.5)
plot(x,p, type="l")
```

# Mischung von Normalverteilungen IV





# Fragen

- ▶ Welche Parameter und welche Gewichte haben die einzelnen Verteilungen?
- ▶ Wieviele Mischverteilungen brauchen wir?
- ▶ Zu welcher Verteilung gehört welche Beobachtung

# Latente Klassen

Idee der Modellierung: Führe latente Klassen  $S_i$  ein

$$x_i | S_i \sim N(\mu_{S_i}, \sigma_{S_i}^2)$$

$$\mu_j \sim N(\mu_0, \sigma_0^2)$$

$$\sigma_j^2 \sim IG(a, b)$$

$$p(S_i = j) = \pi_j$$

$$(\pi_1, \dots, \pi_K) \sim \text{Diri}(\alpha, \dots, \alpha)$$

Dirichlet-Verteilung:

$$p(\pi_1, \dots, \pi_K) \propto \prod \pi_i^{\alpha_i - 1}$$

## Full conditionals

- ▶ Die full conditional von  $\mu_j$  sind Normalverteilungen, wobei die  $x_i$  mit  $S_i = j$  eingehen
- ▶ Analog ist die full conditional von  $\sigma_j^2$  eine IG-Verteilung
- ▶  $p(S_i = j)$  berechnet sich aus Dichte von  $x_i$  mit  $\mu_j$  und  $\sigma_j^2$  (und Priori)
- ▶ Full Conditional von  $\pi$  ist Dirichlet mit  $(\alpha + n_1, \dots, \alpha + n_k)$ , wobei  $n_j$  die aktuelle Anzahl der Beobachtungen in Klasse  $j$  ist

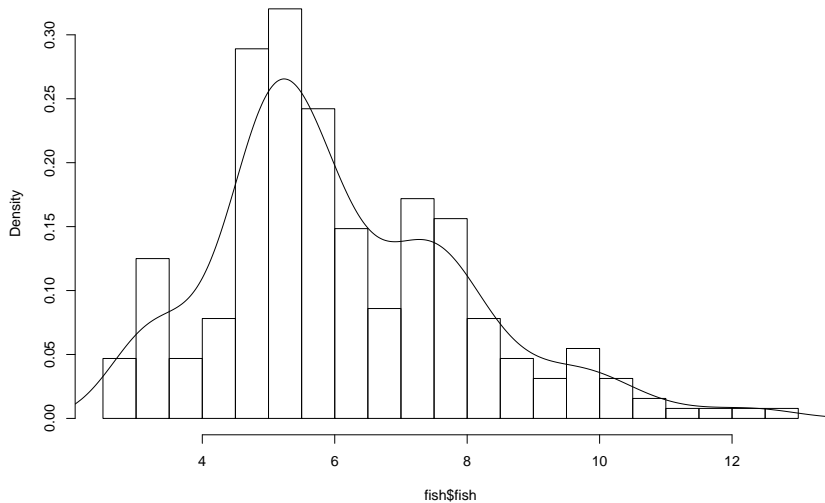
## Beispiel I

Länge von 256 Fischen (aus D. M. Titterington, A. F. M. Smith and U.E. Makov (1985) Statistical Analysis of Finite Mixture Distributions. Wiley.)

```
data(fish, package="bayesmix")  
hist(fish$fish, freq=FALSE, n=22)  
lines(density(fish$fish))
```

## Beispiel II

Histogram of fish\$fish



# bayesmix-Paket I

```
library(bayesmix)
model <- BMMmodel(fish, k = 4,
  initialValues = list(S0 = 2),
  priors = list(kind = "independence",
    parameter = "priorsFish",
    hierarchical = "tau"))
```

```
print(model)
```

```
## Data for nodes: b0, B0inv, nu0Half, g0Half, g0G0Half, k
## Initial values for nodes: eta, mu, tau, S0
##
## Model specification in BUGS language:
##
## var
```

## bayesmix-Paket II

```
##  b0,
##      B0inv,
##      nu0Half,
##      g0Half,
##      g0G0Half,
##      k,
##      N,
##      eta[4],
##  mu[4],
##  tau[4],
##  nu0S0Half,
##      S0,
##      e[4],
##  y[256],
##  S[256];
##
##  model      {
```

## bayesmix-Paket III

```
## for (i in 1:N) {  
##     y[i] ~ dnorm(mu[S[i]],tau[S[i]]);  
##     S[i] ~ dcat(eta[]);  
## }  
## for (j in 1:k) {  
##     mu[j] ~ dnorm(b0,B0inv);  
##     tau[j] ~ dgamma(nu0Half,nu0S0Half);  
## }  
## S0 ~ dgamma(g0Half,g0G0Half);  
## nu0S0Half <- nu0Half * S0;  
##  
## eta[] ~ ddirch(e[]);  
## }
```



# JAGS

- ▶ **JAGS** oder **Just another Gibbs sampler** ist eine Software zur MCMC-Simulation.
- ▶ Definition der Modelle entspricht der bei WinBUGS bzw. OpenBUGS
- ▶ Anbindung an R und andere Sprachen

## Fortsetzung Beispiel I

```
control <- JAGScontrol(variables = c("mu", "tau", "eta",  
  "S"), burn.in = 1000, n.iter = 5000, seed = 10)
```

```
z <- JAGSrun(fish, model = model, control = control)
```

```
## Compiling model graph  
##   Declaring variables  
##   Resolving undeclared variables  
##   Allocating nodes  
## Graph information:  
##   Observed stochastic nodes: 256  
##   Unobserved stochastic nodes: 266  
##   Total graph size: 1047  
##  
## Initializing model
```

## Fortsetzung Beispiel II

```
zSort <- Sort(z, by = "mu")  
zSort
```

```
##  
## Call:  
## JAGSrun(y = fish, model = model, control = control)  
##  
## Markov Chain Monte Carlo (MCMC) output:  
## Start = 1001  
## End = 6000  
## Thinning interval = 1  
##  
## Empirical mean, standard deviation and 95% CI for eta  
##           Mean      SD      2.5%  97.5%  
## eta[1] 0.1189 0.02483 0.07530 0.1731  
## eta[2] 0.5065 0.04458 0.41275 0.5863  
## eta[3] 0.2746 0.04544 0.19466 0.3742
```

## Fortsetzung Beispiel III

```
## eta[4] 0.1000 0.02937 0.05098 0.1658
```

```
##
```

```
## Empirical mean, standard deviation and 95% CI for mu
```

```
##           Mean           SD    2.5%   97.5%
```

```
## mu[1] 3.382 0.15393 3.122   3.729
```

```
## mu[2] 5.278 0.07337 5.130   5.416
```

```
## mu[3] 7.437 0.14866 7.106   7.699
```

```
## mu[4] 9.979 0.39132 9.181  10.696
```

```
##
```

```
## Empirical mean, standard deviation and 95% CI for sigma
```

```
##           Mean           SD    2.5%   97.5%
```

```
## sigma2[1] 0.2996 0.11811 0.1456 0.5876
```

```
## sigma2[2] 0.3504 0.08497 0.2357 0.5174
```

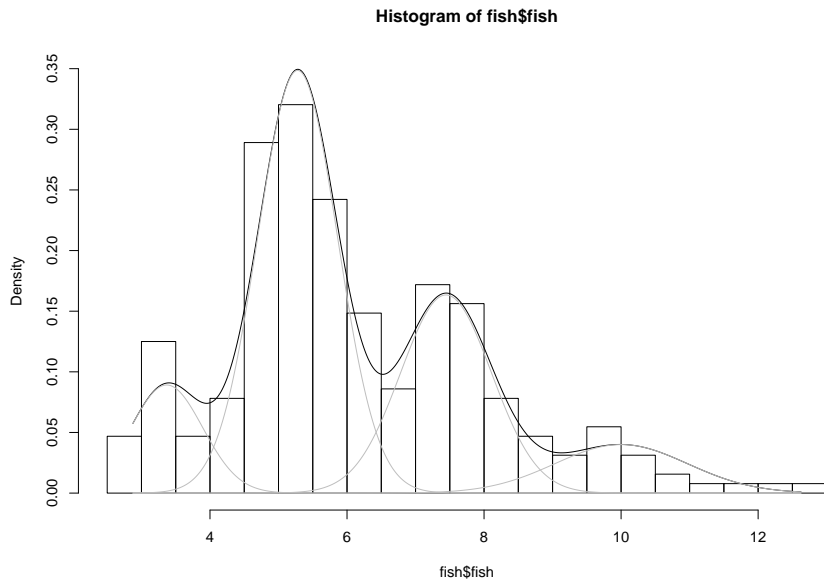
```
## sigma2[3] 0.4815 0.19657 0.2404 0.9607
```

```
## sigma2[4] 1.0023 0.34858 0.5354 1.8441
```

## Medianmodell I

```
postmed<-apply(zSort$results, 2, median)
x<-seq(min(fish$fish), max(fish$fish), length=1000)
d1 <- postmed[257]*dnorm(x, postmed[261], sqrt(postmed[265]))
d2 <- postmed[258]*dnorm(x, postmed[262], sqrt(postmed[266]))
d3 <- postmed[259]*dnorm(x, postmed[263], sqrt(postmed[267]))
d4 <- postmed[260]*dnorm(x, postmed[264], sqrt(postmed[268]))
d <- d1 + d2 + d3 + d4
hist(fish$fish, freq=FALSE, n=22, ylim=c(0,max(d)))
lines(x, d)
lines(x, d1, col="grey")
lines(x, d2, col="grey")
lines(x, d3, col="grey")
lines(x, d4, col="grey")
```

## Medianmodell II



## Posteriori-Verteilung der Klassen I

```
S <- apply(zSort$results[,1:256],2,  
          bioimagetools::table.n,4, percentage=TRUE)  
barplot(S, col=c("red","green","blue","orange"))
```

# Posteriori-Verteilung der Klassen II

