Solution Sheet

Gero Walter

Lund University, 15.12.2015

```
###################################################################
# Solutions for exercise sheet on generalized Bayesian inference    #
# with sets of conjugate priors for dealing with prior-data conflict #
###################################################################
```

Exercise 1

$$f(p \mid s) \propto f(s \mid p) f(p) \tag{1}$$
$$\propto p^s (1-p)^{n-s} p^{\alpha^{(0)}-1} (1-p)^{\beta^{(0)}-1} \tag{2}$$
$$= p^{\alpha+s-1} (1-p)^{\beta+n-s-1} \tag{3}$$

which has the form of the Beta distribution Eq. (3) – remember that $B(\alpha, \beta)$ is just a normalisation constant – with the parameters $\alpha^{(n)} = \alpha^{(0)} + s$ and $\beta^{(n)} = \beta^{(0)} + n - s$.

Exercise 2

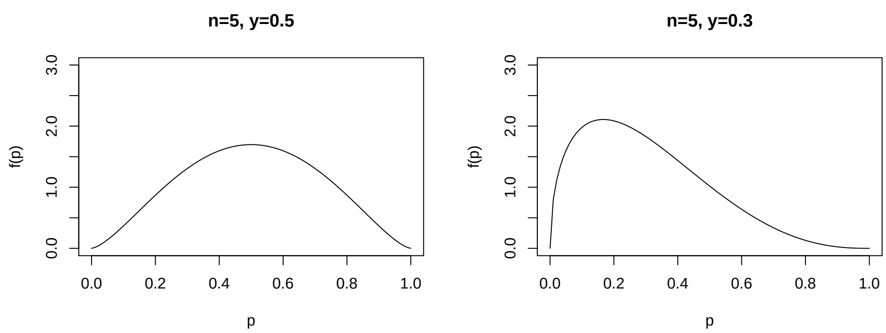
From (5), we get $\alpha^{(0)} = n^{(0)} y^{(0)}$ and $\beta^{(0)} = n^{(0)} - n^{(0)} y^{(0)}$. Inserting into (4) on both sides for each equation (for $^{(0)}$ and $^{(n)}$ versions of α and β), we get

$$n^{(n)} y^{(n)} = n^{(0)} y^{(0)} + s \tag{4}$$
$$n^{(n)} - n^{(n)} y^{(n)} = n^{(0)} - n^{(0)} y^{(0)} + n - s\,. \tag{5}$$

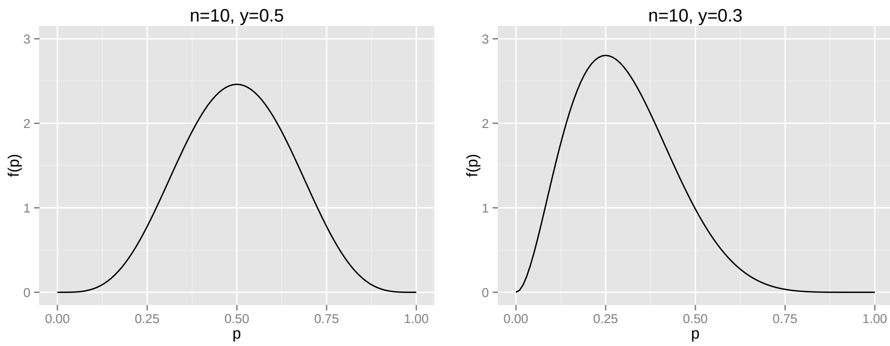
Inserting the first into the second equation and solving for $n^{(n)}$, we get $n^{(n)} = n^{(0)} + n$. Dividing the first equation by $n^{(n)} = n^{(0)} + n$, we get $y^{(n)} = \frac{n^{(0)} y^{(0)} + s}{n^{(0)} + n}$.

Exercise 3

```r
# ----------- Exercise 3 -----------
dbetany <- function(x, n, y, ...){
  dbeta(x, shape1=n*y, shape2=n*(1-y), ...)
}

# with basic plots
xvec <- seq(0,1, length.out=101)
par(mfrow=c(1,2))
plot(xvec, dbetany(x=xvec, n=5, y=0.5), type="l",
     xlab="p", ylab="f(p)", main="n=5, y=0.5", ylim=c(0,3))
plot(xvec, dbetany(x=xvec, n=5, y=0.3), type="l",
     xlab="p", ylab="f(p)", main="n=5, y=0.3", ylim=c(0,3))

# with ggplot2
library(ggplot2)
```




```r
library(gridExtra)
p1 <- qplot(xvec, dbetany(x=xvec, n=10, y=0.5), geom="line",
            xlab="p", ylab="f(p)", main="n=10, y=0.5", ylim=c(0,3))
p2 <- qplot(xvec, dbetany(x=xvec, n=10, y=0.3), geom="line",
            xlab="p", ylab="f(p)", main="n=10, y=0.3", ylim=c(0,3))
grid.arrange(p1, p2, nrow=1, ncol=2, widths=c(1,1))
```



Exercise 4

(i) $n^{(0)} \to 0$:
$y^{(n)} \to \frac{s}{n}$, the ML estimate;
$\mathrm{Var}(p \mid s) \to \frac{\frac{s}{n}(1-\frac{s}{n})}{n+1}$ so increases disregarding the enumerator.
An informative prior will decrease posterior varaiance!

(ii) $n^{(0)} \to \infty$:
$y^{(n)} \to y^{(0)}$, data is ignored with infinitely strong prior;
$\mathrm{Var}(p \mid s) \to 0$, the posterior contracts on $y^{(0)}$.

(iii) $n \to \infty$ when $s/n = \text{const}$:
$y^{(n)} \to \frac{s}{n}$, the ML estimate;
$\mathrm{Var}(p \mid s) \to 0$, the posterior contracts on $\frac{s}{n}$.

Exercise 5

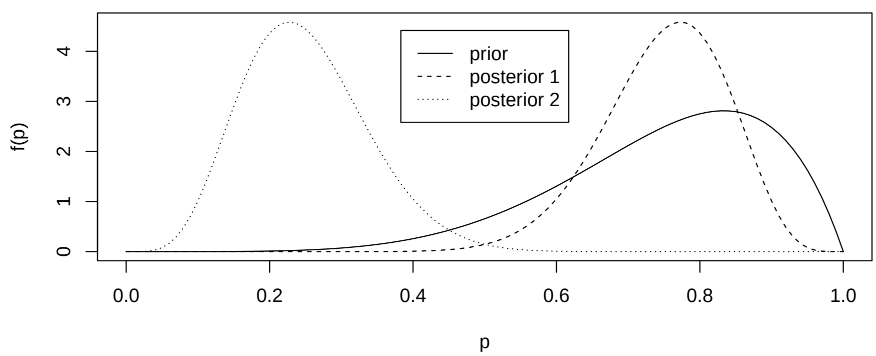
$$
\begin{aligned}
f(\theta \mid \boldsymbol{x}) &\propto f(\boldsymbol{x} \mid \theta) f(\theta) && (6)\\
&\propto \exp\left\{\psi \cdot \tau(\boldsymbol{x}) - n\mathbf{b}(\psi)\right\} \exp\left\{n^{(0)}\left[y^{(0)} \cdot \psi - \mathbf{b}(\psi)\right]\right\} && (7)\\
&= \exp\left\{\left(n^{(0)}y^{(0)} + \tau(\boldsymbol{x})\right)\psi - (n^{(0)} + n)\mathbf{b}(\psi)\right\} && (8)\\
&= \exp\left\{(n^{(0)} + n)\left[\frac{n^{(0)}y^{(0)} + \tau(\boldsymbol{x})}{n^{(0)} + n}\,\psi - \mathbf{b}\right]\right\} && (9)\\
&= \exp\left\{n^{(n)}\left[y^{(n)} \cdot \psi - \mathbf{b}(\psi)\right]\right\}. && (10)
\end{aligned}
$$

Exercise 6

(individual)

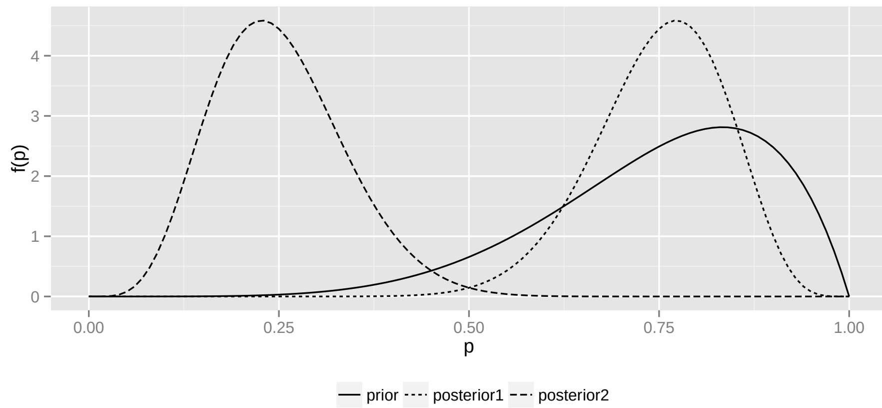
Exercise 7

```
nn <- function(n0, n)
  n0 + n

yn <- function(n0, y0, s, n)
  (n0*y0 + s)/(n0 + n)

xvec <- seq(0,1, length.out=101)
prio1 <- dbetany(x=xvec, n=8, y=0.75)
post1 <- dbetany(x=xvec, n=nn(n0=8, n=16),
                 y=yn(n0=8, y0=0.75, s=12, n=16))
post2 <- dbetany(x=xvec, n=nn(n0=8, n=16),
                 y=yn(n0=8, y0=0.75, s=0, n=16))

# with basic plots
plot(xvec, prio1, type="l", xlab="p", ylab="f(p)",
     ylim=c(0,max(c(prio1, post1, post2))))
lines(xvec, post1, lty=2)
lines(xvec, post2, lty=3)
legend(x=0.5, y=3.5, lty=1:3, xjust=0.5, yjust=0.5,
       c("prior", "posterior 1", "posterior 2"))
```




```
# with ggplot2
library(reshape2)
df1 <- data.frame(p=xvec, "prior"=prio1,
                  "posterior1"=post1, "posterior2"=post2)
df1 <- melt(df1, "p")
bottomlegend <- theme(legend.position = "bottom",
                      legend.direction = "horizontal",
```

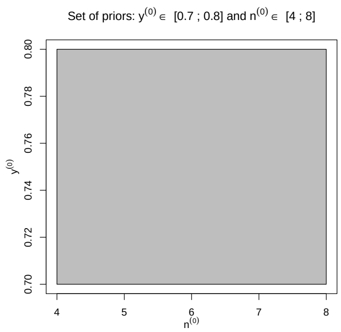
```
                    legend.title = element_blank())
ggplot(df1, aes(x=p, y=value, group=variable, linetype=variable)) +
  geom_line() + bottomlegend + ylab("f(p)")
```



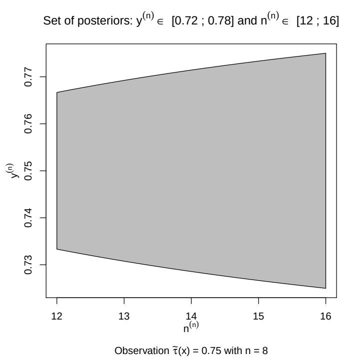
Exercise 8

```
# ----------- Exercise 8 -----------
# installing the luck package
#install.packages("TeachingDemos")
#luckpath <- "http://download.r-forge.r-project.org/src/contrib/luck_0.9.tar.gz"
#install.packages(luckpath, repos = NULL, type = "source")
library(luck)

## Loading required package:  TeachingDemos
##
## Attaching package:  'luck'
##
## Das folgende Objekt ist maskiert 'package:utils':
##
##     data

# (i)
data1 <- LuckModelData(n=8, tau=6)
luck1 <- LuckModel(n0=c(4,8), y0=c(0.7, 0.8), data=data1)
luck1

## generalized iLUCK model with prior parameter set:
##   lower n0 = 4   upper n0 = 8
##   lower y0 = 0.7   upper y0 = 0.8
##  giving a main parameter prior imprecision of 0.1
## and data object with sample statistic tau(x) = 6 and sample size n = 8
```

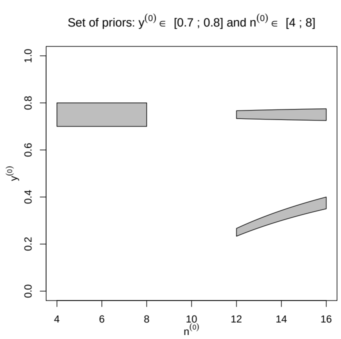
```r
data2 <- LuckModelData(n=8, tau=0)
luck2 <- LuckModel(n0=c(4,8), y0=c(0.7, 0.8), data=data2)
luck2

## generalized iLUCK model with prior parameter set:
##   lower n0 = 4   upper n0 = 8
##   lower y0 = 0.7   upper y0 = 0.8
##  giving a main parameter prior imprecision of 0.1
## and data object with sample statistic tau(x) = 0 and sample size n = 8

# you can access the object slots with functions of the same name
y0(luck2)

##      lower upper
## [1,]   0.7   0.8

n0(luck2)

##      lower upper
## [1,]     4     8

data(luck2)

## data object with sample statistic tau(x) = 0 and sample size n = 8

tau(data2)

## tau
##   0

n(data2)

## n
## 8

# (ii)
#?luck::plot
plot(luck1) # the luck package plots in basic plots only
```



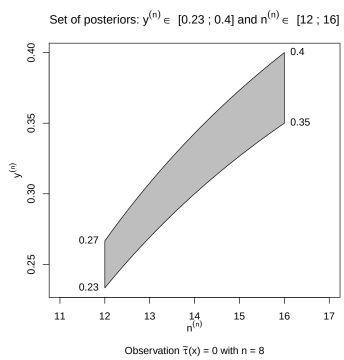

```
plot(luck1, control=controlList(posterior=TRUE))
```



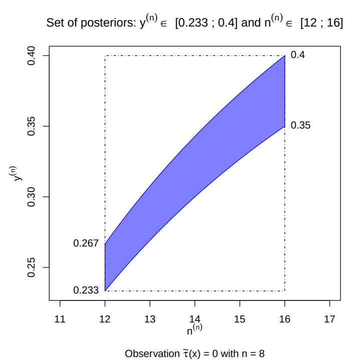

```
# prior and both posterior sets in one plot
plot(luck1, xlim=c(4,16), ylim=c(0,1))
plot(luck1, add=TRUE, control=controlList(posterior=T, annotate=F))
plot(luck2, add=TRUE, control=controlList(posterior=T, annotate=F))
```



```r
# there is an option to display the y values for the four corners
# of the set; you might need to set xlim to make them visible
plot(luck2, xlim=c(11,17),
     control=controlList(posterior=TRUE, numbers=TRUE))
```




```r
# using some other options in controlList()
plot(luck2, xlim=c(11,17),
     control=controlList(posterior=TRUE, numbers=TRUE, rDigits=3,
                         polygonCol=rgb(r=0, g=0, b=1, alpha=0.5),
                         borderCol="blue", rectangle=TRUE))
```



Exercise 9

$\tau(\boldsymbol{x})$ does not appear in (23); for any sample of size n, the prior imprecision decreases by the factor $\frac{n^{(0)}}{n^{(0)}+n}$!

Exercise 10

The cases $\tilde{\tau}(\boldsymbol{x}) < \underline{y}^{(0)}$ and $\tilde{\tau}(\boldsymbol{x}) > \overline{y}^{(0)}$ correspond to prior-data conflict; in both cases $\underline{n}^{(0)}$ is used to calculated the extreme value of $y^{(n)}$ closer to $\tilde{\tau}(\boldsymbol{x})$, while

$\overline{n}^{(0)}$ is used to calculated the other extreme of $y^{(n)}$. The 'banana shape' shows this as well, one extreme is found at the left side corners of $\Pi^{(n)}$, the other on the right side corners of $\Pi^{(n)}$.

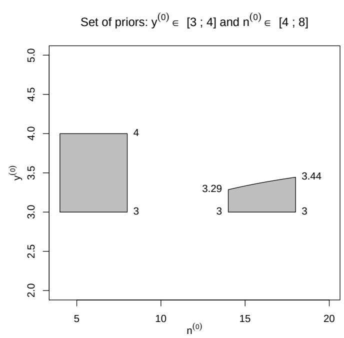
Exercise 11

When $\tilde{\tau}(\boldsymbol{x}) \in [\underline{y}^{(0)}, \overline{y}^{(0)}]$, then both $\underline{y}^{(n)}$ and $\overline{y}^{(n)}$ are calculated with $\overline{n}^{(0)}$; both extremes are found on the right side corners of $\Pi^{(n)}$.

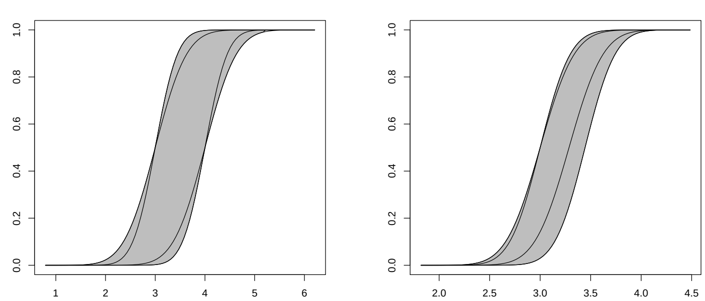
Exercise 12

```
# ----------- Exercise 12 -----------
# different ways to create a ScaledNormalData object
data3 <- ScaledNormalData(data1) # from a plain LuckModelData object
data3

## ScaledNormalData object containing a mean of 0.75 for sample size 8 .

data4 <- ScaledNormalData(mean=3, n=10) # with mean and sample size
data4

## ScaledNormalData object containing a mean of 3 for sample size 10 .

data5 <- ScaledNormalData(rnorm(10)) # with a vector of observations
data5

## ScaledNormalData object containing data of sample size 10
##  with mean -0.6737432 and variance 0.4461565 .

data6 <- ScaledNormalData(mean=3, n=10, sim=TRUE) # simulating according to
data6                                             # mean and sample size

## ScaledNormalData object containing data of sample size 10
##  with mean 2.844176 and variance 0.8792738 .

# two ways to create a ScaledNormalLuckModel object
luck3 <- ScaledNormalLuckModel(luck1) # from a plain LuckModel object
luck3

## generalized iLUCK model for inference from scaled normal data
## with prior parameter set:
##   lower n0 = 4   upper n0 = 8
##   lower y0 = 0.7   upper y0 = 0.8
##  giving a main parameter prior imprecision of 0.1
## corresponding to a set of normal priors
##  with means in [ 0.7 ; 0.8 ] and variances in [ 0.125 ; 0.25 ]
## and ScaledNormalData object containing a mean of 0.75 for sample size 8 .
```

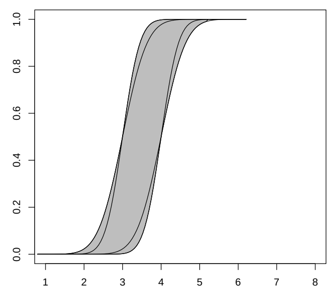
```
luck4 <- ScaledNormalLuckModel(n0=c(4,8), y0=c(3,4), data=data4)
luck4                                  # by supplying n0 and y0

## generalized iLUCK model for inference from scaled normal data
## with prior parameter set:
##    lower n0 = 4   upper n0 = 8
##    lower y0 = 3   upper y0 = 4
##   giving a main parameter prior imprecision of 1
## corresponding to a set of normal priors
##   with means in [ 3 ; 4 ] and variances in [ 0.125 ; 0.25 ]
## and ScaledNormalData object containing a mean of 3 for sample size 10 .

# prior and posterior parameter sets
plot(luck4, xlim=c(4,20), ylim=c(2,5), control=controlList(numbers=T))
plot(luck4, control=controlList(posterior=T, annotate=F, numbers=T),
     add=TRUE)
```

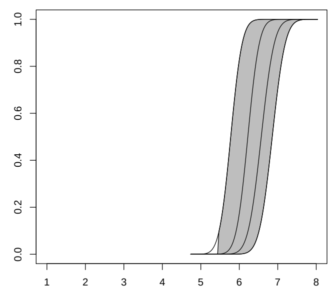



```
# prior and posterior sets of cdfs
#?luck::cdfplot
cdfplot(luck4)
cdfplot(luck4, control=controlList(posterior=TRUE))
```



```r
# scaled normal LuckModel with prior-data conflict
luck5 <- luck4
data(luck5) <- ScaledNormalData(mean=8, n=10)
cdfplot(luck5, xlim=c(1,8))
cdfplot(luck5, xlim=c(1,8), control=controlList(posterior=TRUE))
```

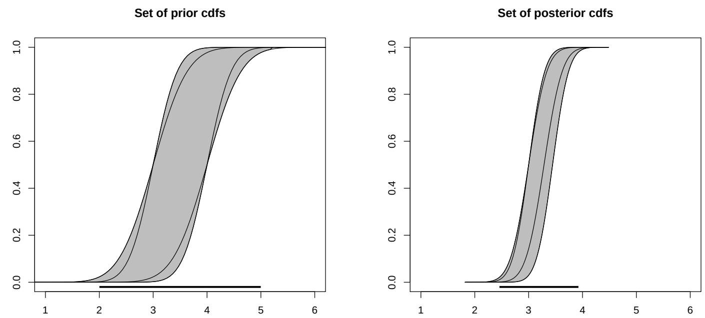




Exercise 13

```r
# ----------- Exercise 13 -----------
#?unionHdi
                # outputs a list containing the interval bounds
unionHdi(luck4) # and the parameters which produce the bounds

## $borders
## [1] 2.020018 4.979982
##
## $lowpars
## $lowpars$n
##
## 4
##
## $lowpars$y
## lower
##     3
##
##
## $upppars
## $upppars$n
##
## 4
##
## $upppars$y
```

```
## upper
##     4

# prior highest density interval and interval length
unionHdi(luck4)$borders

## [1] 2.020018 4.979982

diff(unionHdi(luck4)$borders)

## [1] 2.959964

# posterior highest density interval and interval length
# no prior-data conflict: much shorter
unionHdi(luck4, posterior=TRUE)$borders

## [1] 2.476178 3.906412

diff(unionHdi(luck4, posterior=TRUE)$borders)

## [1] 1.430235

# prior-data conflict: less shorter, reflecting prior-data conflict
unionHdi(luck5, posterior=TRUE)$borders

## [1] 5.315810 7.380965

diff(unionHdi(luck5, posterior=TRUE)$borders)

## [1] 2.065155

# prior and posterior sets of cdfs with highest density intervals
cdfplot(luck4, main="Set of prior cdfs", xlim=c(1,6))
lines(unionHdi(luck4)$borders, rep(-0.02,2), lwd=3, lend=2)
cdfplot(luck4, control=controlList(posterior=TRUE),
        main="Set of posterior cdfs", xlim=c(1,6))
lines(unionHdi(luck4, posterior=TRUE)$borders, rep(-0.02,2), lwd=3, lend=2)
```



Exercise 14

(individual)