The Beta Prior, Likelihood, and Posterior

The Beta distribution (and more generally the Dirichlet) are probably my favorite distributions. However, sometimes only limited information is available when trying set up the distribution. For example maybe you only know the lowest likely value, the highest likely value and the median, as a measure of center. That information is sufficient to construct a basic form of the distribution. The idea of this post is not to elaborate in detail on Bayesian priors and posteriors but to give a real working example of using a prior with limited knowledge about the distribution, adding some collected data and arriving at a posterior distribution along with a measure of its uncertainty.

The example below is a simple demonstration on how a prior distribution and current data can be combined and form a posterior distribution. Earlier this year I gave a presentation at a conference where I modified this simple version of my code to be substantially more complex and I used the Dirichlet distribution to make national predictions based on statewide and local samples. So this approach has some very useful applied statistical properties and can be modified to handle some very complex distributions.

Prior: beta(52.22,9.52); Data: B(50,25); Posterior: beta(77.22,34.52)
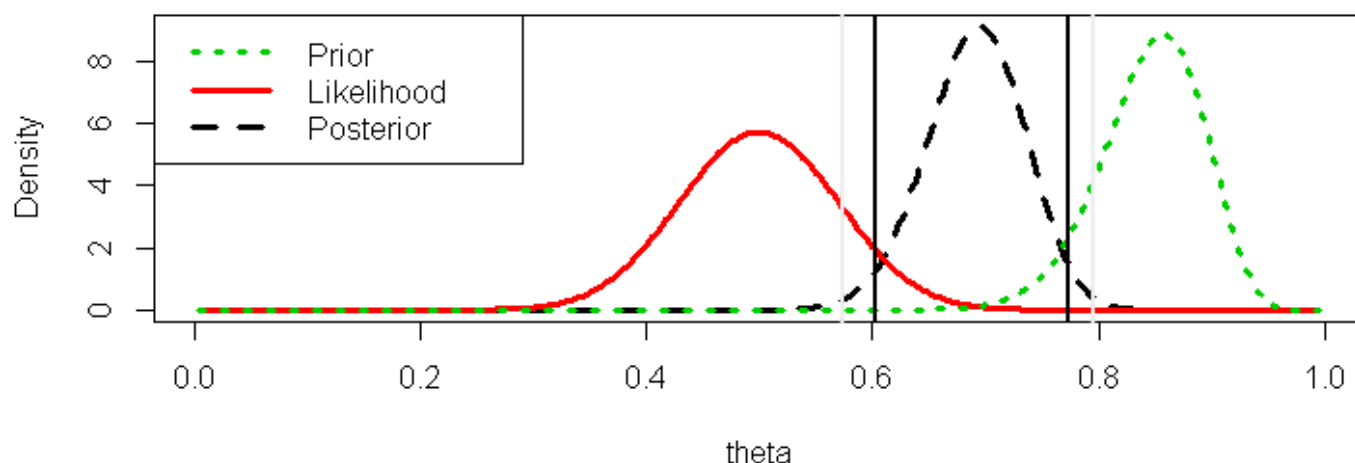
A Note on Posterior Intervals

The posterior interval (also called a *credible interval or credible region*) provides a very intuitive way to describe the measure of uncertainty. Unlike a **confidence interval (discussed in one of my previous posts)**, a *credible interval* does in fact provide the probability that a value exists within the interval. With this interval it is based on calculating the probability of different values given the data. The graph above shows the different values (identified as theta) as well as the simulated posterior interval limits

(alpha=.05 in black and .01 in light gray). In other words the probability that theta is a member of the 95% credible interval is 0.95 (written as $P(\theta \in CI) = 0.95$).

The Prior and Posterior Distribution: An Example

The code to run the beta.select() function is found in the **LearnBayes** package. This is a great function because by providing two quantiles one can determine the shape parameters of the Beta distribution. This is useful to find the parameters (or a close approximation) of the prior distribution given only limited information. If additional quantiles are known then they can be incorporated to better determine the shape parameters of the Beta distribution.

```
001.  library(LearnBayes)
002.  Q = data.frame(
003.  quantile=c(
004.  median=0.5,
005.  maximum=0.99999,
006.  minimum=0.00001),
007.  prior=c(
008.  median=0.85,
009.  maximum=0.95,
010.  minimum=0.60)
011.  )
012.
013.  optimalBeta = function(Q) {
014.  q1q = Q$quantile[1]
015.  q1p = Q$prior[1]
016.  q2q = Q$quantile[2]
017.  q2p = Q$prior[2]
018.  q3q = Q$quantile[3]
019.  q3p = Q$prior[3]
020.
021.  # find the beta prior using quantile1 and quantile2
022.  q.med = list(p=q1q, x=q1p)
023.  q.max = list(p=q2q, x=q2p)
024.  q.min = list(p=q3q, x=q3p)
025.
026.  # prior parameters using median and max, and median and min
027.  prior.A = beta.select(q.med,q.max)
028.  prior.B = beta.select(q.med,q.min)
029.
030.  prior.Aa = prior.A[1]
031.  prior.Ab = prior.A[2]
032.
033.  prior.Ba = prior.B[1]
034.  prior.Bb = prior.B[2]
035.
036.  ## find the best possible beta prior
037.  ## Set a start and stop point range to find the best parameters
038.  if (prior.Aa < prior.Ba) {
039.  start.a = prior.Aa
040.  stop.a = prior.Ba
041.  } else {
```

```
042.    start.a = prior.Ba
043.    stop.a = prior.Aa
044.    }
045.
046.    if (prior.Ab < prior.Bb) {
047.    start.b = prior.Ab
048.    stop.b = prior.Bb
049.    } else {
050.    start.b = prior.Bb
051.    stop.b = prior.Ab
052.    }
053.    seq.a = seq(from=start.a, to=stop.a, length.out=1000)
054.    seq.b = seq(from=start.b, to=stop.b, length.out=1000)
055.
056.    seq.grid = expand.grid(seq.a, seq.b)
057.
058.    prior.C.q1 = qbeta(q1q, seq.grid[,1], seq.grid[,2])
059.    prior.C.q2 = qbeta(q2q, seq.grid[,1], seq.grid[,2])
060.    prior.C.q3 = qbeta(q3q, seq.grid[,1], seq.grid[,2])
061.
062.    ## Different distance measurements, manhattan, euclidean, or otherwise.
063.    ## It would be interesting to run a simulation to measure a variety of
           distance measurements.
064.    prior.C.delta = abs(prior.C.q1 - q1p) + abs(prior.C.q2 - q2p) +
           abs(prior.C.q3 - q3p)
065.    ## prior.C.delta = sqrt( (prior.C.q1 - q1p)^2 + (prior.C.q2 - q2p)^2 +
           (prior.C.q3 - q3p)^2 )
066.
067.    optimize.seq = cbind(seq.grid, prior.C.q1, prior.C.q2, prior.C.q3,
           prior.C.delta)
068.
069.    ## Minimize the delta, if the min-delta is not unique then choose the first
           occurence
070.    best.a = optimize.seq[,1][ optimize.seq[,6]==min(optimize.seq[,6])][1]
071.    best.b = optimize.seq[,2][ optimize.seq[,6]==min(optimize.seq[,6])][1]
072.
073.    return(list(a=best.a,b=best.b))
074.    }
075.
076.    prior.dist = optimalBeta(Q)
077.
078.    ############################################################
079.    ## Take a look at only the prior
080.    ############################################################
081.    curve(dbeta(x,prior.dist$a,prior.dist$b)) # plot the prior
082.    abline(v=Q$prior[1])
083.
084.    ############################################################
085.    ## Take a look at only the likelihood with given successes
086.    ############################################################
087.    calcLikelihood = function(successes, total){
088.    curve(dbinom(successes,total,x)) # plot the likelihood
089.    }
090.
091.    calcLikelihood(45, 50) ## e.g. 45/50 sucesses
092.    ## calculate some properties of the Beta distribution
093.    calcBetaMode = function(aa, bb) {
094.    beta.mode = (aa - 1)/(aa + bb - 2)
095.    return(beta.mode)
096.    }
097.    calcBetaMean = function(aa, bb) {
```

```
098.   beta.mean = (aa)/(aa + bb)
099.   return(beta.mean)
100.   }
101.   calcBetaVar = function(aa, bb) {
102.   beta.var = (aa * bb)/(((aa + bb)^2) * (aa + bb + 1))
103.   return(beta.var)
104.   }
105.   calcBetaMedian = function(aa, bb) {
106.   beta.med = (aa-1/3)/(aa+bb-2/3)
107.   return(beta.med)
108.   }
109.   calcBetaSkew = function(aa, bb) {
110.   beta.skew = ( 2*(bb-aa)*sqrt(aa+bb+1) ) /( (aa+bb+2)/sqrt(aa+bb) )
111.   return(beta.skew)
112.   }
113.
114.   ################################################################
115.   ## Take a look at the prior, likelihood, and posterior
116.   ################################################################
117.   priorToPosterior = function(successes, total, a, b) {
118.   ## Note the rule of succession
119.   likelihood.a = successes + 1
120.   likelihood.b = total - successes + 1
121.
122.   ## Create posterior
123.   posterior.a = a + successes;
124.   posterior.b = b + total - successes
125.   theta = seq(0.005, 0.995, length = 500)
126.
127.   ## Calc density
128.   prior = dbeta(theta, a, b)
129.   likelihood = dbeta(theta, likelihood.a, likelihood.b)
130.   posterior = dbeta(theta, posterior.a, posterior.b)
131.
132.   ## Plot prior, likelihood, and posterior
133.
134.   ## Can be used to scale down the graph if desired.
135.   ## However, the density is different for each prior, likelihood, posterior
136.   m.orig = apply( cbind(prior, likelihood, posterior), 2, max)
137.   m = max(c(prior, likelihood, posterior))
138.
139.   plot(theta, posterior, type = "l", ylab = "Density", lty = 2, lwd = 3,
140.   main = paste("Prior: beta(", round(a,2), ",", round(b,2), "); Data: B(",
           total, ",", successes, "); ",
141.   "Posterior: beta(", round(posterior.a,2), ",", round(posterior.b,2), ")",
           sep=""), ylim = c(0, m), col = 1)
142.   lines(theta, likelihood, lty = 1, lwd = 3, col = 2)
143.   lines(theta, prior, lty = 3, lwd = 3, col = 3)
144.   legend("topleft",y=m, c("Prior", "Likelihood", "Posterior"), lty = c(3, 1,
           2),
145.   lwd = c(3, 3, 3), col = c(3, 2, 1))
146.
147.   prior.mode = calcBetaMode(a, b)
148.   likelihood.mode = calcBetaMode(likelihood.a, likelihood.b)
149.   posterior.mode = calcBetaMode(posterior.a, posterior.b)
150.   prior.mean = calcBetaMean(a, b)
151.   likelihood.mean = calcBetaMean(likelihood.a, likelihood.b)
152.   posterior.mean = calcBetaMean(posterior.a, posterior.b)
153.   prior.med = calcBetaMedian(a, b)
154.   likelihood.med = calcBetaMedian(likelihood.a, likelihood.b)
155.   posterior.med = calcBetaMedian(posterior.a, posterior.b)
```

```r
156. prior.var = calcBetaVar(a, b)
157. likelihood.var = calcBetaVar(likelihood.a, likelihood.b)
158. posterior.var = calcBetaVar(posterior.a, posterior.b)
159. prior.skew = calcBetaSkew(a, b)
160. likelihood.skew = calcBetaSkew(likelihood.a, likelihood.b)
161. posterior.skew = calcBetaSkew(posterior.a, posterior.b)
162.
163. print(paste("Mode: prior=",prior.mode,"; Likelihood=",likelihood.mode,";
         Posterior=",posterior.mode))
164. print(paste("Mean: prior=",prior.mean,"; Likelihood=",likelihood.mean,";
         Posterior=",posterior.mean))
165. print(paste("~Approx Median (for a and b > 1): prior=",prior.med,";
         Likelihood=",likelihood.med,", for Posterior=",posterior.med))
166. print(paste("Var: prior=",prior.var,"; Likelihood=", likelihood.var,";
         Posterior=",posterior.var))
167. print(paste("Skewness: prior=",prior.skew,"; Likelihood=",likelihood.skew,";
         Posterior=",posterior.skew))
168. return(list(a=posterior.a,b=posterior.b))
169. }
170.
171. posterior.out = priorToPosterior(25,50, prior.dist$a, prior.dist$b) # 25/50
         is current data
172. beta.sim = rbeta(1000000,posterior.out$a, posterior.out$b)
173. abline(v=quantile(beta.sim, prob=c(.05/2, 1-.05/2)), col='#000000', lwd=2)
174. abline(v=quantile(beta.sim, prob=c(.01/2, 1-.01/2)), col='#EEEEEE', lwd=2)
```

Share this:

Like this:

Posted by Wesley on September 4, 2013

http://statistical-research.com/the-beta-prior-likelihood-and-posterior/