Likelihood/Posterior Functions

```r
require(dirichletprocess)
```

```
## Loading required package: dirichletprocess
```

```r
dp <- DirichletProcessGaussian(scale(faithful$waiting))
dp <- Fit(dp, 1000, progressBar = F)
```

LikelihoodDP

This returns a vector of length 272. Each of the datapoints evaluated with its fitted cluster parameters.

```r
length(LikelihoodDP(dp))
```

```
## [1] 272
```

LikelihoodFunction

This returns a function that can then be evaluated. By default it uses the last fitted clusters and weights:

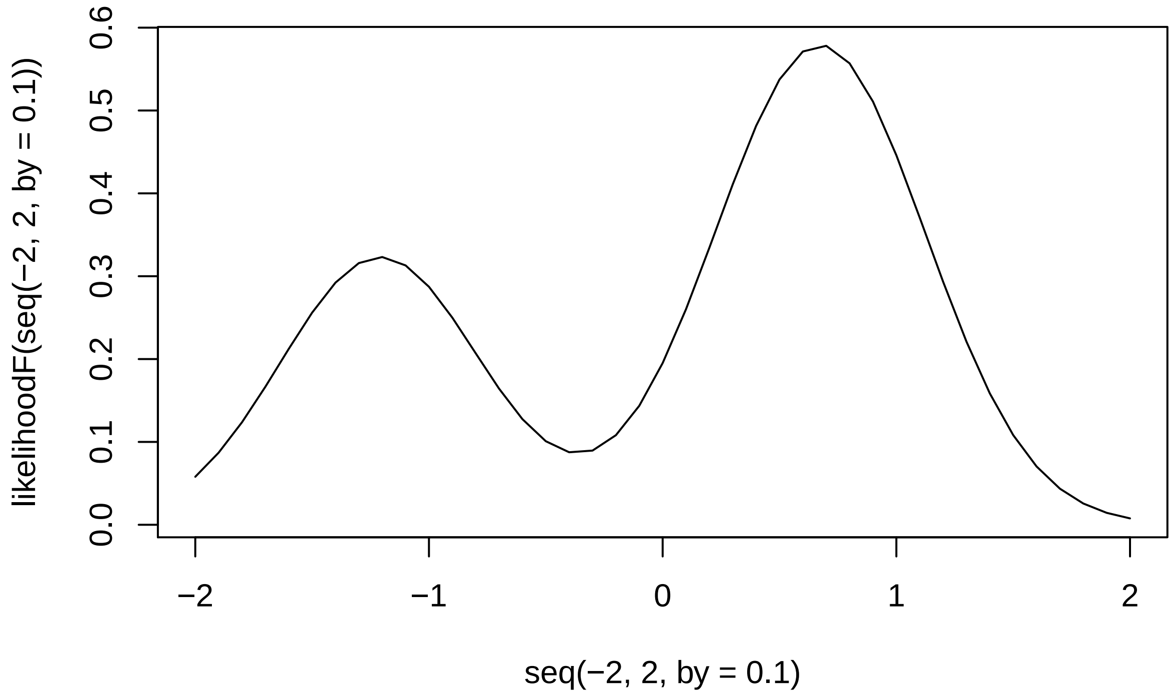
```r
cbind(Weight = dp$pointsPerCluster,
      MuParams = dp$clusterParameters[[1]],
      SigmaParam = dp$clusterParameters[[2]])
```

```
##      Weight   MuParams SigmaParam
## [1,]     94 -1.2087895  0.4267083
## [2,]    178  0.6742647  0.4509048
```

Which are then wrapped into a function

```r
likelihoodF <- LikelihoodFunction(dp)

plot(seq(-2, 2, by=0.1),
     likelihoodF(seq(-2, 2, by=0.1)),
     type="l")
```



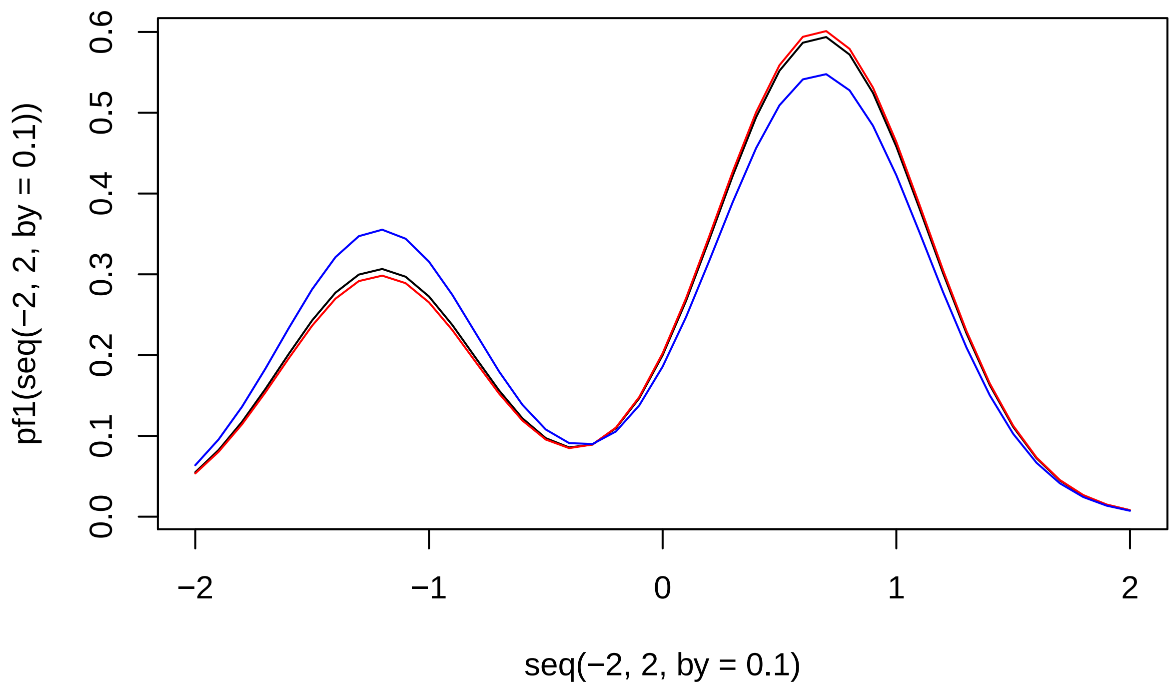
PosteriorFunction

Instead of using the fitted parameters exactly, we draw from the distribution of the fitted parameters, so each call of the function will be slightly different.

Here we draw thrice and plot the results.

```
pf1 <- PosteriorFunction(dp)
pf2 <- PosteriorFunction(dp)
pf3 <- PosteriorFunction(dp)

plot(seq(-2, 2, by=0.1),
     pf1(seq(-2, 2, by=0.1)), type="l")
lines(seq(-2, 2, by=0.1),
     pf2(seq(-2, 2, by=0.1)), type="l", col="red")
lines(seq(-2, 2, by=0.1),
     pf3(seq(-2, 2, by=0.1)), type="l", col="blue")
```



P(Y_{n+1}|pi)

So I think to calculate your interested quantity you need to do the following:

```r
yNP1 <- seq(0, 2, by=0.1)

pfEval <- matrix(nrow=length(yNP1), ncol=500)

j <- 1

for(i in 500:100){

  pf <- PosteriorFunction(dp, i)
  pfEval[, j] <- pf(yNP1)

  j <- j + 1

}

head(pfEval[, 1:5])
```

```
##           [,1]      [,2]      [,3]      [,4]      [,5]
## [1,] 0.1739152 0.2204616 0.2067913 0.1851379 0.1643034
## [2,] 0.2212319 0.2811688 0.2542432 0.2253165 0.2100700
## [3,] 0.2789098 0.3487314 0.3094896 0.2770509 0.2699716
## [4,] 0.3412968 0.4170605 0.3672830 0.3363455 0.3402828
## [5,] 0.4012385 0.4790763 0.4212334 0.3972877 0.4146357
## [6,] 0.4510063 0.5276445 0.4646480 0.4526803 0.4845796
```

So each column is a sample of your predictive density and each row is a $Y_N + 1$.