Code

Teng Wei Yeo

```r
knitr::opts_chunk$set(fig.width=8, fig.height=5)
```

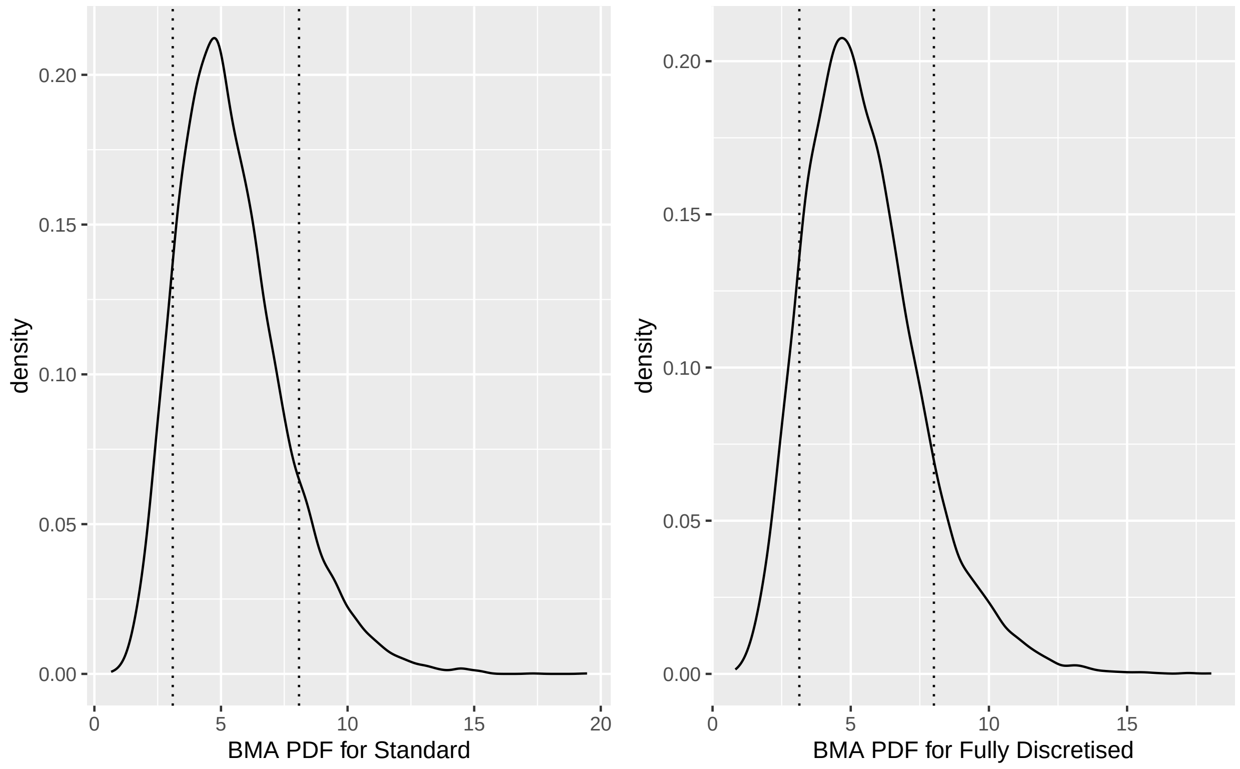
```r
# Teng Wei Yeo, S2566430
# MATH11188 Statistical Research Skills: Assignment 1.
# Ideas are from Sloughter, J. M., Gneiting, T., and Raftery, A. E. (2010),
# 'Probabilistic wind speed forecasting using ensembles and Bayesian model
# averaging', Journal of the American Statistical Association, 105(489),
# pp. 25-35.

### Brief Outline ###
# This code aims to simulate if there is any difference in the BMA PDF for
# the fully discretised method, versus the standard method.
# This code assumes there is only 1 location, with one observation for each
# day over 20 days. For this example, all observations are training data.

# An ensemble of 4 members are used. This toy example assumes that the
# individual prediction PDF of each member and all the hyperparameters
# are known a priori.

# The E and CM-1 steps suggested in Sloughter et al. (2010) for deriving
# the weights which maximise the likelihood is then applied to derive
# the BMA PDF which best fits the observed data.

# The BMA PDF of the standard method, vs. the fully discretised method,
# are compared. Note that no actual forecasting is being done in this toy
# example. Neither is there the need for an iterative EMCE algorithm, since the
# hyperparameters of the individual PDFs are known a priori. The goal is to
# simply compare the shape of the BMA PDFs as a result of the two different
# methods.

# The output of this code is a graph showing the BMA PDF of the Standard Method,
# and the BMA PDF of the Fully Discretised Method.
# The code also prints the 77.8% symmetric prediction interval for both methods.

# The conclusion is that both methods arrive at very similar BMA PDFs,
# and have almost identical 77.8% symmetric prediction intervals.
```

```r
############# Model set-up #############
library(ggplot2)
library(gridExtra)
set.seed(4)
# True distribution of observations
true_a = 1.5
true_b = 0.3
```

```
obs <- rgamma(20, shape=true_a, rate=true_b) # generate 20 observations
true_var = true_a/true_b^2

# Discretisation of the observations
obs2 <- rep(0, 20)
for (i in 1:length(obs)){
  if (obs[i] <= 1){
    obs2[i] = 0
  } else if (obs[i] <= 3/2){
    obs2[i] = 1
  } else{
    obs2[i] = round(obs[i], 0)
  }
}

# Arbitrarily chosen hyperparameters for the gamma distributions of each of the
# four forecast members.
beta = c(0.1, 0.2, 0.4, 0.5)
alpha = beta^2 * true_var

############# Start of BMA approach #############
### 1) Standard Method ###
# Starting weights
w = c(0.25, 0.25, 0.25, 0.25)
y = obs2
z_hat = c(0,0,0,0)

# Optimising weights using E and CM-1 steps
for (i in 1:length(y)){
  if (y[i] == 0){ # use CDF for values = 0
    combined = sum(w * pgamma(1, shape = alpha, rate = beta))
    z_hat = z_hat + w * pgamma(1, shape = alpha, rate = beta) /
      combined
  } else{ # use PDF for all other values
    combined = sum(w * dgamma(y[i], shape = alpha, rate = beta))
    z_hat = z_hat + w * dgamma(y[i], shape = alpha, rate = beta) /
      combined
    }
}
w = z_hat/20

# Calculating the maximum likelihood, for standard method
sum = 0
for (i in 1:length(y)){
  if (y[i] == 0){
    sum = sum + log(sum(w * pgamma(1, shape = alpha, rate = beta)))
  } else{
    sum = sum + log(sum(w * dgamma(y[i], shape = alpha, rate = beta)))
  }
}

# Simulate 10000 random variates for BMA PDF of Standard Method.
collect = w[1] * rgamma(10000, shape = alpha[1], rate = beta[1]) +
          w[2] * rgamma(10000, shape = alpha[2], rate = beta[2]) +
          w[3] * rgamma(10000, shape = alpha[3], rate = beta[3]) +
          w[4] * rgamma(10000, shape = alpha[4], rate = beta[4])
collect_df = data.frame(collect)

# Plot density of the 10000 random variates, with 77.8% interval
plot1 <- ggplot(collect_df, aes(x = collect)) +
          geom_density() +
          geom_vline(xintercept=quantile(collect, probs = c(0.111,0.889)),
                     linetype="dotted") +
          xlab("BMA PDF for Standard")
w_saved = w

### 2) Fully Discretised Method ###
# Starting weights
w = c(0.25, 0.25, 0.25, 0.25)
z_hat = c(0,0,0,0)

# Optimising weights
for (i in 1:length(y)){
  if (y[i] == 0){
    combined = sum(w * pgamma(1, shape = alpha, rate = beta))
    z_hat = z_hat + w * pgamma(1, shape = alpha, rate = beta) /
      combined
  } else if (y[i] == 1){
    combined = sum(w * (pgamma(1.5, shape = alpha, rate = beta) -
                     pgamma(1, shape = alpha, rate = beta)))
    z_hat = z_hat + w * (pgamma(1.5, shape = alpha, rate = beta) -
                         pgamma(1, shape = alpha, rate = beta)) /
                         combined
  } else {
    combined = sum(w * (pgamma(y[i] + 0.5, shape = alpha, rate = beta) -
                     pgamma(y[i] - 0.5, shape = alpha, rate = beta)))
    z_hat = z_hat + w * (pgamma(y[i] + 0.5, shape = alpha, rate = beta) -
                         pgamma(y[i] - 0.5, shape = alpha, rate = beta)) /
                         combined
    }
}
w = z_hat/20

# Calculating the maximum likelihood, for fully discretised method
sum_2 = 0
for (i in 1:length(y)){
  if (y[i] == 0){
    sum_2 = sum_2 + log(sum(w * pgamma(1, shape = alpha, rate = beta)))
  } else if (y[i] == 1){
    sum_2 = sum_2 + log(sum(w * (pgamma(1.5, shape = alpha, rate = beta) -
                             pgamma(1, shape = alpha, rate = beta))))
  } else {
    sum_2 = sum_2 + log(sum(w * (pgamma(y[i] + 0.5, shape = alpha, rate = beta) -
                               pgamma(y[i] - 0.5, shape = alpha, rate = beta))))
  }
}
```

```
# Simulate 10000 random variables for BMA PDF of Fully Discretised BDA.
collect_2 = w[1] * rgamma(10000, shape = alpha[1], rate = beta[1]) +
            w[2] * rgamma(10000, shape = alpha[2], rate = beta[2]) +
            w[3] * rgamma(10000, shape = alpha[3], rate = beta[3]) +
            w[4] * rgamma(10000, shape = alpha[4], rate = beta[4])
collect_df_2 = data.frame(collect_2)

# Plot density of the 10000 random variates, with 77.8% interval
plot2 <- ggplot(collect_df_2, aes(x = collect_2)) +
          geom_density() +
          geom_vline(xintercept=quantile(collect_2, probs = c(0.111,0.889)),
                     linetype="dotted") +
          xlab("BMA PDF for Fully Discretised")
```

```
### Comparisons ###

# Plot final graphs
grid.arrange(plot1, plot2, ncol=2)
```



```
print(paste("Maximum Log Likelihood for the Standard Method BDA is", sum))
```

```
## [1] "Maximum Log Likelihood for the Standard Method BDA is -53.5719945269859"
```

```
w_saved # Weights for Standard Method BMA
```

```
## [1] 0.1408354 0.2799299 0.3239744 0.2552603
```

```
quantile(collect, probs = c(0.111,0.889)) # 77.8% Symmetric Prediction Interval
```

```
##    11.1%    88.9%
## 3.094391 8.082349
```

```r
print(paste("Maximum Log Likelihood for the Fully Discretised BDA is", sum_2))
```

```
## [1] "Maximum Log Likelihood for the Fully Discretised BDA is -54.303538762206"
```

```r
w # Weights for Fully Discretised BMA
```

```
## [1] 0.1399465 0.2797674 0.3247332 0.2555529
```

```r
quantile(collect_2, probs = c(0.111,0.889)) # 77.8% Symmetric Prediction Interval
```

```
##    11.1%    88.9%
## 3.139986 8.008944
```

— END OF DOCUMENT —