ENG 107

Problem Set 3: Analytical Methods, BMC, and Grid Methods

Due: Friday, February 7

Hugh Shields

2: PDF of Usable Fuel

Here, we only know the fuel sensor reading is 34 liters and that the sensor error can be approximated as following a Gaussian distribution with a standard deviation of 20 liters. Thus, we pick the fuel sensor reading as our mean value and expect that the distribution of usable fuel F is

$$F \sim \mathcal{N}(\mu = 34, \sigma = 20),$$

and we plot the PDF of this distribution below.

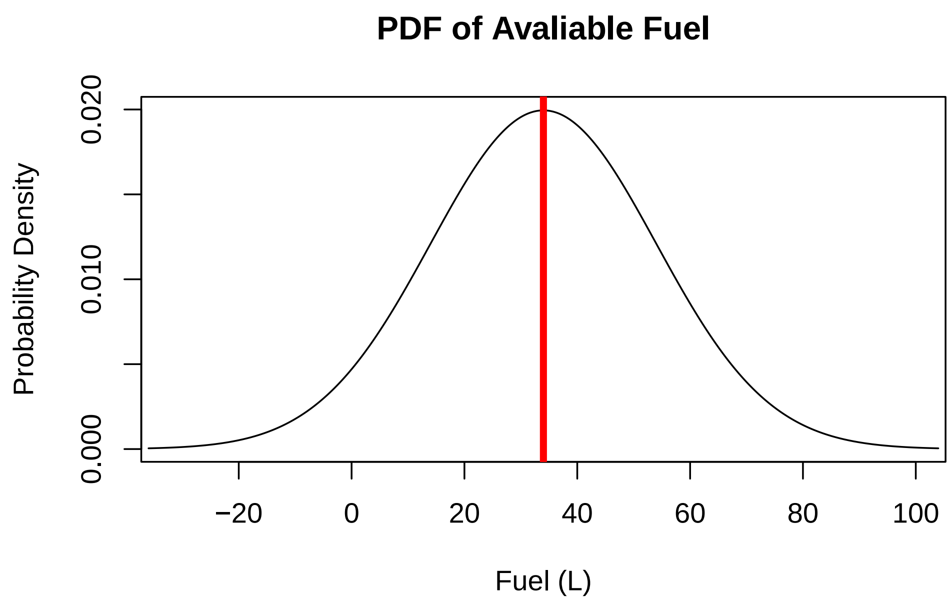

Figure 1: Probability distribution function of usable fuel. Note that the mean $\mu = 34$ L, is marked in red.

We see that the expected value of available fuel, $\mathbb{E}(F)$ is just the mean value of the distribution (34 L), and because the normal distribution is symmetric, the most likely value of available fuel (mode) is also just 34 L. The probability of negative fuel in the tank is just the integral of the distribution in Figure 1 from $-\infty$ to 0, which we calculate to be 4.46% using R's `pnorm` function. The estimate of the error in sensor reading being normally distributed makes sense, but a negative value of fuel is nonphysical.

3: Proper Prior

Because we know that the tank can hold between 0 and 182 L, so we chose a prior that is uniformly distributed between 0 and 182. In this prior, the probability of fuel measurements being less than 0 or greater than 182 L is zero.

Other distributions could be used in this case, but we choose uniform because of its simplicity in capturing the required restraints, operating under the assumption that any fuel tank level is equally likely. If we had more information (e.g. the fuel person usually fills the tank to half capacity), we could use a more informative prior.

4: Grid-Based Method

Here, we construct a vector from $[-50+0, 182+50]$ in 0.01 increments. We then build our prior by setting all values on the interval $[0, 182]$ equal to the normalized uniform distribution (in this case, every value is $\frac{182}{0.01}+1$) and all values outside the interval to 0. We then generate our likelihood using the distribution from Section 2. Multiplying these values yields the posterior, which we normalize and plot over the distribution from Section 2 (see Figure 2).

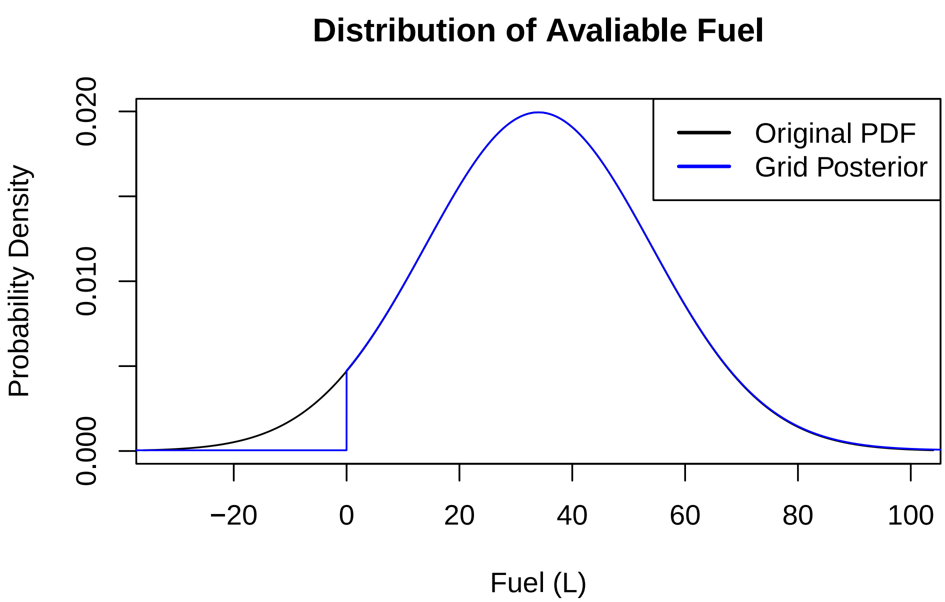

Figure 2: PDF from Figure 1 with the grid generated posterior overplotted in blue.

We see that our posterior now excludes negative numbers (the probability of negative fuel is zero), and the probability of positive values in the distribution is slightly raised. Thus, we a truncated normal distribution.

We here choose an evenly spaced grid with an interval of 0.01 because it samples the distribution well in the area of interest. For a much tighter distribution, an uneven grid spacing might be appropriate. We here assume that the posterior does not have any special features in between grid points.

5: Bayesian Monte Carlo

In this case, we draw our samples from the prior $\sim \mathcal{U}(0, 128)$ and use those samples to compute likelihood values. We then compute our weights vector by normalizing the likelihood vector by its sum. Finally, we redraw samples from the prior sample vector (with replacement) with probability given by the weighting vector. The results are then plotted with a histogram and shown alongside the original PDF in Figure 3.

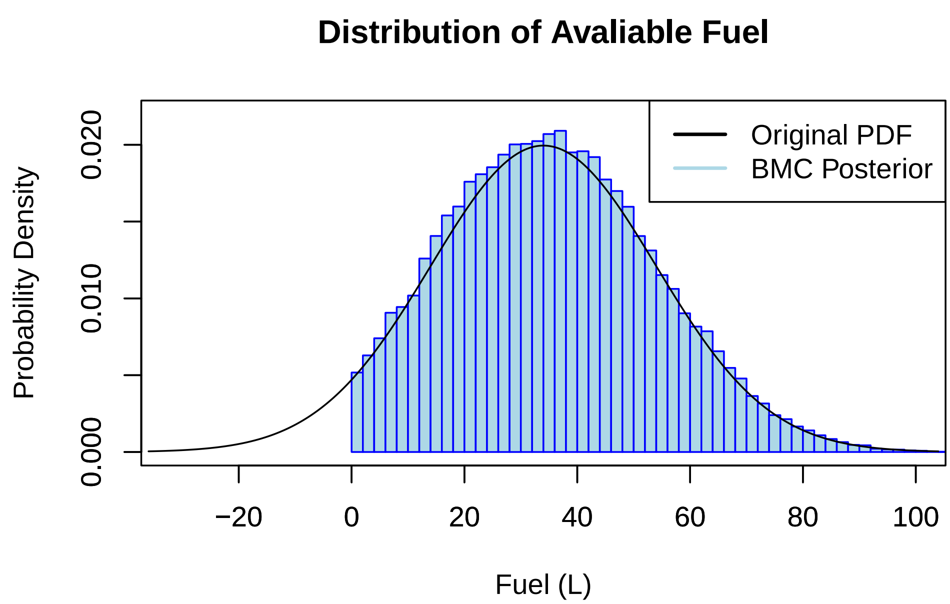

Figure 3: Here, we show a histogram of the Bayes Monte Carlo computed posterior with the PDF from Figure 1 overplotted.

Figure 3 shows a very similar result to Figure 2: The issue of any probability density in the negative part of of the fuel range has been solved and the probability across the positive range has been increased slightly.

In this case, we choose 10^5 samples, as this produces a relatively smooth distribution with 50 bins. A larger sample size would have better approximated the true distribution (a truncated normal), reducing uncertainty in the distribution.

6: Kalman Filtering

In this context, a Kalman Filter could be useful in updating the fuel distribution over time, given a time series of readings from the fuel gauge. To do so, we assume a constant rate of fuel consumption C,

$$C \sim \mathcal{N}\left(\mu = \frac{18}{60}, \sigma = \frac{2}{60}\right),$$

where we have converted the mean and standard deviation computed from liters per hour to liters per minute. Given this rate, we can model the expected change in fuel

$$\delta F = C \times \Delta t,$$

where Δt is the time elapsed in minutes from the last reading. We can then subtract this from the previously fuel distribution (which initially will just be the distribution from Section 2):

$$F_{\text{modeled}} = F_{\text{previous}} - \delta F$$

Given the properties of Gaussian and assuming F_{previous} and δC are uncorrelated, the distribution F_{modeled} can be analytically computed as

$$F_{\text{modeled}} \sim \mathcal{N}\left(\mu = \mu_{F_{\text{previous}}} - \mu_{\delta F}, \sigma = \sqrt{\sigma^2_{F_{\text{previous}}} + \sigma^2_{\delta F}}\right)$$

Then, F_{modeled} is the distribution that we update, using the new observed fuel distribution F_{observed}, with Kalman filtering. In other words, we will have that

$$\mu_{\text{updated}} = \mu_{F_{\text{modeled}}} + K\left[\mu_{F_{\text{observed}}} - \mu_{F_{\text{modeled}}}\right]$$

and

$$\sigma^2_{\text{updated}} = \sigma^2_{F_{\text{modeled}}} - K\,\sigma^2_{F_{\text{modeled}}}$$

where

$$K = \frac{\sigma^2_{F_{\text{modeled}}}}{\sigma^2_{F_{\text{observed}}} + \sigma^2_{F_{\text{modeled}}}}.$$

In the case of the next update, F_{updated} becomes F_{previous}.

Using Kalman filtering in this case requires **three key assumptions**:

1. The rate of fuel consumption is constant in time (i.e. the fuel will be consumed linearly). This is a simplification, as the period of time when the aircraft is descending likely requires less fuel, high wind areas require more fuel, etc.

2. All of the distributions considered are Gaussian. As we elaborated before, this can lead to non-zero probabilities of negative fuel, which is non-physical. Additionally, this means that Kalman filtering will become less accurate as the fuel in the tank gets lower, which is obviously not ideal, as that is the point in the flight in which accurate fuel measurements would be most important.

3. The distributions of fuel level and fuel consumption are uncorrelated. Again, this is a simplification, as in real aircraft there is a higher fuel consumption when the plane is carrying more fuel.

7: Estimated Available Flight Time

Here, we chose to use a Monte Carlo approach to estimate flight times. To do so, we draw n samples from our posterior distribution of usable fuel from Sections 4 and 5, which is just a truncated normal distribution with mean $\mu = 34$ L, standard deviation $\sigma = 20$ L and bounds of $[0, 128]$ L. We implement this sampling in code using the `truncnorm` package. We then draw n samples from our distribution of fuel consumption C,

$$C \sim \mathcal{N}\left(\mu = \frac{18}{60}, \sigma = \frac{2}{60}\right),$$

where we have converted the mean and standard deviation computed from liters per hour to liters per minute. Finally, we simply divide the available fuel F by the fuel consumption rate C to get our distribution of time (in minutes) until the plane runs out of fuel.

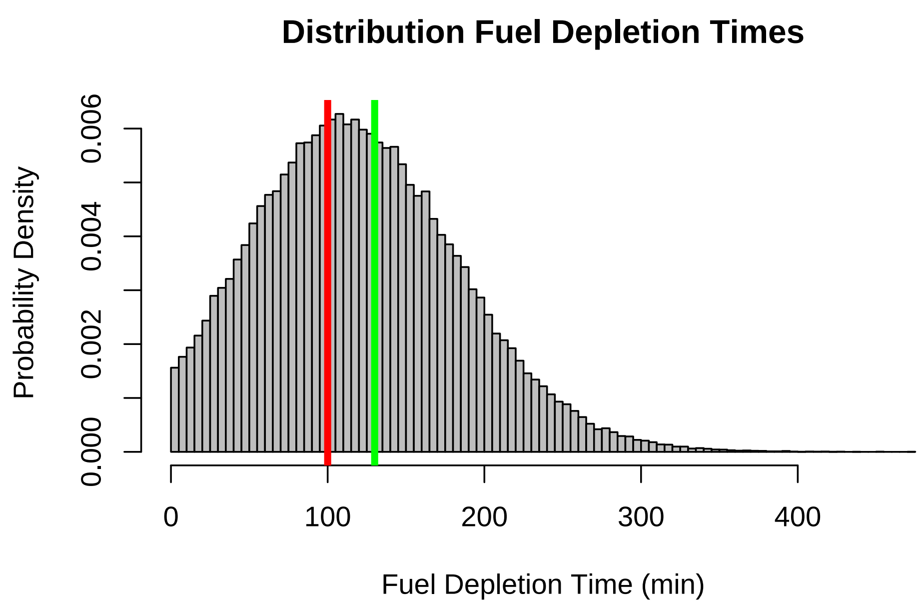

Figure 4: Here, we show a histogram of the Monte Carlo sampling of time of fuel depletion. The green line marks 130 minutes, and the red line marks 100 minutes.

a. To determine the probability of arriving at an airport 100 minutes away with 30 minute reserve, we compute the number of samples greater than or equal to 130 minutes and divide by the total number of samples. We can think of this as the area of the histogram right of the green line in Figure 4 divided by the total histogram area. In this case, we compute the probability as 42.34%.

b. To determine the probability running out of fuel before arriving at an airport 100 minutes away, we compute the number of samples less than 100 minutes and divide by the total number of samples. We can think of this as the area of the histogram left of the red line in Figure 4 divided by the total histogram area. In this case, we compute the probability as 39.37%.

In both cases, we have assumed that the plane has no capacity to glide (i.e. the flight is over when the fuel tank is empty), the rate of fuel consumption is constant in time, and the fuel consumption is not correlated with fuel volume.

We choose a sample size of 10^5 once more, as it produces a relatively smooth histogram. Running the simulation for more samples would allow us to better sample the tails and produce more precise probability estimates.

A Code

See `ps3_shields.R` below. The analysis is reproducable, as a seed is set in the code and all figures and output statistics are generated by sourcing it. This file can be found at https://github.com/hughshields/ENG_107.

```
###########################################
##  file: ps3_shields.R
##  ENG 107 Problem Set 3 Solutions
#####################################################
##  author: Hugh Shields
##  copyright by the author
##  distributed under the GNU general public license
##  https://www.gnu.org/licenses/gpl.html
##  no warranty (see license details at the link above)
#################################################
# version 1: last changes: Feb. 7, 2025 by Hugh Shields
# contact: nikolaus.h.shields.gr@dartmouth.edu
#################################################
# citations:
# - R help files accessed through R-studio for syntax
# - discussions during ENG 107 class time
# - An introduction to R (2010) by Longhow Lam
# - example codes from ENG 107 Canvas
# - truncnorm package documentation
#   (https://cran.r-project.org/web/packages/truncnorm/truncnorm.pdf)
###################################################
# how to run:
# - save the file in a directory
# - go to the directory with this file
# - open R
# - type 'install.packages("truncnorm")'
# - type 'source(ps3_shields.R)'
# - read the printed results in the console and
#   open the new pdf file to analyze the results
###########################################

# Clear any existing variables and plots.
rm(list = ls())
graphics.off()

# define a seed for reproducibility
set.seed(0)
```

```
# 2. plot PDF of useable fuel
pdf(file="PDF_of_usable_fuel.pdf", 6,4)
fuel = seq(-36, 104, length=10000)
fuel.mean = 34
fuel.sd = 20
prob = dnorm(fuel, mean=fuel.mean, sd=fuel.sd)
plot(fuel, prob, type="l", lwd=1, xlab="Fuel (L)", ylab = "Probability Density",
     main = "PDF of Avaliable Fuel", xlim = c(-32, 100))
abline(v=34,col="red",lty=1,lwd=4)
dev.off()

# print the probability of negative fuel
neg_prob = pnorm(0, mean = fuel.mean, sd = fuel.sd)
print(sprintf("2. The probability of negative fuel in the tank is %.2f%%", neg_prob*100)

# 4. Grid method to determine posterior
fuel.min = 0
fuel.max = 182
grid = seq(fuel.min-50, fuel.max+50, by = 0.01)
# create a uniformly distributed prior between [0,128] and normalize to sum to 1
prior = ifelse(grid < 0 | grid > 128, 0, 1 / ((128/0.01)+1))
# specify likelihood from above
likelihood = dnorm(grid, mean=fuel.mean, sd=fuel.sd)
# generate normalized posterior
posterior = (prior * likelihood)/sum(prior * likelihood)

# plot grid posterior over original PDF
pdf(file="grid_posterior.pdf", 6,4)
plot(fuel, prob, type="l", lwd=1, xlab="Fuel (L)", ylab = "Probability Density",
     main = "Distribution of Avaliable Fuel", xlim = c(-32, 100))
par(new = TRUE)
plot(grid, posterior, type="l", col="blue", lwd=1, axes = FALSE, xlab = "", ylab = "",
     xlim = c(-32, 100))
legend("topright", legend = c("Original PDF", "Grid Posterior"), col = c("black", "blue"
dev.off()

# 5. Bayesian Monte Carlo to determine posterior
# sample N values from the prior
N = 100000
prior_sample = runif(N, min = fuel.min, max = fuel.max)
# compute the likelihood values from the prior sampling
likelihood_values = dnorm(prior_sample, mean = fuel.mean, sd = fuel.sd)
```

```r
weights = likelihood_values/sum(likelihood_values)
# resample from the prior sample using the weights (with replacement)
posterior_samples = sample(prior_sample, size = N, replace = TRUE, prob = weights)

# plot histogram (probability=TRUE shows density) of BMC results and overplot original
pdf(file="BMC_posterior.pdf", 6,4)
hist(posterior_samples, breaks=50, probability=TRUE, xlab="Fuel (L)",
     ylab = "Probability Density", main = "Distribution of Avaliable Fuel",
     col="lightblue", border="blue", xlim=c(-32, 100), ylim=c(0, 0.022))
par(new = TRUE)
plot(fuel, prob, type="l", lwd=1, xlab="", ylab = "", main = "",
     xlim = c(-32, 100), ylim=c(0, 0.022))
legend("topright", legend = c("Original PDF", "BMC Posterior"), col = c("black", "lightb
dev.off()

# 7. Computing distribution of remaining flight time
N = 100000
# install and load truncnorm package to generate samples from the fuel distribution
library(truncnorm)
# sample the available fuel distribution (truncated normal distribution)
fuel_sample = rtruncnorm(N, mean = fuel.mean, sd = fuel.sd, a = fuel.min, b = fuel.max)
# sample the rate of fuel consumption distribution
rate.mean = 18/60
rate.sd = 2/60
rate_sample = rnorm(N, mean=rate.mean, sd=rate.sd)

# compute available flight time
time_sample = fuel_sample/rate_sample

# plot histogram of time when fuel runs out
pdf(file="fuel_depletion_dist.pdf", 6,4)
hist(time_sample, breaks=100, probability=TRUE, xlab="Fuel Depletion Time (min)",
     ylab = "Probability Density", main = "Distribution Fuel Depletion Times",
     col="grey", border="black")
# plot lines at 100 and 130 minutes
abline(v=130,col="green",lty=1,lwd=4)
abline(v=100,col="red",lty=1,lwd=4)
dev.off()

# 7.a compute the probability of arriving at the airport 100 min away with 30
# min of fuel reserve
successes = time_sample[time_sample>=130]
success_rate = length(successes)/N
```

```
print(sprintf("7a. The probability of arriving with 30 minutes of fuel reserve is %.2f%%

# 7.a compute the probability of running out of fuel befor arriving at the
# airport 100 min away
failures = time_sample[time_sample<100]
failure_rate = length(failures)/N
print(sprintf("7b. The probability of running out of fuel is %.2f%%", failure_rate*100))
```