HPCSE II - Exercise 3

Anian Ruoss

March 30, 2019

Part I

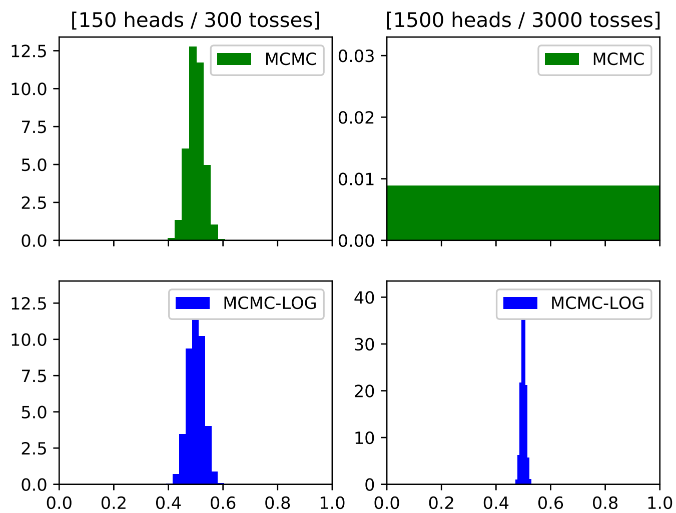

Figure 1: Histogram of the posterior distribution $p\left(H|d\right)$ from the coin toss example sampled via Metropolis-Hastings MCMC. The top row samples are computed in the original scale and the bottom row samples are computed in the logarithmic scale. For a large number of tosses the MCMC in the original scale experiences overflow errors and thus fails to produce valid samples. In the logarithmic scale we observe that increasing the number of tosses (while keeping the number of heads and tails equal) reduces the variance of the posterior distribution as the prior is more certain that the coin is fair.

Part II

Task 1

a)

Listing 1: Korali output when determining the optimal x- and y-coordinates in terms of grass height.

```
[Korali] Starting CMAES. Parameters: 2, Seed: 0x5C9F5947
...
[Korali] Parameter 'x' Value: 4.085200
[Korali] Parameter 'y' Value: 3.747600
[Korali] Total Elapsed Time: 0.002758s
```

Listing 2: Terminal output for *check_cows* with the optimal x- and y-coordinates determined in Listing 1.

```
Searching for cows near (4.085200, 3.747600)...
...
New cows found: 9
Total cows found so far: 9
Herr Kueheli says: "I knew we should look around the spot with the tallest grass!."
Herr Kueheli says: "The rest of the cows should be around here. Lets try these
nearby points:"
[4.08, 3.95]
[4.28, 3.75]
[3.88, 3.75]
[4.08, 3.55]
>>> Time until deadline: 55 minutes. <<<
```

b)

If the grass is high at one point it should be of roughly equal height in its immediate surroundings for a sufficiently "nice" (e.g. continuous) function as we would expect grass growth to be from our experience in the real world. Thus Mr. Kueheli suggests to move roughly 200 meters along each axis (in both directions), as he can only see that far, and check if there are any more cows there. The suggested locations can be found in Listing 2 and the corresponding *check_cows* outputs are presented in listings 3, 4, 5, and 6.

Listing 3: Terminal output for *check_cows* with one of the suggested x- and y-coordinates from in Listing 2.

```
Searching for cows near (4.080000, 3.950000)...
...
New cows found: 15
Total cows found so far: 24
Herr Kueheli says: "That was good, but we need to find them faster!."
>>> Time until deadline: 50 minutes. <<<
```

Listing 4: Terminal output for *check_cows* with one of the suggested x- and y-coordinates from in Listing 2.

```
Searching for cows near (4.280000, 3.750000)...
New cows found: 0
Total cows found so far: 24
Herr Kueheli says: "That was not good enough, perhaps this strategy does not really
                   work."
>>> Time until deadline: 45 minutes. <<<
```

Listing 5: Terminal output for *check_cows* with one of the suggested x- and y-coordinates from in Listing 2.

```
Searching for cows near (3.880000, 3.750000)...
...
New cows found: 1
Total cows found so far: 25
Herr Kueheli says: "That was not good enough, perhaps this strategy does not really
                    work."
>>> Time until deadline: 40 minutes. <<<
```

Listing 6: Terminal output for *check_cows* with one of the suggested x- and y-coordinates from in Listing 2.

```
Searching for cows near (4.080000, 3.550000)...
New cows found: 0
Total cows found so far: 25
Herr Kueheli says: "That was not good enough, perhaps this strategy does not really
                    work."
>>> Time until deadline: 35 minutes. <<<
```

We found another 16 cows which is roughly to be expected as cows tend to be where the grass is high which is roughly around the maximum as outlined above. Of course, we cannot expect all cows to be concentrated around this area as then they would eat all the grass and move on to other places. For this reason we assume the cows to be distributed around different areas of Mr. Kueheli's farm.

c)

We use Korali's TMCMC sampling engine to gain a better understanding of the grass growth at the farm and display the results in Figure 2.

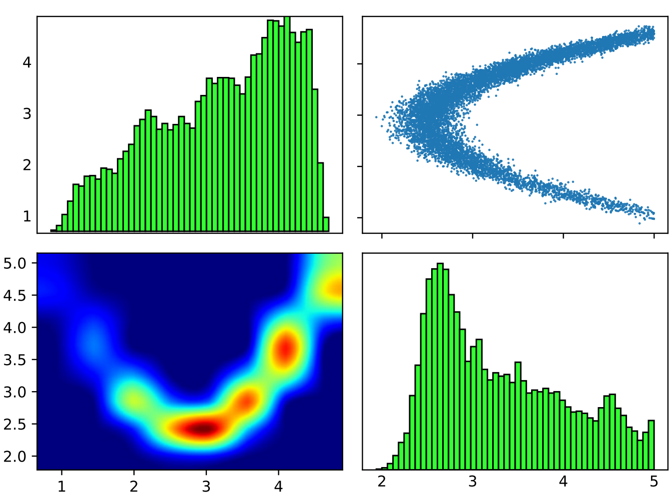

Figure 2: Visualization of the grass growth distribution. We oberserve maxima at roughly $(4.08, 3.75)$ (as obtained in Listing 1) and $(3, 2.5)$.

With this information we check for cows at the maximum $(3, 2.5)$ and in its immediate surroundings (as we can see that the grass is also fairly high there) and we present the terminal outputs in listings 7, 8, and 9.

Listing 7: Terminal output for *check_cows* for the coordinates of the maximum observed in Figure 2.

```
Searching for cows near (3.000000, 2.500000)...
...
New cows found: 90
Total cows found so far: 115
Herr Kueheli says: "Great, this strategy is really effective!."
>>> Time until deadline: 30 minutes. <<<
```

Listing 8: Terminal output for *check_cows* for a point in the neighborhood of the maximum observed in Figure 2.

```
Searching for cows near (3.200000, 2.500000)...
...
New cows found: 32
Total cows found so far: 147
Herr Kueheli says: "Great, this strategy is really effective!."
>>> Time until deadline: 25 minutes. <<<
```

Listing 9: Terminal output for *check_cows* for a point in the neighborhood of the maximum observed in Figure 2.

```
Searching for cows near (2.800000, 2.500000)...
...
New cows found: 37
Total cows found so far: 184
Herr Kueheli says: "Thanks so much, you and Korali helped me find all my cows in
                    time!."
```

The initial strategy relied on the fact that enough cows could be found around the point of maximum grass height. However, the maximum only gives us a point estimate and no information about the underlying distribution of the grass growth (and thus no information about the grass height in the neighborhood of the global maximum). Thus by sampling the distribution, we were able to locate lager areas of high grass and correspondingly also more cows. In this concrete case, we observe that the global maximum has a relatively small neighborhood of high grass whereas other local maxima have larger areas of high grass surrounding them. Naturally, we expect more cows in larger areas of high grass than in a small area with very high grass as cows tend to value some personal space[1]. Of course, we don't know this before looking at the underlying distribution so the initial strategy is not the most efficient (of course, this assumes the we can actually sample from the distribution).

d)

We can use Korali to investigate the posterior distribution of the parameters pH and mm given the grass growth and suitable priors. Naturally we model the experts' clues as prior distributions of the pH and the rain volume. More concretely, we set $p(\text{pH}) \sim \mathcal{U}(4, 9)$ and $p(\text{mm}) \sim \mathcal{N}(90, 20)$. Korali requires us to set bounds for the Gaussian prior on mm. We choose the lower bound of 0 (as

[1] This has been verified via personal communication with a large sample of cows.

we cannot have negative rain) and the upper bound of 180 (for symmetry around the mean). Employing Korali's CMA-ES solver we maximize the posterior and we display the results in Listing 10.

Listing 10: Korali output when maximizing the posterior of the pH and mm parameters.

```
[Korali] Starting CMAES. Parameters: 3, Seed: 0x5C9D72B9
...
[Korali] Parameter 'Sigma' Value: 0.144379
[Korali] Parameter 'pH' Value: 7.472116
[Korali] Parameter 'mm' Value: 115.494066
[Korali] Total Elapsed Time: 0.006133s
```

We observe that both parameter values at the maximum of the posterior lie within the optimal range for pumpkins. However, the uncertainty — as quantified by the value of σ — is rather high. For this, reason we sample from the posterior distribution of the parameters with Korali's TMCMC sampling engine to get a better estimate of the soil and we display the results in Figure 3.

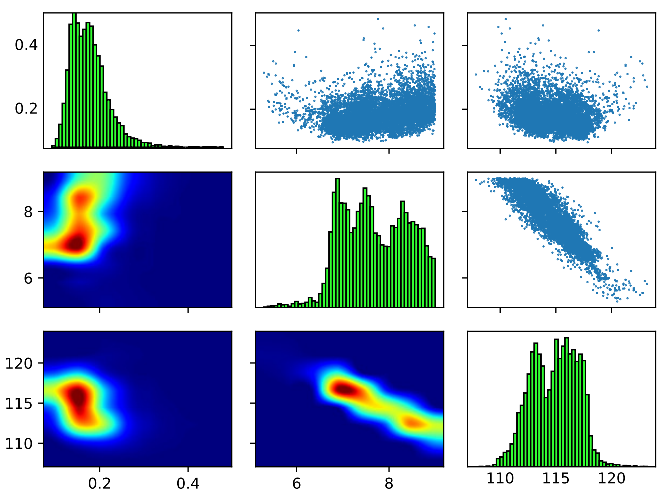

Figure 3: Visualization of the posterior distribution of the parameters. The histogram in the center shows that the pH distribution is skewed to the right of 7.5. The bottom-right histogram shows that the rain volume falls almost entirely within the range $[110, 120]$.

We observe that there is very probably enough rain, i.e. more than $100mm$ per month. However, we also observe that the pH at the maximum of the posterior is atypical in the sense that it does not give us a good estimate of the pH distribution as it is very likely that the actual soil conditions will fall within the range $[7.5, 9]$. Unfortunately, this means that we cannot fully recommend that Ms. Kleineblume plant pumpkins at this point of the year as the soil is probably to basic for her crop.

Task 2

a)

We use the heat2DSolver to model the temperature distribution on the steel sheets. Since we want to determine how many candles were used during the manufacturing we consider three cases with different parameters (x- and y-coordinate for every candle):

- one candle: candle can be applied anywhere on the sheet, i.e. uniformly between 0 and 1 for both axes.
- two candles: the first candle is located on the left half, the second on the right half so we restrict the upper and lower bounds for the x-axis to $[0, 0.5]$ and $[0.5, 1]$ respectively.
- three candles: similar to the two-candle case the third candle is restricted to $[0.5, 1]$ on the x-axis.

Since we are comparing three different models we want to compare the model evidence to determine the best model. For this reason, we employ Korali's TMCMC sampling engine to obtain the log-evidence of the likelihood (since we don't have any prior information on the candle positions) for all three models which we show in listings 11, 12, and 13.

Listing 11: Korali output when sampling the likelihood of the heat distribution given the model with one candle.

```
[Korali] Starting TMCMC. Parameters: 3, Seed: 0x5C9F4376
...
[Korali] Finished. Evidence: -123.834360.
[Korali] Total Time: 43.688758s - Sampling Time: 43.565325s - Engine Time: 0.123391s.
[Korali] Saving results to file: tmcmc.txt.
```

Listing 12: Korali output when sampling the likelihood of the heat distribution given the model with two candles.

```
[Korali] Starting TMCMC. Parameters: 5, Seed: 0x5C9F43D7
...
[Korali] Finished. Evidence: -111.435075.
[Korali] Total Time: 55.662284s - Sampling Time: 55.459889s - Engine Time: 0.202349s.
[Korali] Saving results to file: tmcmc.txt.
```

Listing 13: Korali output when sampling the likeilhood of the heat distribution given the model with three candles.

```
[Korali] Starting TMCMC. Parameters: 7, Seed: 0x5C9F4443
...
[Korali] Finished. Evidence: -88.339730.
[Korali] Total Time: 67.403041s - Sampling Time: 67.192638s - Engine Time: 0.210330s.
[Korali] Saving results to file: tmcmc.txt.
```

We observe that the 3-candles model has the highest evidence and thus it is most likely that 3 workers treated the sheet. To find the most likely positions of every candle we maximize the likelihood with Korali's CMA-ES solver and display the output for the optimal model (and the two other models for completeness) in listings 14, 15, and 16.

Listing 14: Korali output when maximizing the likelihood of the heat distribution given the model with one candle.

```
[Korali] Starting CMAES. Parameters: 3, Seed: 0x5C9F48B3
...
[Korali] Finished - Reason: Function value differences (5.68e-13) < (1.00e-12)
[Korali] Parameter 'Sigma' Value: 6.112970
[Korali] Parameter 'pos_x_1' Value: 0.721352
[Korali] Parameter 'pos_y_1' Value: 0.754621
[Korali] Total Elapsed Time: 12.402701s
```

Listing 15: Korali output when maximizing the likelihood of the heat distribution given the model with two candles.

```
[Korali] Starting CMAES. Parameters: 5, Seed: 0x5C9F490D
...
[Korali] Parameter 'Sigma' Value: 3.074527
[Korali] Parameter 'pos_x_1' Value: 0.282510
[Korali] Parameter 'pos_y_1' Value: 0.250302
[Korali] Parameter 'pos_x_2' Value: 0.736871
[Korali] Parameter 'pos_y_2' Value: 0.770096
[Korali] Total Elapsed Time: 20.062159s
```

Listing 16: Korali output when maximizing the likelihood of the heat distribution given the model with three candles.

```
[Korali] Starting CMAES. Parameters: 7, Seed: 0x5C9F4710
...
[Korali] Parameter 'Sigma' Value: 1.039126
[Korali] Parameter 'pos_x_1' Value: 0.208636
[Korali] Parameter 'pos_y_1' Value: 0.219382
[Korali] Parameter 'pos_x_2' Value: 0.705800
[Korali] Parameter 'pos_y_2' Value: 0.709230
[Korali] Parameter 'pos_x_3' Value: 0.804045
[Korali] Parameter 'pos_y_3' Value: 0.802827
[Korali] Total Elapsed Time: 37.414965s
```

Hence, for our selected model we find the most likely candle positions to be:

- candle one at $[0.208636, 0.219382]$
- candle two at $[0.705800, 0.709230]$
- candle three at $[0.804045, 0.802827]$

b)

We now have prior information on the candle positions that we can incorporate into our model. Hence, we formulate the following prior distributions for the x-coordinates: candle-$1_x \sim \mathcal{N}(0.25, 0.05)$, candle-$2_x \sim \mathcal{N}(0.75, 0.05)$, and candle-$3_x \sim \mathcal{N}(0.75, 0.05)$. We keep the upper and lower bounds from the previous subtask for both axes. Since we have prior information we now investigate the posterior distribution of the parameters with the same solvers as described in the previous subtask and we present the results in listings 17 and 18.

Listing 17: Korali output when sampling the posterior distribution of the candle positions for the 3-candle model.

```
[Korali] Starting TMCMC. Parameters: 7, Seed: 0x5C9F4D00
...
[Korali] Finished. Evidence: -66.664045.
[Korali] Total Time: 62.823597s - Sampling Time: 62.608716s - Engine Time: 0.214820s.
[Korali] Saving results to file: tmcmc.txt.
```

Listing 18: Korali output when maximizing the posterior distribution of the candle positions for the 3-candle model.

```
[Korali] Starting CMAES. Parameters: 7, Seed: 0x5C9F4CC1
...
[Korali] Finished - Reason: Function value differences (8.38e-13) < (1.00e-12)
[Korali] Parameter 'Sigma' Value: 1.039860
[Korali] Parameter 'pos_x_1' Value: 0.209252
[Korali] Parameter 'pos_y_1' Value: 0.220602
[Korali] Parameter 'pos_x_2' Value: 0.803022
[Korali] Parameter 'pos_y_2' Value: 0.804381
[Korali] Parameter 'pos_x_3' Value: 0.708072
[Korali] Parameter 'pos_y_3' Value: 0.707453
[Korali] Total Elapsed Time: 36.914642s
```

We observe that the model evidence is higher than before implying that incorporating the prior information provides a "better fit" with respect to the observed data. Hence, we conclude that applying the candles at the new optimal positions should result in a build quality similar to that of sheet #004392. The new optimal candle positions are:

- candle one at $[0.209252, 0.220602]$
- candle two[2] at $[0.708072, 0.707453]$
- candle three at $[0.803022, 0.804381]$

[2]The numbering of candles two and three is arbitrary and thus we swap the numbers to keep the optimal positions consistent with the previous subtask.