

Replication materials for Krüger, Lerch, Thorarinsdottir and Gneiting (2019), ‘Predictive Inference Based on Markov Chain Monte Carlo Output’

Fabian Krüger

26 March 2019

Motivation

The simulation study and data example in the paper are quite computer intensive. We nevertheless intend to make the paper’s main results (Figures 1-3) replicable. To this end, we offer R functions which can run the simulation study and data example, or possibly (and more realistically) smaller versions thereof. The relevant functions are part of the package `scoringRules` (Jordan et al., 2018). The main replication functions are:

- `run_mcstudy`, which runs the simulation study in Section 4;
- `run_casestudy`, which runs the case study in Section 5.

Both functions create an object that can be plotted via `plot`, as demonstrated below.

Replicating Figures 1 and 2

Running the simulation

The simulation study in the paper is based on 1000 Monte Carlo iterations; computing time is linear in the number of iterations. Therefore, the function `run_mcstudy` has an input argument (`nr_iterations`) which allows to run a small-scale study with fewer iterations (by default, `nr_iterations = 50`). Via another input parameter, `zoom`, users can decide whether to run the large grid of sample sizes in Figure 1 (corresponding to `zoom = FALSE`) or the finer grid in Figure 2 (`zoom = TRUE`). The following code chunk provides an example (produced using an Intel i5 processor on a Linux machine with 8 GB RAM).

```
library(scoringRules)

# Number of Monte Carlo iterations to be done
R <- 50

# Run simulation underlying Figure 1
run1 <- run_mcstudy(nr_iterations = R, zoom = FALSE)

## [1] "2019-03-26 16:41:13 - now starting run_mcstudy"
## [1] "2019-03-26 16:42:22 - now running iteration 10 out of 50"
## [1] "2019-03-26 16:43:35 - now running iteration 20 out of 50"
## [1] "2019-03-26 16:44:52 - now running iteration 30 out of 50"
## [1] "2019-03-26 16:46:14 - now running iteration 40 out of 50"
## [1] "2019-03-26 16:47:36 - now running iteration 50 out of 50"
## [1] "2019-03-26 16:47:44 - now finishing run_mcstudy"
```

The object `run1` is a list containing the simulation results:

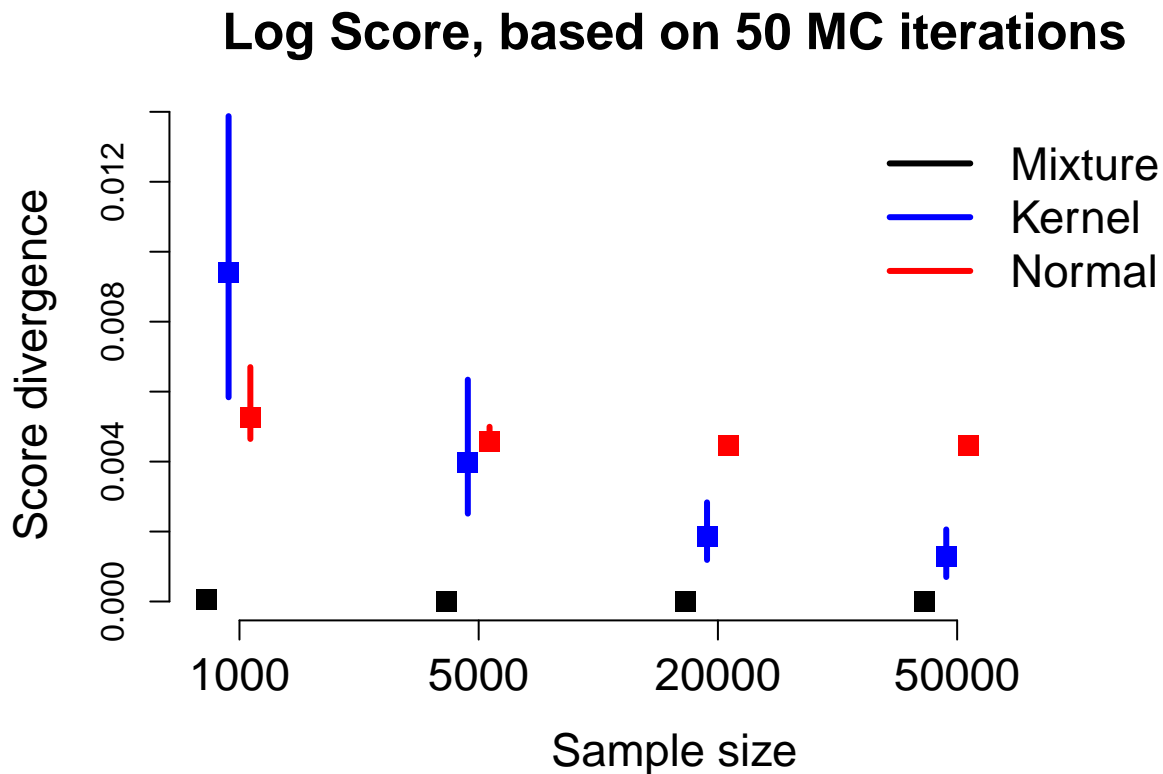
```
str(run1)
```

```
## List of 2
## $ input :List of 6
## ..$ s      : num 2
## ..$ a      : num 0.5
## ..$ n      : num 12
## ..$ nr_iterations: num 50
## ..$ zoom    : logi FALSE
## ..$ random_seed : num 816
## $ output:'data.frame': 1400 obs. of 5 variables:
## ..$ iteration : int [1:1400] 1 1 1 1 1 1 1 1 1 1 ...
## ..$ sample_size: num [1:1400] 1000 1000 1000 1000 1000 1000 1000 1000 5000 5000 5000 ...
## ..$ rule      : chr [1:1400] "logs" "crps" "logs" "crps" ...
## ..$ approx    : chr [1:1400] "norm" "norm" "mixp" "mixp" ...
## ..$ value     : num [1:1400] 4.99e-03 7.32e-04 4.27e-05 5.12e-06 8.80e-03 ...
## - attr(*, "class")= chr "mcstudy"
```

Plotting the results

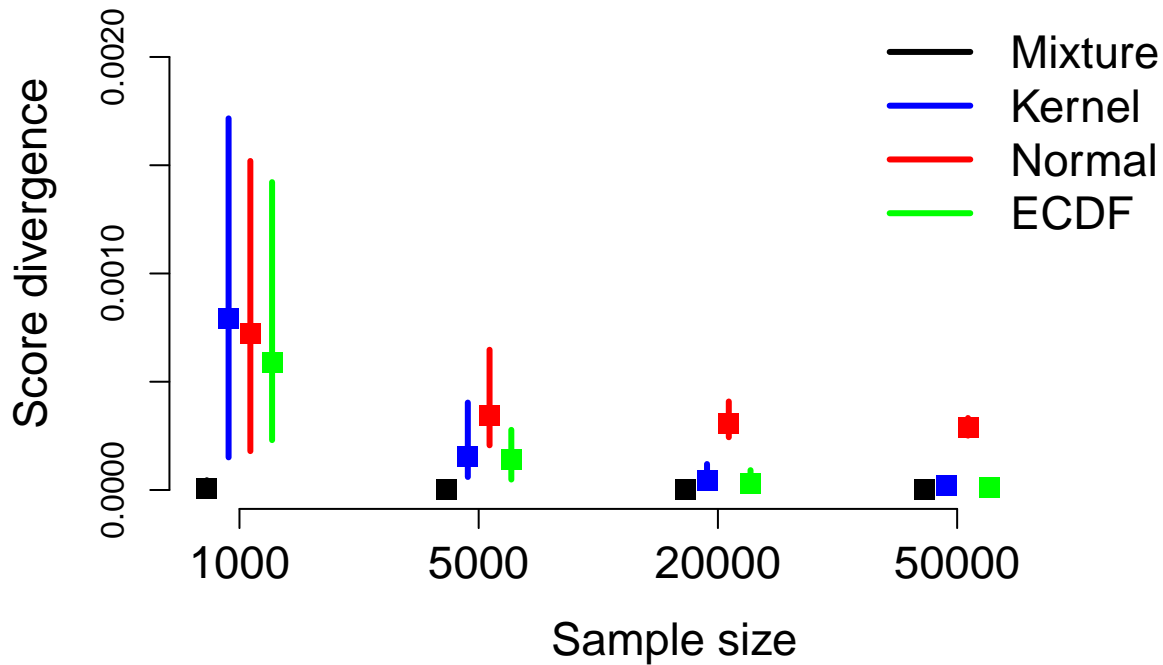
The data in run1 can be plotted via plot, see ?plot.mcstudy. Some examples follow.

```
# Produce a plot that's similar to Figure 1 (left panel)
plot(run1, scoring_rule = "logs")
```



```
# ... Figure 1 (right panel)
plot(run1, scoring_rule = "crps")
```

CRPS, based on 50 MC iterations



The output for Figure 2 is suppressed here for brevity; the associated code is `run2 <- run_mcstudy(nr_iterations = R, zoom = TRUE)`, followed by `plot(run2)`.

Replicating Figure 3

Producing the results

The data example in the paper is a recursive forecasting exercise, i.e., a statistical model is fitted to an expanding sample of data, making a prediction for the next time step in each case. In the paper we consider various sizes of the MCMC sample, as well as 16 parallel runs using different random seeds. The function `replicate_casestudy` allows to run a reduced (or full) version of the case study. Computation time can be reduced by considering a smaller number of parallel chains (the default input is `nr_of_chains = 3`), or by reducing the maximal size of the MCMC chain at each point in time (the default is `max_mcmc_sample_size = 5000`). Furthermore, in contrast to the results in the paper's Section 4, `run_casestudy` uses the computational shortcut described in Appendix A of the paper.

```
data(gdp)
# Run the case study underlying Figure 3
run3 <- run_casestudy(data_df = gdp)
```

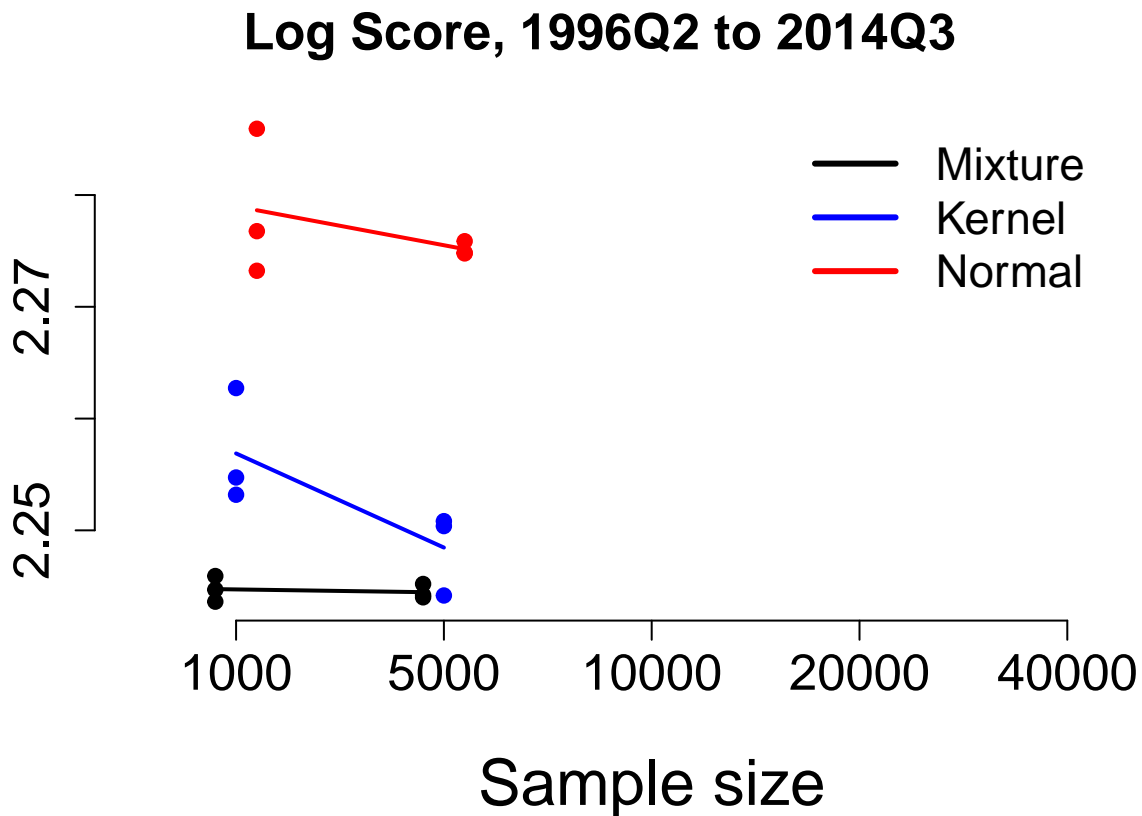
```
## [1] "2019-03-26 16:47:45 - now starting run_casestudy"
## [1] "2019-03-26 16:52:14 - now running date 10 out of 74"
## [1] "2019-03-26 16:57:20 - now running date 20 out of 74"
## [1] "2019-03-26 17:02:22 - now running date 30 out of 74"
```

```
## [1] "2019-03-26 17:08:09 - now running date 40 out of 74"
## [1] "2019-03-26 17:15:00 - now running date 50 out of 74"
## [1] "2019-03-26 17:20:25 - now running date 60 out of 74"
## [1] "2019-03-26 17:26:01 - now running date 70 out of 74"
## [1] "2019-03-26 17:28:52 - now finishing run_casestudy"
```

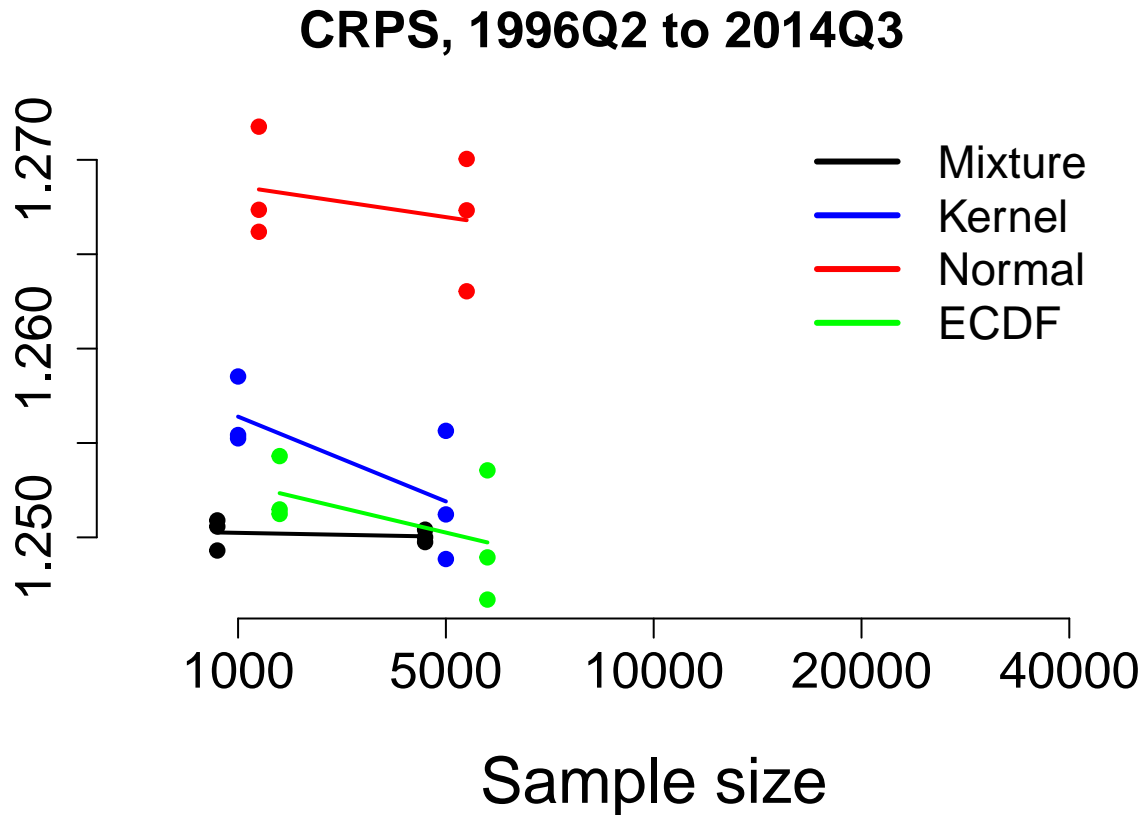
Plotting

Similar to `run_mcstudy`, `run_casestudy` produces a list of data that can be visualized via `plot`:

```
# Produce a plot similar to Figure 3 (left panel)
plot(run3, scoring_rule = "logs")
```



```
# ...Figure 3 (right panel)
plot(run3, scoring_rule = "crps")
```



Since `max_mcmc_sample_size = 5000` (smaller than the value of 40000 used in the paper), the axes of the figures are not filled completely; instead, the sample sizes not computed (10000, 20000 and 40000) are left blank.

References

Jordan, A., Krüger, F. and Lerch, S. (2018): *scoringRules: Scoring Rules for Parametric and Simulated Distribution Forecasts*. R package version 0.9.5, URL: <https://cran.r-project.org/package=scoringRules>.

Krüger, F., Lerch, S., Thorarinsdottir, T. and Gneiting, T. (2019): ‘Predictive Inference Based on Markov Chain Monte Carlo Output’, arXiv preprint, available at <https://arxiv.org/abs/1608.06802>.