Code accompanying Ehm et al (‘Of quantiles and expectiles: Consistent scoring functions, Choquet representations, and forecast rankings’, 2015)

Fabian Krüger

February 5, 2016

Introduction

`murphydiagram` is an add-on package for the R programming language. It contains code and data for the paper by Ehm, Gneiting, Jordan and Krüger (‘Of quantiles and expectiles: Consistent scoring functions, Choquet representations, and forecast rankings’, JRSS-B, 2015).

R itself is freely available at https://www.r-project.org/. The `murphydiagram` package is now hosted on the official repository for R packages, at https://cran.r-project.org/web/packages/murphydiagram/index.html. To install the package, type

```
install.packages("murphydiagram")
```

The package has the following components:

- Forecast/realization data sets studied in Sections 4.1 and 4.2 of the paper
- Functions to produce the displays that we call Murphy diagrams (see Sections 1 and 3.3 of the paper)
- Functions to compute analytical expressions for the synthetic example in Section 3.3 and Appendix B
- Code for various scoring functions considered in the paper
- A function for the fluctuation test (Giacomini and Rossi, *Journal of Applied Econometrics*, 2010) which we consider in the rejoinder to the paper’s discussion at JRSS-B

Note: The code used to construct our data sets from publicly available sources is not part of the `murphydiagram` package, but can be found in a separate zip folder (`construct_forecasts.zip`), available at https://sites.google.com/site/fk83research/code. The provider websites mentioned in the package manual should be consulted for the official, unprocessed versions of the data sources. Please contact Fabian Krüger (fabian.krueger@h-its.org) for further information.

Replication examples

Here we illustrate how the package can be used to replicate the results in the paper.

Data examples from Section 4

```r
library(murphydiagram)

# Figure 6, top row
data(inflation_mean)
# Left panel
murphydiagram(inflation_mean$spf, inflation_mean$michigan, inflation_mean$rlz,
              labels = c("SPF", "Michigan"))
```

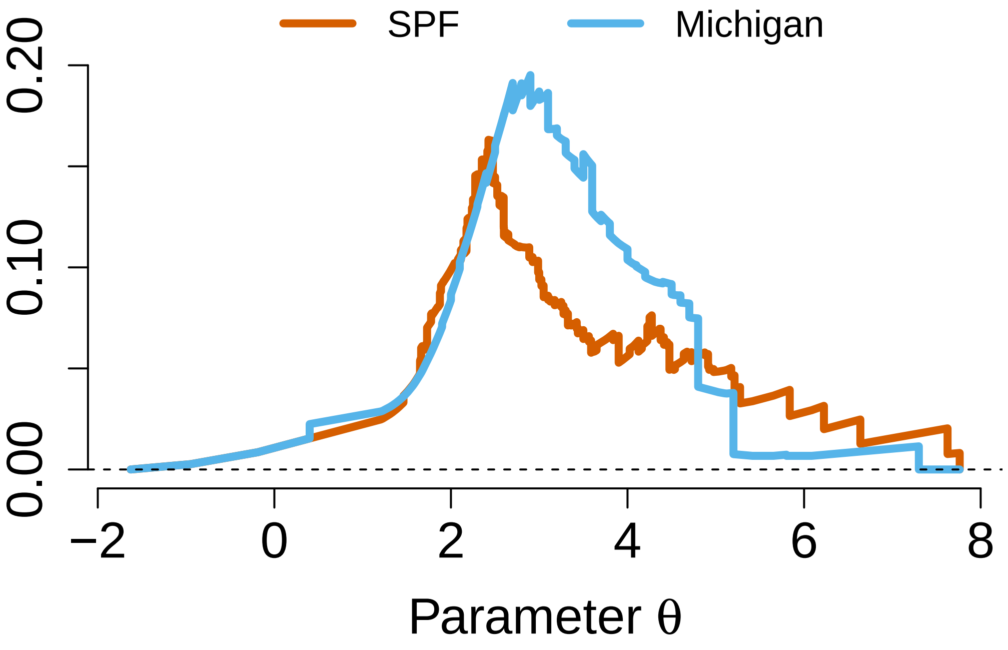

Figure 1: Replicates the top left panel of Figure 6 in the paper.

```r
# Right panel
murphydiagram_diff(inflation_mean$spf, inflation_mean$michigan, inflation_mean$rlz,
                   lag_truncate = 4)
```

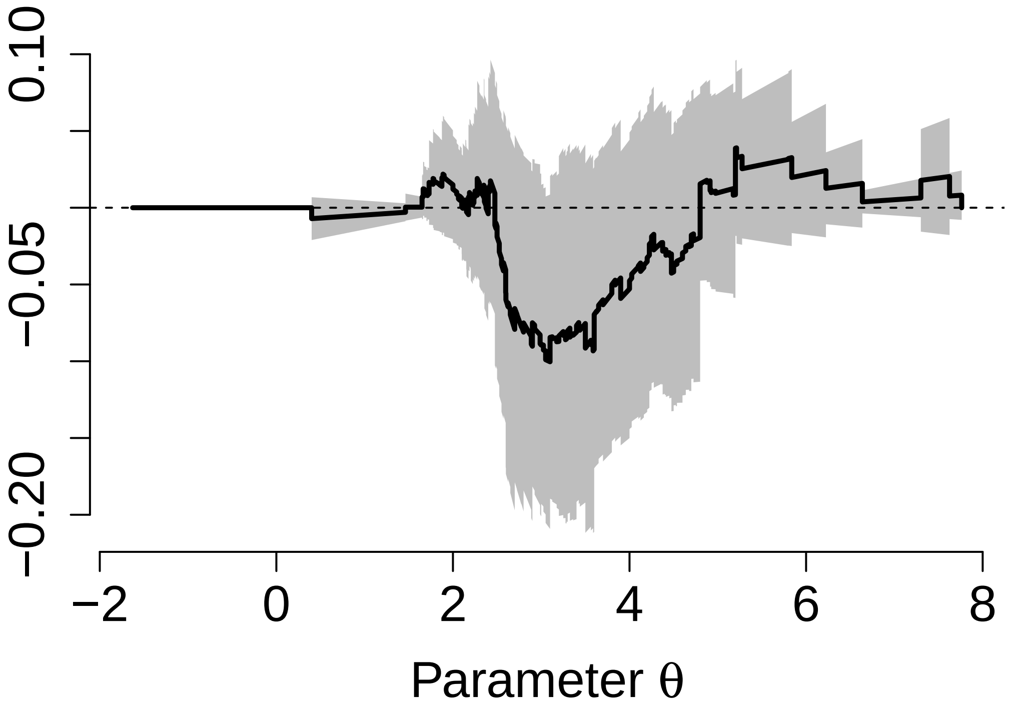

Figure 2: Replicates the top right panel of Figure 6 in the paper.

```r
# Figure 6, middle row
data(recession_probability)
# Left panel
murphydiagram(recession_probability$spf, recession_probability$probit,
              recession_probability$recession, labels = c("SPF", "Probit"))
```

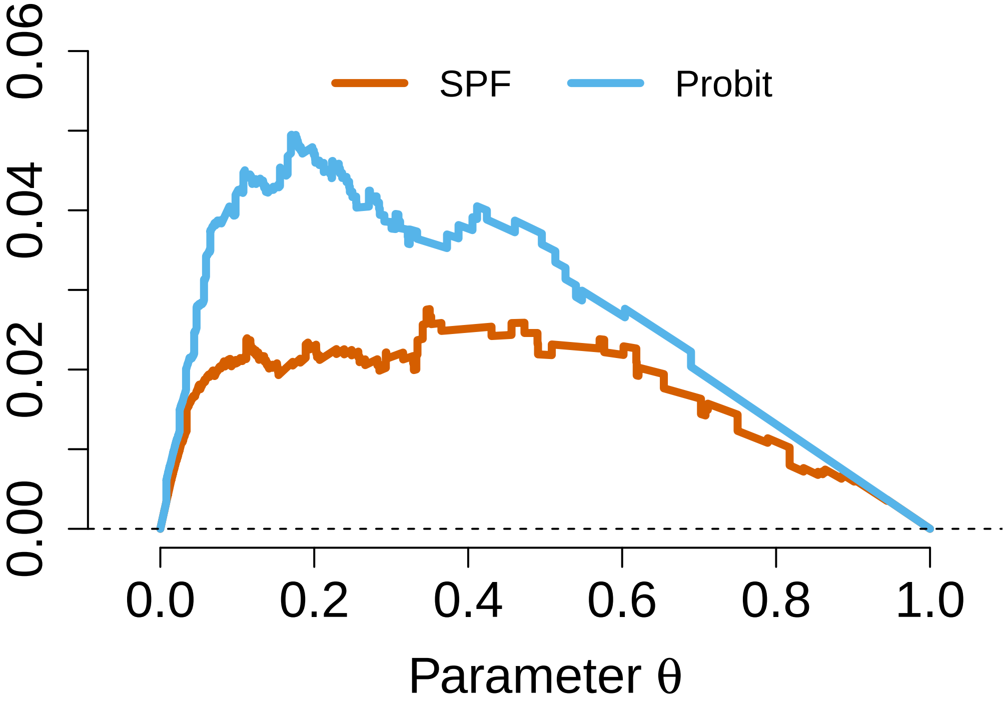

Figure 3: Replicates the middle left panel of Figure 6 in the paper.

```
# Right panel
murphydiagram_diff(recession_probability$spf, recession_probability$probit,
                   recession_probability$recession)
```

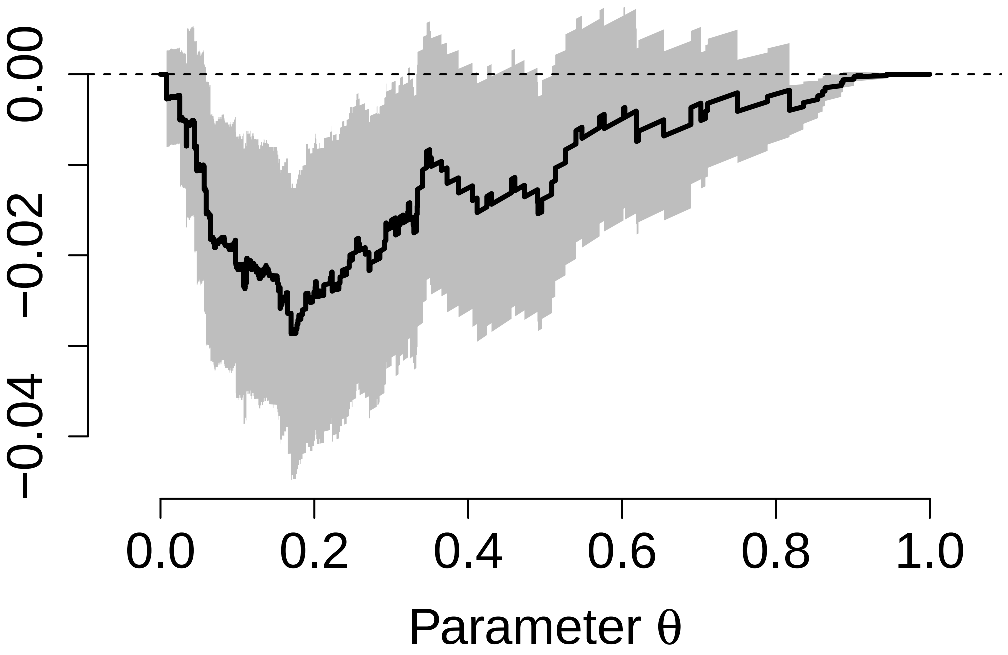

Figure 4: Replicates the middle right panel of Figure 6 in the paper.

Synthetic example from Section 3.3

```r
# Color palette, obtained from http://www.cookbook-r.com/Graphs/Colors_%28ggplot2%29/
cbbPalette <- c("#000000", "#E69F00", "#56B4E9", "#009E73")
cbbPalette <- cbbPalette[c(1, 4, 2, 3)]

# Labeling stuff
forecasters <- c("P", "C", "U", "SR")
names <- c("Perfect", "Climatological", "Unfocused", "Sign-Reversed")
x_label <- expression(paste("Parameter ", theta))

# Figure 2, top left

# Grid for theta
theta_grid1 <- seq(-3, 3, 0.01)
# Expected scores for all forecasters
scores1 <- sapply(forecasters, expected_score_mean, theta = theta_grid1)
# Plot
matplot(x = theta_grid1, y = scores1[, 4:1], type = "l", lty = 1, col = cbbPalette[4:1],
        lwd = 2, bty = "n", xlab = x_label, ylab = expression("Expected Score"))
legend("topright", names, col = cbbPalette, lwd = 2, bty = "n")
```

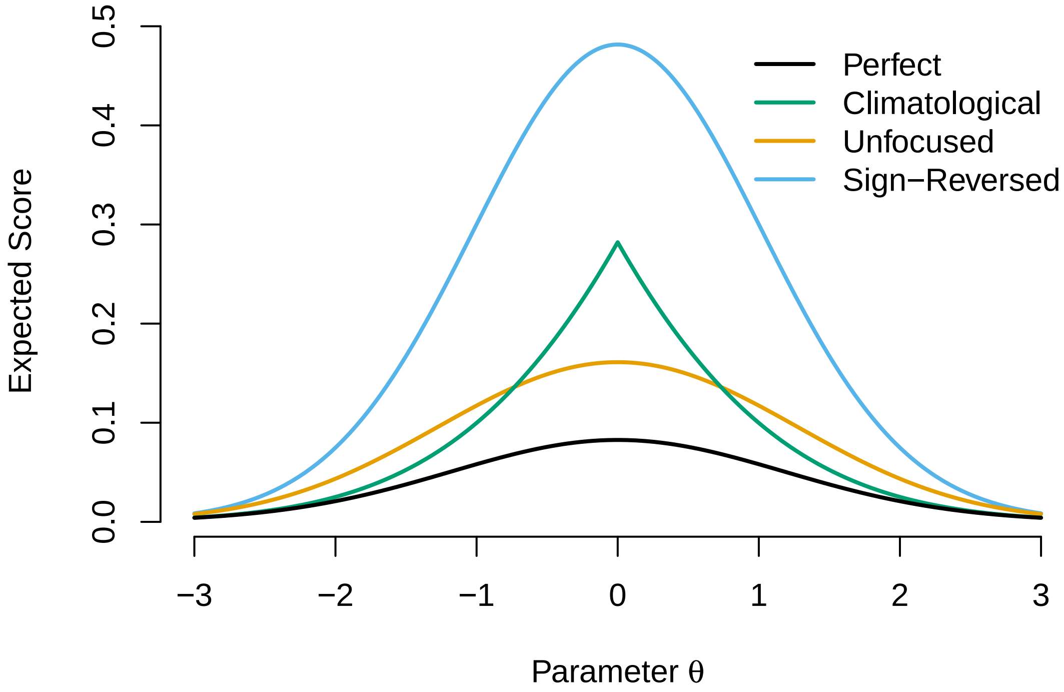

Figure 5: Replicates the top left panel of Figure 2 in the paper.

```r
# Figure 2, bottom left

# Grid for theta
theta_grid2 <- seq(-2, 4, 0.01)
# Expected scores for all forecasters
scores2 <- 10*sapply(forecasters, expected_score_quantile, theta = theta_grid2,
                     alpha = 0.9)
# Plot
matplot(x = theta_grid2, y = scores2[, 4:1], type = "l", lty = 1, col = cbbPalette[4:1],
        lwd = 2, bty = "n", xlab = x_label, ylab = expression("Expected Score x 10"),
        ylim = c(0, 2.5))
legend("topright", names, col = cbbPalette, lwd = 2, bty = "n")
```

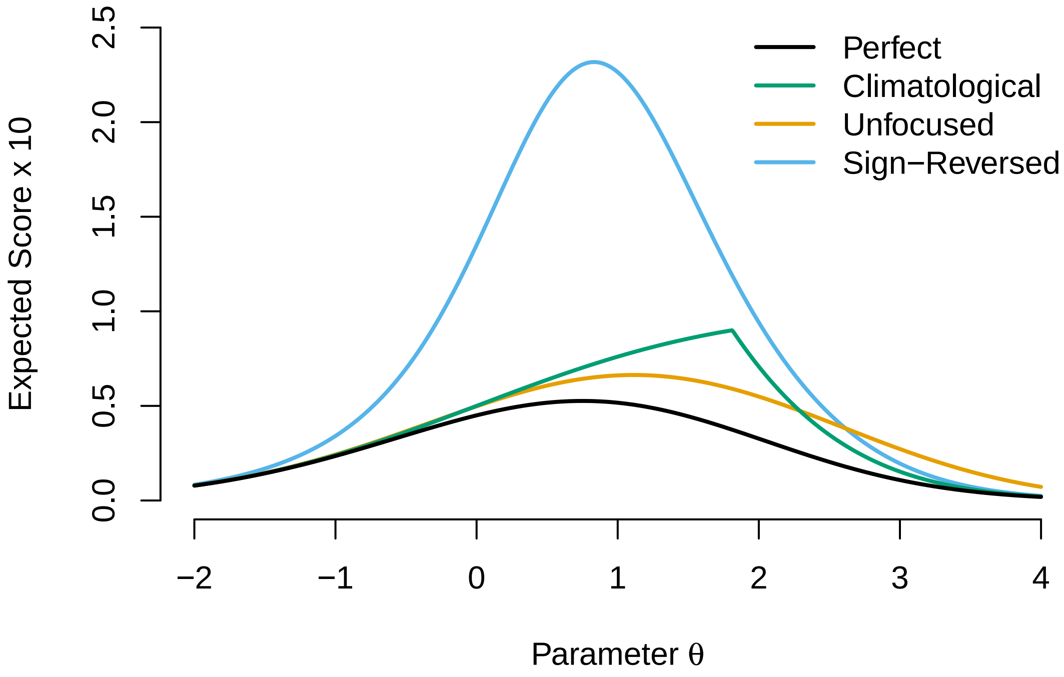

Figure 6: Replicates the bottom left panel of Figure 2 in the paper.