

Supplementary Digital Content - The Performance of Hypotension Prediction Index May Be Overestimated Due to Selection Bias

Johannes Enevoldsen and Simon Tilma Vistisen

Contents

Setup	1
Simulation of the selection problem	1
Create figure 3	3

Setup

```
library(pROC)
library(tidyverse)

# A few extra packages are used to create the simulation figure. We load these in
# the relevant section.

theme_set(theme_minimal(base_size = 12))
```

Simulation of the selection problem

OBS: The simulation is not an attempt to produce realistic data. It only serves to illustrate how the selection problem results in a “skewed” ROC curve with very high specificity.

We make a simple simulation, to show what the data selection described in the model-development paper can do to the performance of a model that uses MAP as a predictor.

The specific parameters of the simulation are chosen to place most data in a relevant interval (55 to 95 mmHg) and to give MAP a modest predictive ability, similar to what is presented by Davies *et al* (<http://doi.org/10.1213/ANE.0000000000004121>) However, whenever we have MAP values above and below 75 mmHg for both samples corresponding to hypotensive- and nonhypotensive events, the effect of the selection bias will be present.

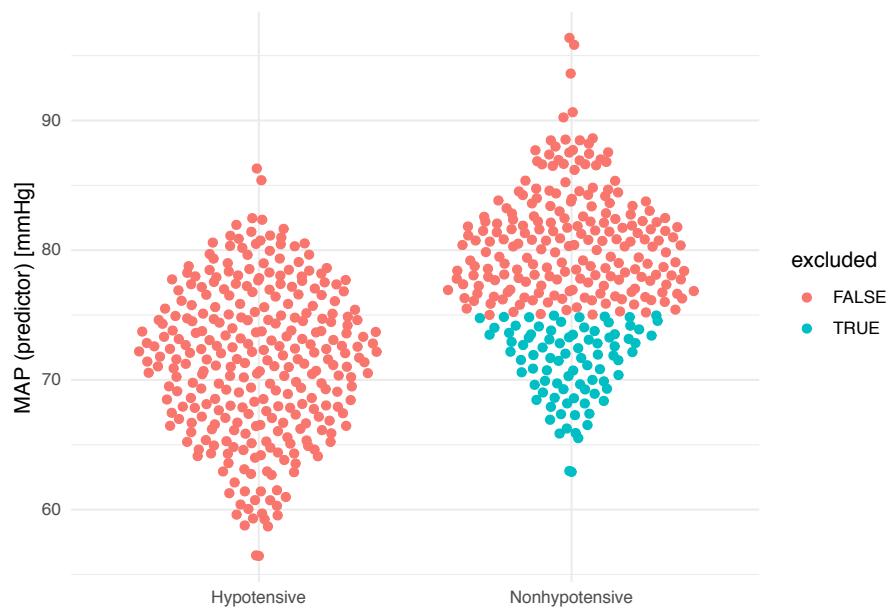
We generate normally distributed data representing the MAP in samples corresponding to hypotensive events and nonhypotensive events. For the hypotension samples, we generate 300 values with a mean of 72 mmHg and a SD of 6 mmHg. For the nonhypotension samples we generate 300 values with a mean of 78 mmHg and a SD of 6 mmHg.

```
set.seed(999)

d <- data.frame(
  Hypotension = rnorm(300, mean = 72, sd = 6),
  Nonhypotension = rnorm(300, mean = 78, sd = 6)) |>
  pivot_longer(everything(), names_to = "outcome",
               values_to = "map_predictor") |>
  mutate(hypo = outcome == "Hypotension",
         excluded = !hypo & map_predictor < 75)
```

The generated data looks like this:

```
ggplot(d, aes(hypo, map_predictor, color = excluded)) +
  ggbeeswarm::geom_quasirandom(size = 2, bandwidth = 0.8) +
  scale_x_discrete(limits = c(TRUE, FALSE), name = "", labels = c("Hypotensive", "Nonhypotensive")) +
  labs(y = "MAP (predictor) [mmHg]")
```



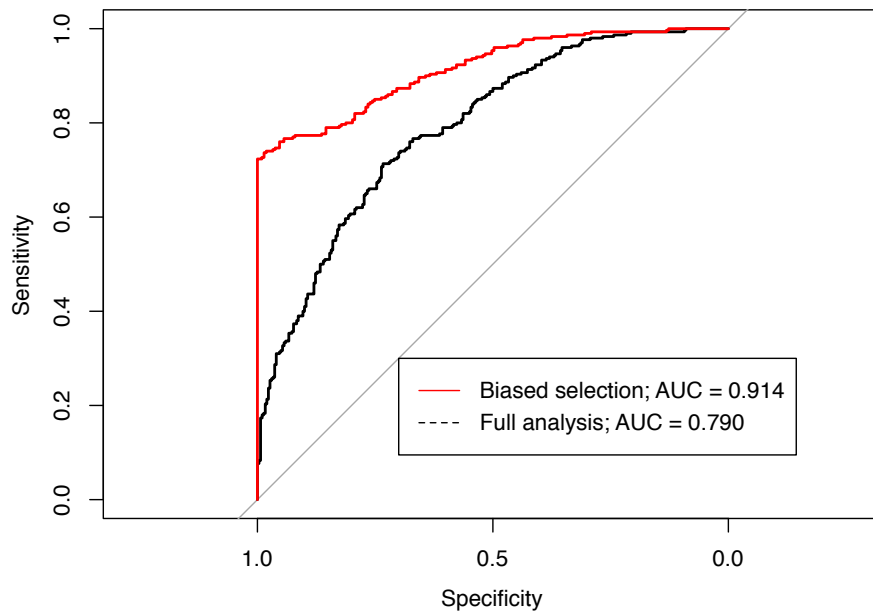
Now we can make the ROC analyses.

```
# With all data
roc_analysis_full <- roc(hypo~map_predictor, data = d)
# Include only data that is either event or non-event per model-development definition
roc_analysis_biased <- roc(hypo~map_predictor, data = filter(d, !excluded))
```

```

plot(roc_analysis_full)
plot(roc_analysis_biased, col = "red", add = TRUE)
legend(
  x = 0.7, y = 0.3,
  legend = c(
    sprintf("Biased selection; AUC = %.3f", auc(roc_analysis_biased)),
    sprintf("Full analysis; AUC = %.3f", auc(roc_analysis_full))
  ),
  lty = c(1, 2, 1),
  col = c("red", "black", "black")
)

```



The ROC curve from the biased data selection shows a sharp drop in specificity (when MAPnow is > 75 mmHg) compared to that in the model-development paper (fig. 1). This difference can be explained by a combination of two effects. First, in the model development, all MAP values in a 30-minute period had to be > 75 mmHg; due to the variation in MAP over 30 minutes, any specific measurement (here, the midpoint) is unlikely to be very close to 75 mmHg. Second, although HPI may be driven predominantly by MAP, other predictors influence HPI as well.

Create figure 3

The remaining code combines plots of MAP values and the ROC analyses into figure 3.

```

library(ggplot2)
library(patchwork)

# Set plot theme
# Plot theme

```

```

# Theme for plot
theme_set(
  theme_minimal(base_size = 11) +
  theme(
    plot.title = ggtext::element_textbox_simple(
      size = rel(1),
      vjust = 0,
      minheight = unit(3, "lines"), width = unit(2.8, "inches")
    ),
    axis.title = element_text(size = rel(1)),
    axis.text = ggplot2::element_text(size = ggplot2::rel(0.9),
      color = "#333333"),
    panel.grid.major = ggplot2::element_line(
      color = "#454545",
      size = 0.3,
      linetype = "dotted"
    ),
    panel.grid.minor = ggplot2::element_blank()
  )
)

plot_colors <- c("#bb612a", "#2A6EBB")

# First we extract all combinations of sensitivity, specificity and thresholds for
# all ROC analyses
roc_data_full <- as.data.frame(roc_analysis_full[c("sensitivities",
  "specificities",
  "thresholds")]) |>
  arrange(desc(specificities), sensitivities)

roc_data_biased <- as.data.frame(roc_analysis_biased[c("sensitivities",
  "specificities",
  "thresholds")]) |>
  arrange(desc(specificities), sensitivities)

# We extract the rows with thresholds closest to 70, 75 and 80
# This is used for labeling the ROC curves.
roc_label_filter <- function(roc_data) {
  roc_data |>
  filter(
    row_number() == which.min(abs(thresholds - 70)) |
    row_number() == which.min(abs(thresholds - 75)) |
    row_number() == which.min(abs(thresholds - 80))
  )
}

roc_labels_full <- roc_label_filter(roc_data_full)
roc_labels_biased <- roc_label_filter(roc_data_biased)

```

```

# Make the 2 ROC panels
roc_plot_full <- ggplot(roc_data_full, aes(1-specificities, sensitivities)) +
  geom_step() +
  geom_abline(slope = 1, intercept = 0, linetype = 2, color = "grey") +
  geom_label_repel(aes(label = glue::glue("MAP < {round(thresholds, 0)}")),
    parse = TRUE,
    size = 3.3,
    box.padding = 0.1,
    point.size = 0.1,
    segment.color = "#555555",
    position = position_nudge_repel(y = -0.25, x=0.2),
    ylim = c(0.15, 1),
    xlim = c(0, 1.1), # extend to outside plotting area
    direction = "y",
    label.size = NA,
    hjust = 0,
    family = "Helvetica",
    data = roc_labels_full
  ) +
  annotate("label",
    x = 1, y = 0,
    label = sprintf("AUC = %.2f", auc(roc_analysis_full)),
    hjust = 1, vjust = -0,
    label.size = NA,
    family = "Helvetica",
    size = 3.5) +
  labs(x = "1 - specificity", y = "Sensitivity") +
  coord_equal()

roc_plot_biased <- ggplot(roc_data_biased, aes(1-specificities, sensitivities)) +
  geom_step() +
  geom_abline(slope = 1, intercept = 0, linetype = 2, color = "grey") +
  geom_label_repel(aes(label = glue::glue("MAP < {round(thresholds, 0)}")),
    parse = TRUE,
    size = 3.3,
    box.padding = 0.1,
    point.size = 0.3,
    segment.color = "#555555",
    position = position_nudge_repel(y = -0.3, x=0.2),
    ylim = c(0.15, 1),
    xlim = c(0, 1.1), # extend to outside plotting area
    direction = "y",
    label.size = NA,
    hjust = 0,
    family = "Helvetica",
    data = roc_labels_biased
  )

```

```

) +
  annotate("label",
    x = 1, y = 0,
    label = sprintf("AUC = %.2f", auc(roc_analysis_biased)),
    hjust = 1, vjust = -0,
    label.size = NA,
    family = "Helvetica",
    size = 3.5) +
  labs(x = "1 - specificity", y = "Sensitivity") +
  coord_equal()

# Make the 2 dot plots
color_scale <- scale_color_manual(values = plot_colors)

map_scale <- scale_y_continuous(breaks = seq(45, 105, by = 10),
  expand = c(0.01,0.01),
  name = "MAP (predictor) [mmHg]")

dot_plot_full <- ggplot(d, aes(outcome, map_predictor, color = outcome)) +
  ggbeeswarm::geom_quasirandom(size = 0.8, bandwidth = 1, show.legend = FALSE) +
  map_scale +
  labs(x = "Event (outcome)",
    title = sprintf("***A** Full analysis (n=%d)", nrow(d))) +
  color_scale +
  theme(panel.grid.major.x = element_blank())

dot_plot_biased <- ggplot(d, aes(outcome, map_predictor, color = outcome,
  # Instead of removing excluded points, we set
  # alpha to 0 for these points.
  # this makes the shapes of the quasirandom
  # scatters identical, so it is clear that we
  # simply exclude MAP values below 75.
  alpha = excluded)) +
  ggbeeswarm::geom_quasirandom(size = 0.8, bandwidth = 1, show.legend = FALSE) +
  map_scale +
  scale_alpha_manual(values = c(1, 0)) +
  labs(x = "Event (outcome)",
    title = sprintf("***B** Samples corresponding to <em>non</em>hypotensive
      events must have a MAP (predictor) ≥ 75 mmHg (n=%d)", nrow(filter(d, !excluded)))) +
  color_scale +
  theme(panel.grid.major.x = element_blank())

# Combine plots
simulation_figure <- (
  (dot_plot_full + dot_plot_biased +
  roc_plot_full + roc_plot_biased) &
  theme(plot.margin = margin(1, 6, 1, 2, "mm"))

```

```
) +  
plot_layout(ncol = 2, byrow = TRUE, heights = c(1, 1))
```

```
ggsave("figure3_simulation.pdf", simulation_figure, device = cairo_pdf,  
width = 16, height = 17, units = "cm", scale = 1)
```

```
ggsave("figure3_simulation.png", simulation_figure, device = ragg::agg_png, bg = "white",  
width = 16, height = 17, units = "cm", scale = 1)
```

simulation_figure

