CV Analyses

Thomas Ward

2021-08-29

```r
library(tidyverse)
library(irr)
library(showtext)
font_add_google("Lato")
showtext_auto()
theme_set(theme_classic(base_family = "Lato"))
```

Load results

```r
results <- read_csv("../data/cv_results.csv", col_types = "ciiiii")
```

Utility functions

Calculate Krippendorff's alpha

Convenience function that can be used with `summarise()`

```r
kripp_alpha <- function(col1, col2) {
    matrix(c(col1, col2), ncol = length(col1), byrow = TRUE) %>%
        irr::kripp.alpha(method = "ordinal") %>%
        pluck("value")
}
```

Calculate Krippendorff's alpha in boostrap

This will be used by dplyr in nested list dataframes:

```r
kripp_alpha_df <- function(df) {
    df %>%
        as.matrix() %>%
        t() %>%
        irr::kripp.alpha(method = "ordinal") %>%
        pluck("value")
}
```

Computer Vision Models Cross-Validated Performance

Below I will calculate the cross-validated performance of the two CV models.

The first is `pgs_combo`, which is the PGS determined by taking the results of the adhesion and appearance networks and calculating the PGS.

The second is `pgs_only`, which is a network that was trained to only classify PGS, not taking into account any of the subcomponents.

```r
per_fold_results  <- results %>%
    select(name, gt, pgs_combo, pgs_only, fold) %>%
    group_by(fold) %>%
    summarise(across(c("pgs_combo", "pgs_only"), list(krippa = ~ kripp_alpha(., gt))))
knitr::kable(per_fold_results, digits = 3, caption = "Per-fold metrics")
```

Table 1: Per-fold metrics

fold	pgs_combo_krippa	pgs_only_krippa
1	0.639	0.737
2	0.716	0.662
3	0.793	0.784
4	0.650	0.555
5	0.806	0.777
6	0.610	0.415
7	0.741	0.477
8	0.558	0.562
9	0.728	0.610
10	0.849	0.811

Now that we have per-fold metrics, we can calculate CV performance metrics, including mean, standard deviation (sd), and standard error (se):

```r
per_fold_results %>%
    select(-fold) %>%
    summarise(
        across(everything(),
            list(mean = mean, sd = sd, se = ~ sd(.) / sqrt(n()))
        )
    ) %>%
    # make into nice table
    pivot_longer(
        everything(),
        names_to = c("user", NA, "statistic", ".value"),
        names_pattern = "pgs_([a-z]+)_(sqerr_)?([a-z]+)_([a-z]+)"
    ) %>%
    # calc 95% intervals using central limit theorem
    rowwise() %>%
    mutate(
        conf.low = mean - 1.96 * se,
        conf.high = mean + 1.96 * se
    ) %>%
    ungroup() %>%
    arrange(statistic, user) %>%
    knitr::kable(
        digits = 2,
        caption = "Cross-Validated Krippendorff's alpha performance"
    )
```

Table 2: Cross-Validated Krippendorff's alpha performance

user	statistic	mean	sd	se	conf.low	conf.high
combo	krippa	0.71	0.09	0.03	0.65	0.77
only	krippa	0.64	0.14	0.04	0.55	0.72

Second surgeon performance

The second surgeon annotated all representative frames in one go. As recommended by Krippendorff, we can calculate the alpha statistic, then use the bootstrap to calculate confidence intervals.

```r
set.seed(1234)
surg2_metrics <- results %>%
    select(gt, pgs_surg2) %>%
    modelr::bootstrap(10000) %>%
    transmute(
        kripa = map_dbl(strap, compose(kripp_alpha_df, as.data.frame))
    ) %>%
    summarise_all(
        list(
            mean = mean,
            low_ci = ~ quantile(., probs = c(0.025)),
            high_ci = ~ quantile(., probs = c(0.975))
        )
    ) %>%
    pivot_longer(everything(), names_to = "statistic")
```

```r
surg2_metrics %>%
    knitr::kable(
        digits = 2,
        caption = "Performance of 2nd Surgeon, CI calculated with the bootstrap"
    )
```

Table 3: Performance of 2nd Surgeon, CI calculated with the bootstrap

statistic	value
mean	0.82
low_ci	0.75
high_ci	0.87

Confusion Matrices

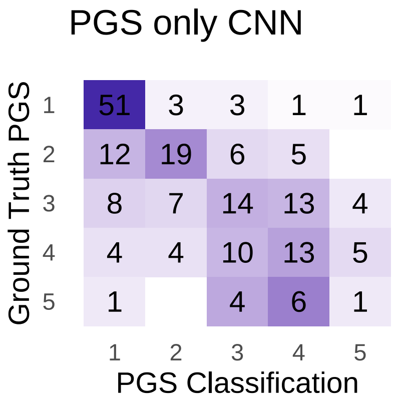
```r
calculate_confusion_matrix <- function(df, groundtruth, prediction) {
    df %>%
        select(gt = {{ groundtruth }}, pred = {{prediction}}) %>%
        count(gt, pred) %>%
        group_by(gt) %>%
        mutate(prop = n / sum(n)) %>%
        ungroup() %>%
        mutate(gt = fct_rev(as.factor(gt)))
}
```

```r
# df and specify groundtruth col and prediction col
plot_confusion_matrix <- function(df, title = "Confusion Matrix") {
    df %>%
        ggplot(aes(pred, gt)) +
        scale_fill_gradient(name = "Proportion\n", low = "white", high = "#000099", limits = c(0, 1)) +
        geom_tile(aes(fill = prop)) +
        geom_text(aes(label = n)) +
        theme_classic() +
        theme(
            axis.ticks = element_blank(),
            axis.line = element_blank()
        ) +
        labs(
            x = "PGS Classification",
            y = "Ground Truth PGS",
            title = title
        )
}
```

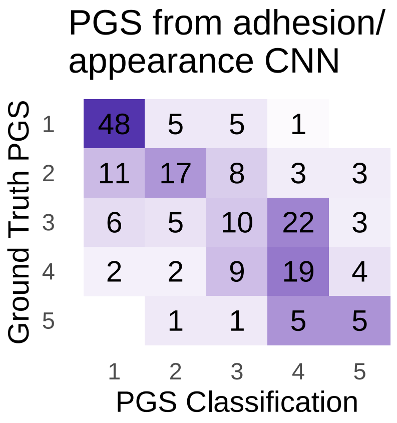
```r
only_results <- calculate_confusion_matrix(results, gt, pgs_only)
combo_results <- calculate_confusion_matrix(results, gt, pgs_combo)
surg2_results <- calculate_confusion_matrix(results, gt, pgs_surg2)
```

```r
only_cf <- plot_confusion_matrix(only_results, "PGS only CNN") +
    theme(legend.position = "none")
combo_cf <- plot_confusion_matrix(
        combo_results,
        "PGS from adhesion/\nappearance CNN"
    ) +
    theme(legend.position = "none")
surg2_cf <- plot_confusion_matrix(surg2_results, "Second Surgeon PGS") +
    theme(legend.position = "none")
```

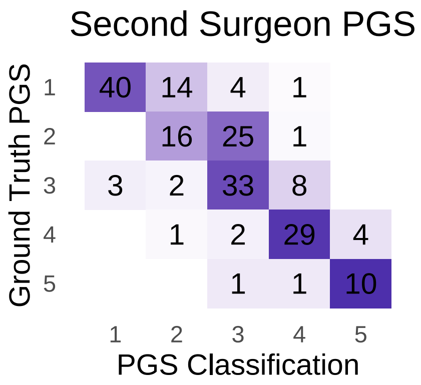
```r
library(patchwork)
(only_cf + plot_spacer() + combo_cf ) / (plot_spacer() + surg2_cf + plot_spacer())
```



	1	2	3	4	5
1	51	3	3	1	1
2	12	19	6	5	
3	8	7	14	13	4
4	4	4	10	13	5
5	1		4	6	1



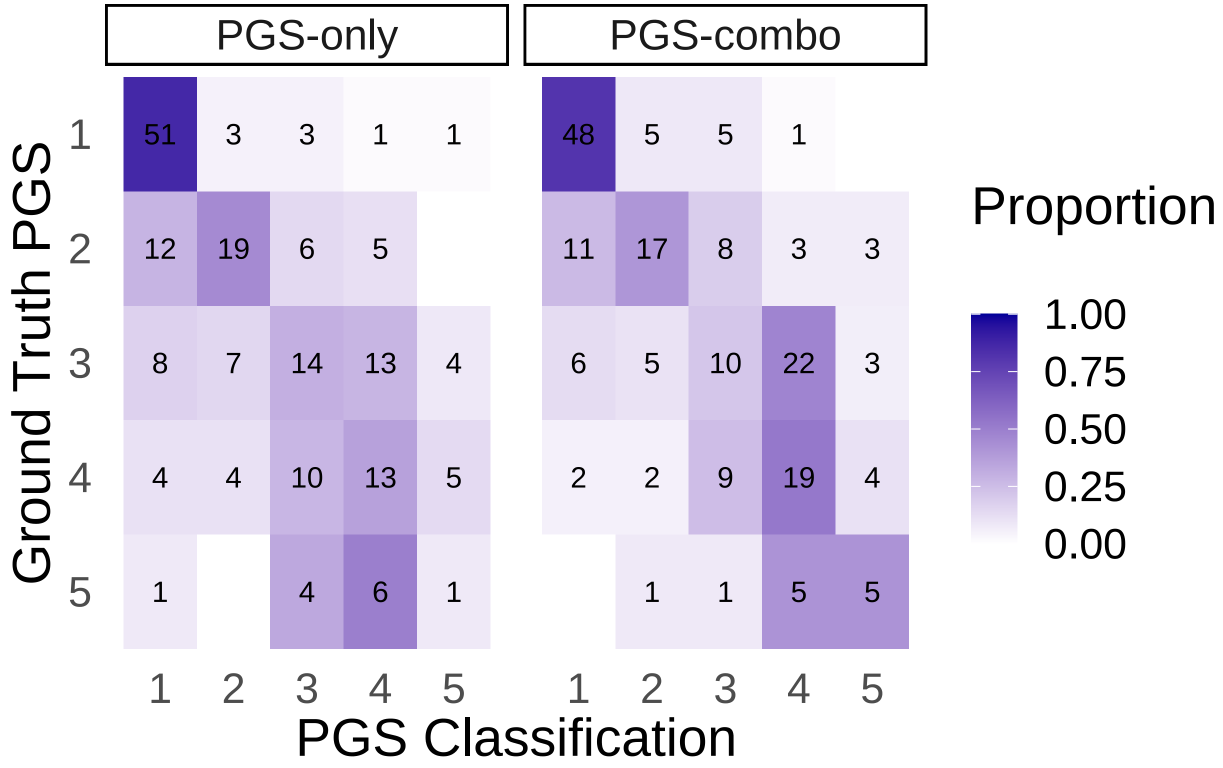
	1	2	3	4	5
1	48	5	5	1	
2	11	17	8	3	3
3	6	5	10	22	3
4	2	2	9	19	4
5		1	1	5	5



	1	2	3	4	5
1	40	14	4	1	
2		16	25	1	
3	3	2	33	8	
4		1	2	29	4
5			1	1	10

Paper figure

```
bind_rows(
        mutate(only_results, network = "PGS-only"),
        mutate(combo_results, network = "PGS-combo")
    ) %>%
    mutate(network = parse_factor(network, levels = c("PGS-only", "PGS-combo"))) %>%
    plot_confusion_matrix(title = "") +
    facet_wrap(vars(network), ncol = 2) +
    theme(text=element_text(size=20))
```



```r
ggsave("../output/confusion_matrices.pdf", width = 10, height = 6)
ggsave("../output/confusion_matrices.svg", width = 10, height = 6)
```

Environment

```r
sessioninfo::session_info()
```

```
## - Session info ---------------------------------------------------------
##  setting  value
##  version  R version 4.0.5 (2021-03-31)
##  os       Fedora 34 (Workstation Edition)
##  system   x86_64, linux-gnu
##  ui       X11
##  language (EN)
##  collate  en_US.UTF-8
##  ctype    en_US.UTF-8
##  tz       America/New_York
##  date     2021-08-29
##
## - Packages -------------------------------------------------------------
##  package     * version date       lib source
##  assertthat    0.2.1   2019-03-21 [1] CRAN (R 4.0.3)
##  backports     1.2.1   2020-12-09 [1] CRAN (R 4.0.4)
##  blob          1.2.1   2020-01-20 [1] CRAN (R 4.0.3)
##  broom         0.7.2   2020-10-20 [1] CRAN (R 4.0.3)
##  cellranger    1.1.0   2016-07-27 [1] CRAN (R 4.0.3)
##  cli           2.3.1   2021-02-23 [1] CRAN (R 4.0.4)
##  colorspace    2.0-0   2020-11-11 [1] CRAN (R 4.0.4)
##  crayon        1.4.1   2021-02-08 [1] CRAN (R 4.0.4)
##  curl          4.3     2019-12-02 [1] CRAN (R 4.0.3)
##  DBI           1.1.0   2019-12-15 [1] CRAN (R 4.0.3)
##  dbplyr        1.4.4   2020-05-27 [1] CRAN (R 4.0.3)
##  digest        0.6.27  2020-10-24 [1] CRAN (R 4.0.4)
##  dplyr       * 1.0.5   2021-03-05 [1] CRAN (R 4.0.4)
##  ellipsis      0.3.1   2020-05-15 [1] CRAN (R 4.0.3)
##  evaluate      0.14    2019-05-28 [1] CRAN (R 4.0.3)
##  fansi         0.4.2   2021-01-15 [1] CRAN (R 4.0.4)
##  farver        2.1.0   2021-02-28 [1] CRAN (R 4.0.4)
##  forcats     * 0.5.1   2021-01-27 [1] CRAN (R 4.0.4)
##  fs            1.5.0   2020-07-31 [1] CRAN (R 4.0.3)
##  gdtools     * 0.2.3   2021-01-06 [1] CRAN (R 4.0.3)
##  generics      0.1.0   2020-10-31 [1] CRAN (R 4.0.4)
##  ggplot2     * 3.3.3   2020-12-30 [1] CRAN (R 4.0.4)
##  glue          1.4.2   2020-08-27 [1] CRAN (R 4.0.3)
##  gtable        0.3.0   2019-03-25 [1] CRAN (R 4.0.3)
##  haven         2.3.1   2020-06-01 [1] CRAN (R 4.0.3)
##  highr         0.8     2019-03-20 [1] CRAN (R 4.0.3)
##  hms           0.5.3   2020-01-08 [1] CRAN (R 4.0.3)
##  htmltools     0.5.1.1 2021-01-22 [1] CRAN (R 4.0.4)
##  httr          1.4.2   2020-07-20 [1] CRAN (R 4.0.3)
##  irr         * 0.84.1  2019-01-26 [1] CRAN (R 4.0.3)
##  jsonlite      1.7.2   2020-12-09 [1] CRAN (R 4.0.4)
##  knitr         1.30    2020-09-22 [1] CRAN (R 4.0.3)
##  labeling      0.4.2   2020-10-20 [1] CRAN (R 4.0.3)
##  lifecycle     1.0.0   2021-02-15 [1] CRAN (R 4.0.4)
##  lpSolve     * 5.6.15  2020-01-24 [1] CRAN (R 4.0.3)
##  lubridate     1.7.9   2020-06-08 [1] CRAN (R 4.0.3)
##  magrittr      2.0.1   2020-11-17 [1] CRAN (R 4.0.4)
##  modelr        0.1.8   2020-05-19 [1] CRAN (R 4.0.3)
##  munsell       0.5.0   2018-06-12 [1] CRAN (R 4.0.3)
##  nvimcom     * 0.9-102 2021-05-17 [1] local
##  patchwork   * 1.1.1   2020-12-17 [1] CRAN (R 4.0.4)
##  pillar        1.5.1   2021-03-05 [1] CRAN (R 4.0.4)
##  pkgconfig     2.0.3   2019-09-22 [1] CRAN (R 4.0.3)
##  ps            1.6.0   2021-02-28 [1] CRAN (R 4.0.4)
##  purrr       * 0.3.4   2020-04-17 [1] CRAN (R 4.0.3)
##  R6            2.5.0   2020-10-28 [1] CRAN (R 4.0.4)
##  Rcpp          1.0.6   2021-01-15 [1] CRAN (R 4.0.4)
##  readr       * 1.4.0   2020-10-05 [1] CRAN (R 4.0.3)
##  readxl        1.3.1   2019-03-13 [1] CRAN (R 4.0.3)
##  reprex        0.3.0   2019-05-16 [1] CRAN (R 4.0.3)
##  rlang         0.4.10  2020-12-30 [1] CRAN (R 4.0.4)
##  rmarkdown   * 2.5     2020-10-21 [1] CRAN (R 4.0.3)
##  rstudioapi    0.11    2020-02-07 [1] CRAN (R 4.0.3)
##  rvest         0.3.6   2020-07-25 [1] CRAN (R 4.0.3)
##  scales        1.1.1   2020-05-11 [1] CRAN (R 4.0.3)
##  sessioninfo   1.1.1   2018-11-05 [1] CRAN (R 4.0.3)
##  showtext    * 0.9-2   2021-01-10 [1] CRAN (R 4.0.4)
##  showtextdb  * 3.0     2020-06-04 [1] CRAN (R 4.0.4)
##  stringi       1.5.3   2020-09-09 [1] CRAN (R 4.0.3)
##  stringr     * 1.4.0   2019-02-10 [1] CRAN (R 4.0.3)
##  svglite       1.2.3.2 2020-07-07 [1] CRAN (R 4.0.3)
##  sysfonts    * 0.8.3   2021-01-10 [1] CRAN (R 4.0.4)
##  systemfonts   0.3.2   2020-09-29 [1] CRAN (R 4.0.3)
##  tibble      * 3.1.0   2021-02-25 [1] CRAN (R 4.0.4)
##  tidyr       * 1.1.3   2021-03-03 [1] CRAN (R 4.0.4)
##  tidyselect    1.1.0   2020-05-11 [1] CRAN (R 4.0.3)
##  tidyverse   * 1.3.0   2019-11-21 [1] CRAN (R 4.0.3)
##  utf8          1.2.1   2021-03-12 [1] CRAN (R 4.0.4)
##  vctrs         0.3.7   2021-03-29 [1] CRAN (R 4.0.4)
##  withr         2.4.1   2021-01-26 [1] CRAN (R 4.0.4)
##  xfun          0.18    2020-09-29 [1] CRAN (R 4.0.3)
##  xml2          1.3.2   2020-04-23 [1] CRAN (R 4.0.3)
##  yaml          2.2.1   2020-02-01 [1] CRAN (R 4.0.3)
##
## [1] /home/thomas/R/x86_64-redhat-linux-gnu-library/4.0
## [2] /usr/lib64/R/library
## [3] /usr/share/R/library
```