1 Example 1: Estimation of $d'$ and $\tau$

It is assumed that we have $n = (2, 2, 6)$ observations in the three categories. We may estimate $\tau$, $d'$ and their standard errors with the `twoAC` function in the `sensR` package:

```r
> library(sensR)
> fit <- twoAC(c(2, 2, 6))
> fit
```

Results for the 2-AC protocol with data c(2, 2, 6):

<table>
<thead>
<tr>
<th></th>
<th>Estimate</th>
<th>Std. Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>tau</td>
<td>0.4160</td>
<td>0.2674</td>
</tr>
<tr>
<td>d.prime</td>
<td>0.7743</td>
<td>0.5417</td>
</tr>
</tbody>
</table>

Two-sided 95% confidence interval for d-prime based on the likelihood root statistic:

<table>
<thead>
<tr>
<th></th>
<th>Lower</th>
<th>Upper</th>
</tr>
</thead>
<tbody>
<tr>
<td>d.prime</td>
<td>-0.271</td>
<td>1.859</td>
</tr>
</tbody>
</table>

Significance test:

Likelihood root statistic = 1.446718 p-value = 0.14798

Alternative hypothesis: d-prime is different from 0

Alternatively we may compute $\tau$ and $d'$ manually:

```r
> n <- c(2, 2, 6)
> gamma <- cumsum(n/sum(n))
> z <- qnorm(gamma)[-3]
```
2 Example 2: Inference for d-prime

The likelihood based confidence intervals and the one-sided discrimination significance test where the null hypothesis is “no sensory difference”, i.e., \( d'_0 = 0 \) using the likelihood root statistic are immediately available using the `twoAC` function:

```r
> twoAC(c(2, 2, 6), d.prime0 = 0, conf.level = 0.95, statistic = "likelihood",
       alternative = "greater")
```

Results for the 2-AC protocol with data c(2, 2, 6):

<table>
<thead>
<tr>
<th></th>
<th>Estimate</th>
<th>Std. Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>tau</td>
<td>0.4160</td>
<td>0.2674</td>
</tr>
<tr>
<td>d.prime</td>
<td>0.7743</td>
<td>0.5417</td>
</tr>
</tbody>
</table>

Two-sided 95% confidence interval for d-prime based on the likelihood root statistic:

- Lower: -0.271
- Upper: 1.859

Significance test:

- Likelihood root statistic: 1.446718
- p-value: 0.073988
- Alternative hypothesis: d-prime is greater than 0

The relative profile likelihood and Wald approximation can be obtained with:

```r
> pr <- profile(fit)
> plot(pr)
> z <- pr$d.prime$d.prime
> w <- (coef(fit)[2, 1] - z)/coef(fit)[2, 2]
> lines(z, exp(-w^2/2), lty = 2)
```

3 Example 3: Power calculations

The function `twoACpwr` computes the power for the 2-AC protocol and a significance test of the users choice. The power assuming \( \tau = 0.5, d' = 1 \), the sample size is \( N = 20 \) for a two-sided preference test with \( \alpha = 0.5 \) is found with:

```r
> twoACpwr(tau = 0.5, d.prime = 1, size = 20, d.prime0 = 0,
          alpha = 0.05, alternative = "two.sided", tol = 1e-05)
```

<table>
<thead>
<tr>
<th></th>
<th>power</th>
<th>actual.alpha</th>
<th>samples discarded</th>
<th>kept</th>
<th>p.1</th>
<th>p.2</th>
<th>p.3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.777677</td>
<td>0.04960103</td>
<td>231</td>
<td>94</td>
<td>137</td>
<td>0.1444</td>
<td>0.2174</td>
</tr>
</tbody>
</table>
Apart from the power, we are told that the actual size of the test, $\alpha$ is close to the nominal 5%. The reason that the two differ is due to the discreteness of the observations, and hence the test statistic. We are also told that with $N = 20$ there are 231 possible outcomes of the 2-AC protocol. In computing the power 94 of these are discarded and power is computed based on the remaining 137 samples. The fraction of samples that are discarded is determined by the tol parameter. If we set this to zero, then no samples are discarded, however, the increase in precision is irrelevant:

```r
> twoACpwr(tau = 0.5, d.prime = 1, size = 20, d.prime0 = 0, alpha = 0.05, alternative = "two.sided", tol = 0)
```

<table>
<thead>
<tr>
<th>power actual.alpha samples discarded kept p.1 p.2 p.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 0.7776788 0.04960103 231 0 231 0.1444 0.2174 0.6382</td>
</tr>
</tbody>
</table>

The last three numbers in the output are the cell probabilities, so with $\tau = 0.5$ and $d' = 1$ we should, for example, expect around 22% of the answers in the “no difference” or “no preference” category.

4 Example 4: Estimation and standard errors via cumulative probit models

Estimates of $\tau$ and $d'$ and their standard errors can be obtained from a cumulative probit model. We begin by defining the three leveled response variable resp and fitting a cumulative link model (CLM) using the function clm in package ordinal with weights equal to the observed counts and a probit link. Standard output contains the following coefficient table:
> library(ordinal)
> response <- gl(3, 1)
> fit.clm <- clm(response ~ 1, weights = c(2, 2, 6), link = "probit")
> (tab <- coef(summary(fit.clm)))

| Estimate | Std. Error | z value | Pr(>|z|) |
|----------|------------|---------|----------|
| 1|2 | -0.8416212 | 0.4518154 | -1.8627547 | 0.06249679 |
| 2|3 | -0.2533471 | 0.4009896 | -0.6318047 | 0.52751451 |

The \( \tau \) and \( d' \) estimates are obtained by:

> theta <- tab[, 1]
> (tau <- (theta[2] - theta[1])/sqrt(2))

2|3
0.4159726

> (d.prime <- (-theta[2] - theta[1])/sqrt(2))

2|3
0.7742595

The variance-covariance matrix of the parameters can be extracted by the `vcov` method and the standard errors computed via the provided formulas:

> VCOV <- vcov(fit.clm)
> (se.tau <- sqrt((VCOV[1, 1] + VCOV[2, 2] - 2 * VCOV[2, 1])/2))

[1] 0.2674311

> (se.d.prime <- sqrt((VCOV[1, 1] + VCOV[2, 2] + 2 * VCOV[2, 1])/2))

[1] 0.5416737

Observe how these estimates and standard errors are identical to those in Example 1.

We could also have used the `clm2twoAC` function from package `sensR` which extract estimates and standard errors from a `clm` model fit object:

> clm2twoAC(fit.clm)

| Estimate | Std. Error | z value | Pr(>|z|) |
|----------|------------|---------|----------|
| tau | 0.4159726 | 0.2674311 | 1.555439 | 0.11984 |
| d-prime | 0.7742595 | 0.5416737 | 1.429384 | 0.15289 |

5 Example 5: A regression model for \( d' \)

Assume that a study is performed and gender differences in the discrimination ability is of interest. Suppose that \((20, 20, 60)\) is observed for women and \((10, 20, 70)\) is observed for men. The standard output from a cumulative probit model contains the coefficient table:

> n.women <- c(2, 2, 6) * 10
> n.men <- c(1, 2, 7) * 10
> wt <- c(n.women, n.men)
> response <- gl(3, 1, length = 6)
> gender <- gl(2, 3, labels = c("women", "men"))
\texttt{fm2 <- clm(response ~ gender, weights = wt, link = "probit")}

\texttt{(tab2 <- coef(summary(fm2)))}

\begin{verbatim}
  Estimate  Std. Error   z value Pr(>|z|)
1|2  -0.8887474  0.1346051 -6.602627 4.039349e-11
2|3  -0.2283095  0.1238663 -1.843194 6.530068e-02
gendermen 0.3205403  0.1741445  1.840658 6.567176e-02
\end{verbatim}

The estimate of \(\tau\) (assumed constant between genders) and the gender specific estimates of \(d'\) can be extracted by:

\texttt{theta <- fm2$alpha}

\texttt{(tau <- (theta[2] - theta[1])/sqrt(2))}

\begin{verbatim}
  2|3
  0.4670001
\end{verbatim}

\texttt{(d.women <- (-theta[2] - theta[1])/sqrt(2))}

\begin{verbatim}
  2|3
  0.7898785
\end{verbatim}

\texttt{(d.men <- d.women + fm2$beta * sqrt(2))}

\begin{verbatim}
  2|3
  1.243191
\end{verbatim}

Again we could use the \texttt{clm2twoAC} function to get the coefficient table for the 2-AC model from the CLM-model:

\texttt{clm2twoAC(fm2)}

\begin{verbatim}
  Estimate  Std. Error   z value Pr(>|z|)
  tau  0.4670001  0.06700014  6.970135 3.1664e-12
  d-prime 0.7898785  0.17021271  4.640538 3.4750e-06
  gendermen 0.4533125  0.24627746  1.840658 6.567176e-02
\end{verbatim}

Observe that \(d'\) for women is given along with the difference in \(d'\) for men and women rather than \(d'\) for each of the genders.

The Wald test for gender differences is directly available from the coefficient table with \(p = 0.0657\). The corresponding likelihood ratio test can be obtained by:

\texttt{fm3 <- update(fm2, ~. - gender)}

\texttt{anova(fm2, fm3)}

\begin{verbatim}
 Likelihood ratio tests of cumulative link models:

 formula: link: threshold:
 fm3 response ~ 1 probit flexible
 fm2 response ~ gender probit flexible

 no.par AIC logLik LR.stat df Pr(>Chisq)
 fm3 2 358.59 -177.29
 fm2 3 357.19 -175.59 3.3997 1 0.06521 .

---
 Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
\end{verbatim}
which is slightly closer to significance. The 95% profile likelihood confidence interval for the difference between men and women on the $d'$-scale is:

```r
> confint(fm2) * sqrt(2)

          2.5 %     97.5 %
gendermen -0.02850933  0.9372283
```

The likelihood ratio test for the assumption of constant $\tau$ is computed in the following. The likelihood ratio statistic and associated $p$-value are

```r
> logLik(fm2)

'log Lik.' -175.593 (df=3)
```

```r
> tw <- twoAC(n.women)
> tm <- twoAC(n.men)
> (LR <- 2 * (tw$logLik + tm$logLik - fm2$logLik))

[1] 0.7682623
```

```r
> pchisq(LR, 1, lower.tail = FALSE)

[1] 0.3807552
```

The Pearson $X^2$ test of the same hypothesis is given by

```r
> freq <- matrix(fitted(fm2), nrow = 2, byrow = TRUE) * 100
> Obs <- matrix(wt, nrow = 2, byrow = TRUE)
> (X2 <- sum((Obs - freq)^2/freq))

[1] 0.7657565
```

```r
> pchisq(X2, df = 1, lower.tail = FALSE)

[1] 0.381533
```

so the Pearson and likelihood ratio tests are very closely equivalent as it is so often the case.

### 6 Regression model for replicated 2-AC data

The data used in example 6 are not directly available at the time of writing. Assume however, that data are available in the R `data.frame repData`. Then the cumulative probit mixed model where preference depends on reference and consumers (cons) are random can be fitted with:

```r
> fm3.agq <- clmm2(preference ~ reference, random = cons, nAGQ = 10, data = repData, link = "probit", Hess = TRUE)
> summary(fm3.agq)
```

Cumulative Link Mixed Model fitted with the adaptive Gauss-Hermite quadrature approximation with 10 quadrature points

Call:
`clmm2(location = preference ~ reference, random = cons, data = repData, Hess = TRUE, link = "probit", nAGQ = 10)`
Random effects:

<table>
<thead>
<tr>
<th>Var</th>
<th>Std.Dev</th>
</tr>
</thead>
<tbody>
<tr>
<td>cons</td>
<td>1.367074</td>
</tr>
</tbody>
</table>

Location coefficients:

| Estimate | Std. Error | z value | Pr(>|z|) |
|----------|------------|---------|---------|
| reference | 0.4059     | 0.1002  | 4.0519  | 5.0809e-05 |

No scale coefficients

Threshold coefficients:

<table>
<thead>
<tr>
<th>Estimate</th>
<th>Std. Error</th>
<th>z value</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>N</td>
<td>0.4823</td>
</tr>
<tr>
<td>N</td>
<td>B</td>
<td>0.8488</td>
</tr>
</tbody>
</table>

log-likelihood: -668.9122

AIC: 1345.824

Condition number of Hessian: 36.79843

Here we asked for the accurate 10-point adaptive Gauss-Hermite quadrature approximation. The 2-AC estimates are available using the \texttt{clm2twoAC} function:

\begin{verbatim}
> clm2twoAC(fm3.agq)

| Estimate | Std. Error | z value | Pr(>|z|) |
|----------|------------|---------|---------|
| tau      | 0.2592192  | 0.02902574 | 8.930665 | < 2.22e-16 |
| d-prime  | -0.9412296 | 0.15975323 | -5.891772 | 3.8208e-09 |
| reference | 0.5740294  | 0.14167004 | 4.051876 | 5.0809e-05 |
\end{verbatim}

The standard deviation of the consumer-specific \( d \)'s is given by:

\begin{verbatim}
> fm3.agq$stDev * sqrt(2)

<table>
<thead>
<tr>
<th>cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.653526</td>
</tr>
</tbody>
</table>
\end{verbatim}

The profile likelihood curve can be obtained using the profile method:

\begin{verbatim}
> pr <- profile(fm3.agq, range = c(0.7, 1.8))
\end{verbatim}

And then plotted using:

\begin{verbatim}
> plpr <- plot(pr, fig = FALSE)
> plot(sqrt(2) * plpr$stDev$x, plpr$stDev$y, type = "l", xlab = expression(sigma[delta]), ylab = "Relative profile likelihood", xlim = c(1, 2.5), axes = FALSE)
> axis(1)
> axis(2, las = 1)
> abline(h = attr(plpr, "limits"))
> text(2.4, 0.17, "95\% limit")
> text(2.4, 0.06, "99\% limit")
\end{verbatim}

The resulting figure is shown in Figure 2. The profile likelihood confidence intervals are obtained using:

\begin{verbatim}
> confint(pr, level = 0.95) * sqrt(2)
\end{verbatim}
The probabilities that consumers prefer yoghurt A, have no preference or prefer yoghurt B can, for an average consumer be obtained by predicting from the CLMM:

```r
> newdat <- expand.grid(preference = factor(c("A", "N", "B"),
  levels = c("A", "N", "B"), ordered = TRUE), reference = factor(c("A",
  "B")))
> pred <- predict(fm3.agq, newdata = newdat)
```

The predictions for the extreme consumers have to be obtained by hand. Here we ask for the predictions for the 5th percentile and 95th percentile of the consumer population for reference A and B:

```r
> q95.refA <- diff(c(0, pnorm(fm3.agq$Theta - qnorm(1 - 0.05) *
  fm3.agq$stDev), 1))
> q05.refA <- diff(c(0, pnorm(fm3.agq$Theta - qnorm(0.05) *
  fm3.agq$stDev), 1))
> q95.refB <- diff(c(0, pnorm(fm3.agq$Theta - fm3.agq$beta -
  qnorm(1 - 0.05) * fm3.agq$stDev), 1))
> q05.refB <- diff(c(0, pnorm(fm3.agq$Theta - fm3.agq$beta -
  qnorm(0.05) * fm3.agq$stDev), 1))
```

Plotting follows the standard methods:

```r
> par(mar = c(0, 2, 0, 0.5) + 0.5)
> plot(1:3, pred[1:3], ylim = c(0, 1), axes = FALSE, xlab = "",
  ylab = "", pch = 19)
> axis(1, lwd.ticks = 0, at = c(1, 3), labels = c("", ""))
> axis(2, las = 1)
> points(1:3, pred[4:6], pch = 1)
> lines(1:3, pred[1:3])
> lines(1:3, pred[4:6], lty = 2)
```

Figure 2: Profile likelihood for $\sigma_\delta$ Horizontal lines indicate 95% and 99% confidence limits.

\[
\begin{align*}
2.5 \% & \quad 97.5 \% \\
stDev & \quad 1.362331 \quad 2.00148
\end{align*}
\]
Figure 3: The probabilities of preferring yoghurt A, having no preference, or preferring yoghurt B for an average consumer ($b = 0$) and for fairly extreme consumers ($b = \pm 1.64\sigma_b$).

The resulting figure is shown in Fig. 3.
7 End note

Versions details for R and the packages sensR and ordinal appear below:

```r
> sessionInfo()

R Under development (unstable) (2020-02-08 r77786)
Platform: x86_64-apple-darwin15.6.0 (64-bit)
Running under: macOS Catalina 10.15.1

Matrix products: default
BLAS: /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRblas.0.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRlapack.dylib

locale:

attached base packages:
[1] stats graphics grDevices utils datasets methods base

other attached packages:
[1] ordinal_2019.12-10 sensR_1.5-2

loaded via a namespace (and not attached):
[1] lattice_0.20-38    codetools_0.2-16    mvtnorm_1.0-12
[4] zoo_1.8-7          ucminf_1.1-4        MASS_7.3-51.5
[7] grid_4.0.0         multcomp_1.4-12    Matrix_1.2-18
[10] sandwich_2.5-1     splines_4.0.0       TH.data_1.0-10
[13] tools_4.0.0        numDeriv_2016.8-1.1 survival_3.1-8
[16] compiler_4.0.0
```

References


