Re-Understanding Ordered Categorical Regression

Steve Hof

30/12/2020

Dealing with discrete ordered Values

In principle, an ordered categorical variable is just a **multinomial prediction problem**.

Multinomial - (unordered events) (just a side note)

If there are K types of events with probabilities $p_1, \ldots, p_K$, then the probability of observing $y_1, \ldots, y_K$ events of each type out of n total trials is:

$$\mathbb{P}(y_1, \ldots, y_K \mid n, p_1, \ldots, p_K) = \frac{n!}{\prod_i y_i!} \prod_{i=1}^{K} p_i^{y_i}$$

Dealing with a multinomial regression requires a link function used to "link" the multinomial to a linear regression. The link function we usually use is the multinomial logit (aka softmax). It takes a vector of scores, one for each of K even types, and computes the probability of a particular type of event k as:

$$\mathbb{P}(k \mid s_1, s_2, \ldots, s_K) = \frac{\exp(S_k)}{\sum_{i=1}^{K} \exp(s_i)}$$

introducing order

We must add the constraint that the categories be ordered. What we'd like is for any associated predictor variable, as it increases, to move predictions progressively through the categories in sequence. The difficulty is how to ensure that the linear model maps onto the outcomes in the right order. We do this with the **cumulative link** function. By linking a linear model to cumulative probability, it is possible to guarantee the ordering of the outcomes. There are two steps:

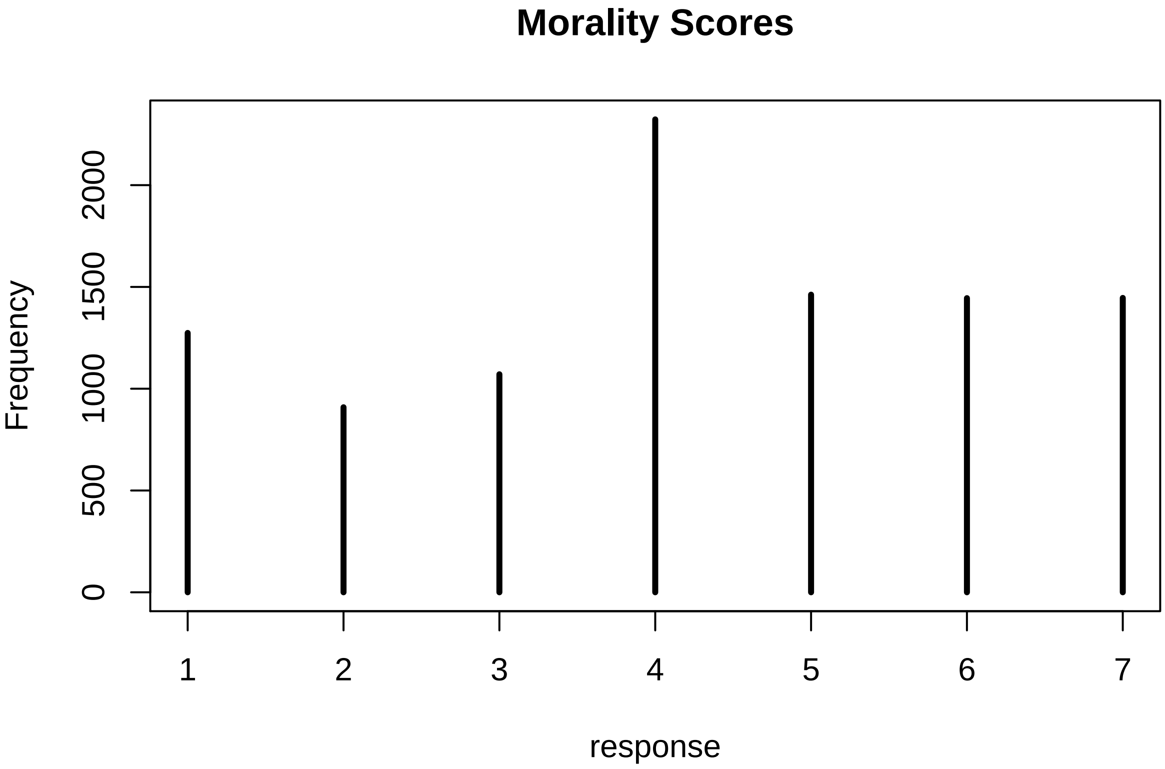
1. Parameterize a distribution of outcomes on the log-cumulative-odds scale.
2. Introduce a predictor(s) to the log-cumulative-odds values, allowing us to model associations between predictors and the outcome while obeying the ordered nature of prediction.

Example in R

The data are from an experiment in which people were told a story and asked to rate the protagonist's decision in terms of its morality on a scale from $1 - 7$. There are 12 columns and 9930 rows, comprising data for 331 unique individuals. The outcome we care about is *response*.

```
data(Trolley)
df = Trolley

# take a look at the distribution
simplehist(df$response, xlim=c(1, 7), xlab = "response", main = "Morality Scores")
```



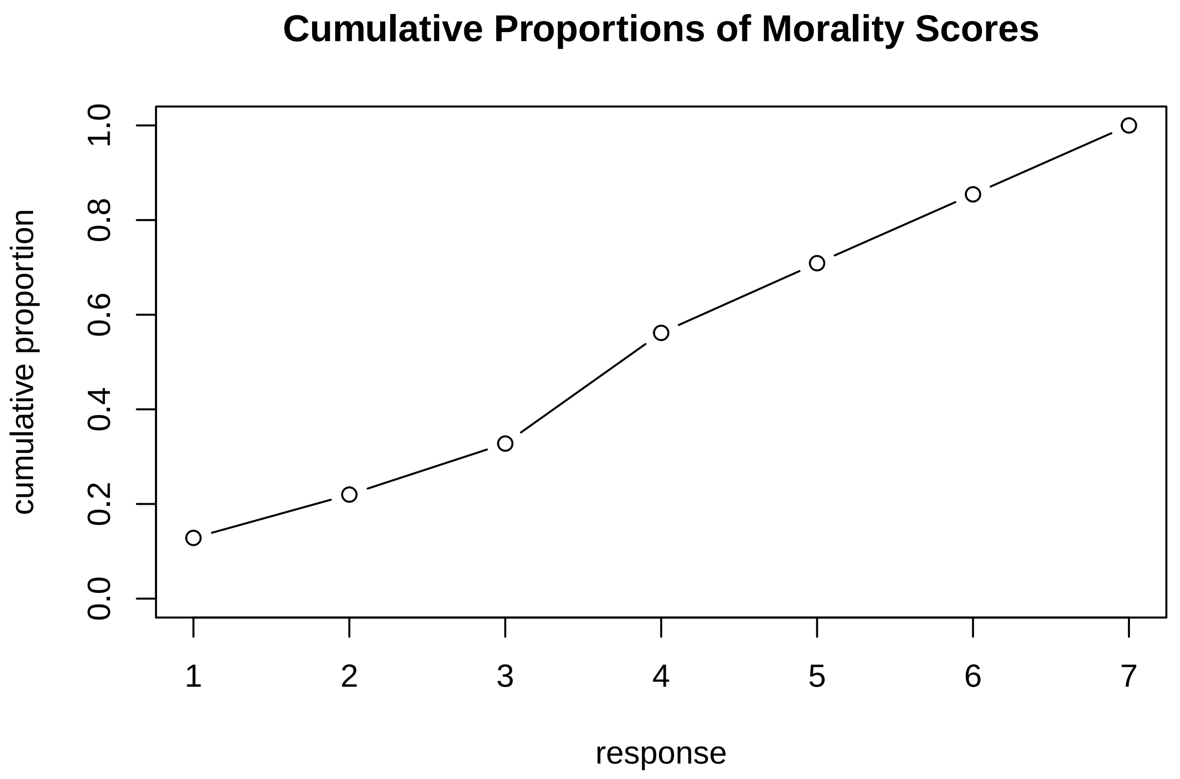
We need to re-describe the histogram on the log-cumulative-odds scale. (constructing the odds of a cumulative probability and then taking a logarithm). Doing so is designed to constrain the probabilities to the $[0, 1]$ interval.

First, we calculate the cumulative proportions of each of the response values, $[1 - 7]$

```
# the proportion of each of the discrete response values
pr_k = table(df$response) / nrow(df)

# the cumulative proportions
cum_pr_k = cumsum(pr_k)

plot(x = 1:7, y = cum_pr_k, type = "b", xlab = "response",
ylab = "cumulative proportion", ylim = c(0, 1), main = "Cumulative Proportions of Morality Scores")
```



Then, to re-describe the histogram in terms of log-cumulative-odds, we need a series of intercept parameters. Each intercept will be on the log-cumulative-odds scale and stand in for the cumulative probability of each outcome. The log-cumulative-odds that a response value y_i is less than or equal to some possible outcome value k is:

$$\log\left(\frac{\mathbb{P}\left(y_i \leq k\right)}{1 - \mathbb{P}\left(y_i \leq k\right)}\right) = \alpha_k$$

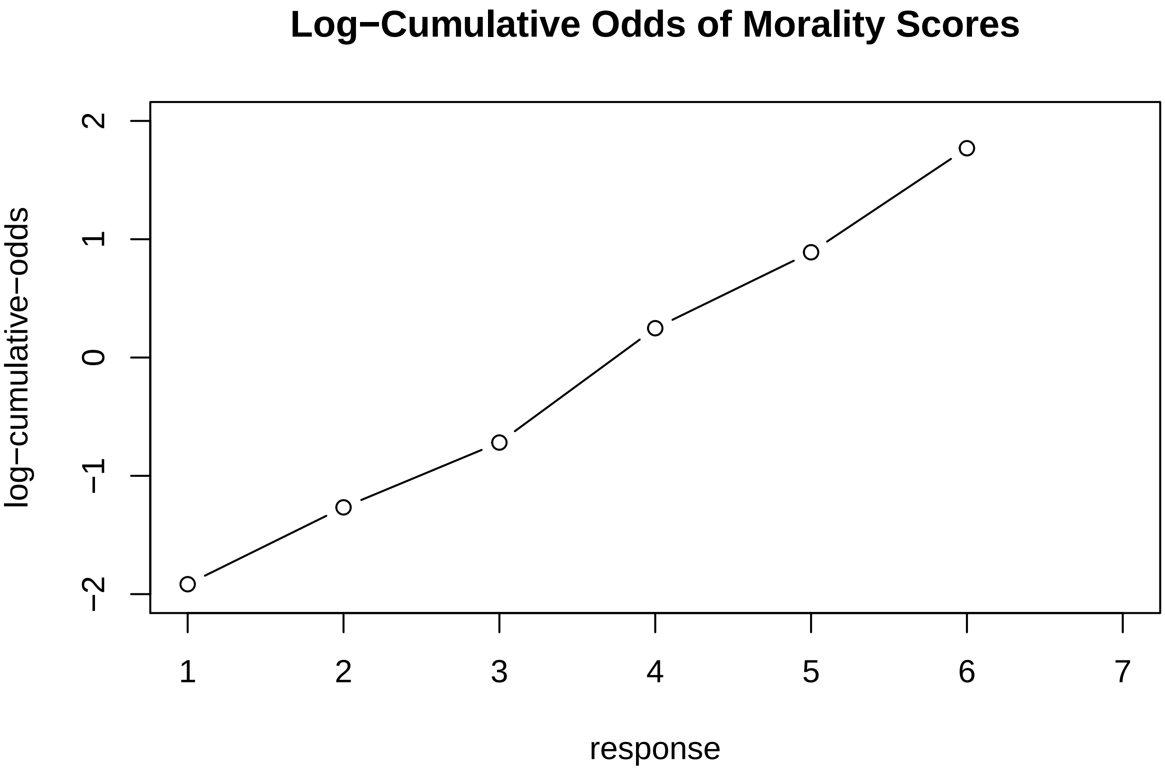
where α_k is an "intercept" unique to each possible outcome value k.

We can compute these intercept parameters directly

```
# GLM link function for binomial response
logit = function(x) log(x / (1 - x))
round(lco <- logit(cum_pr_k), 2)
```

```
##     1     2     3     4     5     6     7
## -1.92 -1.27 -0.72  0.25  0.89  1.77   Inf
```

```
plot(x = 1:7, y = lco, type = "b", xlab = "response",
ylab = "log-cumulative-odds", ylim = c(-2, 2), main = "Log-Cumulative Odds of Morality Scores")
```



Since the largest response value, 7, always has a cumulative probability of 1, we can treat it as a free variable (we do not need a parameter for it). So for $K = 7$ possible response values, we only need $K - 1 = 6$ intercepts.

When performing a Bayesian style regression, what we're generally interested in is the posterior distribution. To use Bayes' theorem to compute the posterior distribution of the intercepts, we'll need to compute the likelihood of each possible response value. We'll use the cumulative probabilities, $P(y_i \leq k)$, to compute the likelihood, $P(y_i = k)$. (recall this annoying process from STAT 260, lol).

Each intercept, α_k, implies a cumulative probability for each k. We then just use the inverse link function to translate from log-cumulative-odds back to cumulative probability. So when we observe k and need its likelihood, we can get the likelihood by subtraction:

$$p_k = \mathbb{P}(y_i = k) = \mathbb{P}(y_i \leq k) - \mathbb{P}(y_i \leq k - 1)$$

We can express our multilevel model in the following way:

$$\begin{aligned}
&R_i \sim \text{Ordered-logit}(\phi_i, \kappa) \;\; \text{[probability of data]} \\
&p_1 = q_1 \;\; \text{[probabilities of each value } k\text{]} \\
&\phi_i = 0 \;\; \text{[linear model]} \\
&\kappa_k \sim bob
\end{aligned}$$