Regression Models for Ordinal Data
Introducing $R$-package \texttt{ordinal}

Rune H B Christensen

DTU Informatics, IMM
Section for Statistics
Technical University of Denmark
rhbc@imm.dtu.dk

August 17th 2011
Examples of ordinal response variables

- MR scannings of cancer (greatly enlarged, enlarged, no change, smaller, much smaller)
- Smoking frequency (never, occasionally, <1 pack/day, >1 pack/day)
- BMI (underweight, normal weight, overweight, obese)
- Questionaire (strongly disagree, disagree, undecided, agree, strongly agree)
Cumulative link models (CLMs)

The cumulative link model — also known as:

- Proportional odds model
- Ordered probit/logit model
- Ordinal regression model

\[
P(Y_i \leq j) = g(\theta_j - \mathbf{x}_i^T \beta)
\]
The wine data

Table: The wine data (Randall, 1989), N=72

<table>
<thead>
<tr>
<th>Variables</th>
<th>Type</th>
<th>Values</th>
</tr>
</thead>
</table>
| bitterness  | response| 1, 2, 3, 4, 5
|             |         | less — more                 |
| temperature | predictor| cold, warm                  |
| contact     | predictor| no, yes                     |
| judges      | random  | 1, . . . , 9                |

- How does the perceived bitterness of wine depend on temperature and contact?
- A linear model is not a good idea
The **ordinal** package — an overview

Main functions:
- Cumulative link models (CLMs):
  \[ \text{clm}(\text{formula, data, link, ....}) \]
- Cumulative link mixed models (CLMMs):
  \[ \text{clmm}(\text{formula, data, link, ....}) \]
  (lmer syntax)

Other functions:
- clm.control
- clmm.control
- 15 additional exported function

Numerous methods:
- summary, anova, predict, confint, ...
Existing implementations of cumulative link models

- `polr` from **MASS** — widely used implementation
- `lrm` from **Design**
- `cumulative` from **VGAM**
- `MCMCglmm` from **MCMCglmm** (mixed models)
Challenges in implementing CLMs

1. Intuitive user interface
2. Efficient computational methods
3. Substantial scope of models
4. Useful methods and auxiliary functions
5. Readable code
6. Comprehensive Documentation
Challenges in implementing CLMs

1. Intuitive user interface
2. Efficient computational methods
3. Substantial scope of models
4. Useful methods and auxiliary functions
5. Readable code
6. Comprehensive Documentation
Fitting and displaying CLMs with ordinal

```r
> fm1 <- clm(rating ~ temp + contact, data = wine, link = "probit")
> summary(fm1)

formula: rating ~ temp + contact
data: wine

link threshold nobs logLik AIC niter max.grad cond.H
probit flexible 72 -85.76 183.52 5(0) 1.44e-13 2.2e+01

Coefficients:

|            | Estimate | Std. Error | z value | Pr(>|z|) |
|------------|----------|------------|---------|----------|
| tempwarm   | 1.4994   | 0.2918     | 5.139   | 2.77e-07 *** |
| contactyes | 0.8677   | 0.2669     | 3.251   | 0.00115 ** |
```

---

Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Threshold coefficients:

<table>
<thead>
<tr>
<th></th>
<th>Estimate</th>
<th>Std. Error</th>
<th>z value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>-0.7733</td>
<td>0.2829</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>0.7360</td>
<td>0.2499</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>2.0447</td>
<td>0.3218</td>
</tr>
</tbody>
</table>

Rune H B Christensen (DTU)
Aliased coefficients

> fm.soup <- clm(SURENESS ~ PRODID * DAY, data = soup)
> summary(fm.soup)

formula: SURENESS ~ PRODID * DAY
data: soup

link threshold nobs logLik AIC niter max.grad cond.H
logit flexible 1847 -2672.08 5374.16 6(1) 1.95e-13 9.4e+02

Coefficients: (1 not defined because of singularities)

| Estimate | Std. Error | z value | Pr(>|z|) |
|----------|------------|---------|----------|
| PRODID2  | 0.6665     | 0.2146  | 3.106    | 0.00189 ** |
| PRODID3  | 1.2418     | 0.1784  | 6.959    | 3.42e-12 *** |
| PRODID4  | 0.6678     | 0.2197  | 3.040    | 0.00237 ** |
| PRODID5  | 1.1194     | 0.2400  | 4.663    | 3.11e-06 *** |
| PRODID6  | 1.3503     | 0.2337  | 5.779    | 7.53e-09 *** |
| DAY2     | -0.4134    | 0.1298  | -3.186   | 0.00144 ** |
| PRODID2:DAY2 | 0.4390       | 0.2590  | 1.695    | 0.09006 . |
| PRODID3:DAY2 | NA          | NA      | NA       | NA       |
| PRODID4:DAY2 | 0.3308       | 0.3056  | 1.083    | 0.27892  |
| PRODID5:DAY2 | 0.3871       | 0.3248  | 1.192    | 0.23329  |
Likelihood ratio tests of cumulative link models:

```r
> fm2 <- update(fm1, ~. - temp)
> anova(fm1, fm2)

Likelihood ratio tests of cumulative link models:

<table>
<thead>
<tr>
<th>formula:</th>
<th>link:</th>
<th>threshold:</th>
</tr>
</thead>
<tbody>
<tr>
<td>fm2 rating ~ contact</td>
<td>probit</td>
<td>flexible</td>
</tr>
<tr>
<td>fm1 rating ~ temp + contact</td>
<td>probit</td>
<td>flexible</td>
</tr>
</tbody>
</table>

no. par   AIC      logLik  LR.stat df Pr(>Chisq)
fm2       5 210.05 -100.026
fm1       6 183.52 -85.761  28.529 1  9.231e-08 ***
---
```

Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Rune H B Christensen (DTU)
Computational challenges

- Robust starting values
  - The `clm` should always converge from the default starting value
  - It should be possible to supply starting values
- Speedy model estimation
  - Speed is maintained despite model scope and flexibility
- Accurate estimates
- Accurate standard errors
Accuracy of parameter estimates

```r
> fm1
formula: rating ~ temp + contact
data: wine

link threshold nobss logLik AIC niter max.grad
probit flexible  72   -85.76  183.52  5(0)  1.44e-13

Coefficients:
temp warm contact yes
 1.4994  0.8677

Threshold coefficients:
1|2  2|3  3|4  4|5
-0.7733  0.7360  2.0447  2.9413
```

- Has the model converged?
- How accurate are these estimates?
Implementation of CLMs in ordinal

Assessment of model convergence

```r
> slice.fm1 <- slice(fm1, parm = c(1, 6))
> par(mfrow = c(1, 2))
> plot(slice.fm1)
```

![Graphs showing relative log-likelihoods](image)
Assessment of parameter accuracy

> convergence(fm1)

    nobs logLik  niter max.grad cond.H logLik.Error
      72  -85.76   5(0)  1.44e-13  2.2e+01  0.00e+00

<table>
<thead>
<tr>
<th>Estimate</th>
<th>Std.Err</th>
<th>Gradient</th>
<th>Error</th>
<th>Cor.Dec</th>
<th>Sig.Dig</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>-0.7733</td>
<td>0.2829</td>
<td>1.59e-14</td>
<td>1.91e-16</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>0.7360</td>
<td>0.2499</td>
<td>1.31e-13</td>
<td>-5.65e-16</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>2.0447</td>
<td>0.3218</td>
<td>-1.44e-13</td>
<td>-8.26e-15</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>2.9413</td>
<td>0.3873</td>
<td>-6.46e-15</td>
<td>-7.72e-15</td>
</tr>
<tr>
<td>tempwarm</td>
<td>1.4994</td>
<td>0.2918</td>
<td>-1.38e-14</td>
<td>-5.00e-15</td>
<td>14</td>
</tr>
<tr>
<td>contactyes</td>
<td>0.8677</td>
<td>0.2669</td>
<td>1.88e-15</td>
<td>-2.25e-15</td>
<td>14</td>
</tr>
</tbody>
</table>

Eigen values of Hessian:
61.616 53.876 32.283 17.241 13.393 2.825
Extending the model class

- Scale effects
  \[
  \text{clm(rating} \sim \text{contact, scale} = \sim \text{temp, data=wine)}
  \]

- Structured thresholds
  \[
  \text{clm(rating} \sim \text{contact, data=wine, threshold="symmetric"} \\
  \text{clm(rating} \sim \text{contact, data=wine, threshold="equidistant"})
  \]

- Nominal effects (partial proportional odds)
  \[
  \text{clm(rating} \sim \text{contact, nominal} = \sim \text{temp, data=wine)}
  \]

- Flexible link functions
- Random effects
  - For grouped and multilevel data
Implementation of CLMs in ordinal

Cumulative link mixed models (CLMMs)

- Multiple random effect terms
  - Nested and crossed random effect structures
  - No correlated random effects (yet)
  - No random slopes (yet)

- Computational methods
  - Laplace approximation
  - Adaptive Gauss-Hermite quadrature (+ non-adaptive GHQ)

Example:

```r
> fm.ran <- clmm(rating ~ contact + temp + (1 | judge), data = wine)
```
Methods for \texttt{clm} fits

- Standard methods: print, summary, anova, predict
- Extractor methods: coef, vcov, logLik, AIC, fitted, ...
- Model development and selection: drop1, add1, step
- Model assessment methods: profile, plot.profile, confint
- Numerous additional methods
Methods for \texttt{clm} fits

- **Standard methods:**
  - print, summary, anova, \texttt{predict} with \texttt{se} and \texttt{CI}

- **Extractor methods:**
  - coef, vcov, logLik, AIC, fitted, …

- **Model development and selection:**
  - drop1, add1, step

- **Model assessment methods:**
  - profile, \texttt{plot.profile}, confint

- **Numerous additional methods**
Summary

- Reliable computational methods
- Methods for assessing convergence
- Extends the basic model with:
  - scale effects
  - nominal effects
  - random effects
  - structured thresholds
- A suite of helpful methods for clm and clmm objects
Future work

- slice and convergence methods for clmm fits
- More flexible random effect structures
- AGQ methods for nested random effects
Acknowledgments

Thanks to the Program Committee

Thanks to Professor Per Bruun Brockhoff

Thank you for listening!
Bibliography

