Visualizing predictions from multinomial models in R

by Christoph Scherber

Introduction

Multinomial models are linear statistical models for which the response variable is a factor with more than two levels. These models (also termed as generalized logit models) are extensions to the more familiar binomial regression models (logistic regression or logit models). In a multinomial model, the response variable can be expressed either as a vector of category names or as a matrix of counts of occurrences of these categories. While setting up multinomial models in R is usually straightforward, visualizing predictions from these models and checking the adequacy of model fit are more difficult to realize. Here, I show examples from a wide range of R packages dealing with multinomial models, and I provide guidelines to model visualization.

What is a multinomial model?

In a traditional logistic regression model, the response variable is a discrete variable that comes either as a binary response (zeroes and ones) or as a binomial response (number of successes in a given number of trials). For example, a binary response variable could be coded as 0 for healthy and 1 for ill. Likewise, in a binomial regression model, the number of successes could be the number of uninfected whereas the (corresponding) number of failures could be the number of infected patients. Now, in a multinomial model, these principles are just extended to the multi-category case; that is, the response variable has more than two response categories. Importantly, these models usually work with a baseline category, that is, one of the categories is selected to be the baseline to which all other categories are then compared. A multinomial model can essentially be expressed as a series of individual logistic regression models. Mathematically, each response category enters with the baseline category in a logit transformation, such that:

Where η_j is the linear predictor for response category j, $ß_0$ and $ß_1$ are the intercept and the slope respectively, and x is a numeric explanatory variable. The linear predictor η_j is related to the explanatory variables using the logit link function: logit = log[p/(1 - p)]

or, in the particular case considered here,

Data structures

Similar to binomial regression, response variables can come in a range of different formats. There are three different ways in which response variables are encountered in a multinomial model:

1) A long vector of names, for example ACBCCBABA (where A, B and C are factor levels) or (in R notation) `c("A","C","B","C","C","B","A","B","A")`

2) A table of contingencies, for example A: 10, B: 30, C: 5

3) A matrix of vectors, for example: A {1, 3, 4, 5}; B {2, 1, 3, 2}; C {3, 1, 5, 4} or (in R notation) `cbind(c(1,3,4,5),c(2,1,3,2),c(3,1,5,4))`

These three data structures are of course interchangeable. Normally, it's a major programming task to provide these structures in a format that is most useful for plotting.
While many textbooks on R programming generally recommend the cbind() notation, this notation is particularly poor when it comes to graphical representation of model predictions. It is generally preferable to use a long vector of names, as this allows the greatest flexibility in plotting the predictions of a given model.

For illustration, let's use the `housing` dataset from the `MASS` library in R (Venables & Ripley 2002) that will be used further down to show the capabilities of different R packages for multinomial models. The dataset contains the following variables (cited from R´s documentation of `housing`):

- **Sat** – The response variable, which is the satisfaction of householders with their present housing circumstances; High, Medium or Low, an ordered factor;
- **Infl** – The perceived degree of influence householders have on the management of the property (ordered factor, High, Medium, Low).
- **Type** - Type of rental accommodation (one of Tower, Atrium, Apartment or Terrace).
- **Cont** – The degree of contact residents are afforded with other residents; Low or High.
- **Freq** – Frequency, which refers to the number of residents in each class. This is part of the response variable. It either enters the model as a `weights` argument or used to generate a long vector of response categories.

We now load the dataset to explore the possibilities of multinomial modelling in R.

```
require(MASS) # loads the MASS library
data(housing) # loads the housing dataset
head(housing) #shows the first lines of the dataset

#     Sat   Infl  Type Cont Freq
#1    Low    Low Tower  Low   21
#2 Medium    Low Tower  Low   21
#3   High    Low Tower  Low   28
#4    Low Medium Tower  Low   34
#5 Medium Medium Tower  Low   22
#6   High Medium Tower  Low   36
```

We need to restructure the dataset into "long" format in order to set up multinomial models and produce graphical displays of the model predictions:

```
mytimes<-housing$Freq
housing.long<-as.data.frame(apply(housing,
2,function(x)rep(x,mytimes)),row.names=FALSE)

head(housing.long)

#  Sat Infl  Type Cont Freq
#1 Low  Low Tower  Low   21
#2 Low  Low Tower  Low   21
#3 Low  Low Tower  Low   21
#4 Low  Low Tower  Low   21
#5 Low  Low Tower  Low   21
#6 Low  Low Tower  Low   21
```

This shows that the variable "Sat" has been replicated 21 times, followed by 21, 28, 34 replicates and so on. Check the dimensions of both datasets:

```
dim(housing)
# [1] 72  5
dim(housing.long)
# [1] 1681    5
```

Hence, we now have a compact dataset (housing) with 72 rows and an expanded version of the same dataset, with 1681 rows.

R packages for multinomial modelling

There is a wide range of R packages available for multinomial modelling, some of which even allow the incorporation of random effects.

The following packages are frequently used and cited for situations where only fixed effects are considered:

- Function `multinom` in the `nnet` library (Venables & Ripley 2002). It's the most commonly used package which is based on neural networks
- Function `polr` in the `MASS` library (Venables & Ripley 2002), is also useful for ordered categorical response variables
- Function `glmnet` in the `glmnet` library (Friedman et al. 2010). It uses shrinkage methods where the coefficient estimates can be shrunk towards zero during model fitting
- Function `vglm` in the `VGAM` library (Yee 2015). It is based on vector generalized additive models, with a wide range of response distributions

Interestingly, an increasing number of packages also fits multinomial models with both fixed and random effects (mixed-effects multinomial models), for example:

- Function `bayesx` in the `R2BayesX` library (Belitz et al. 2017, Umlauf et al. 2015), uses a Bayesian approach for parameter estimation
- Function `npmlt` in the `mixcat` library (Papageorgiou and Hinde 2012)
- Function `clmm` in the `ordinal` library (Christensen 2018), for ordered categorical response variables

Now let's illustrate the use of these functions and how graphical predictions can be obtained.

Generating and plotting predictions from multinomial models

Some of the functions mentioned above can easily be handed over to the `effects` package (Fox 2003, Fox and Hong 2009), where trellis graphs based on the `lattice` library (Sarkar 2008) can be obtained.

```
model1=multinom(Sat ~ Infl + Type + Cont, weights = Freq,
                        data = housing)

model2=multinom(Sat ~ Infl + Type + Cont, data = housing.long)
```

`model1` will not work with the `effects` package. Therefore, we need to use the reshaped `housing.long` dataset:

```
library(effects)
plot(Effect("Infl",model2))
#shows predictions with confidence bands

plot(Effect("Infl",model2),multiline=T)
#shows multiple lines of predictions in the same plot

plot(Effect("Infl",model2),style="stacked")
# shows a stacked vertical bar chart of predictions
```

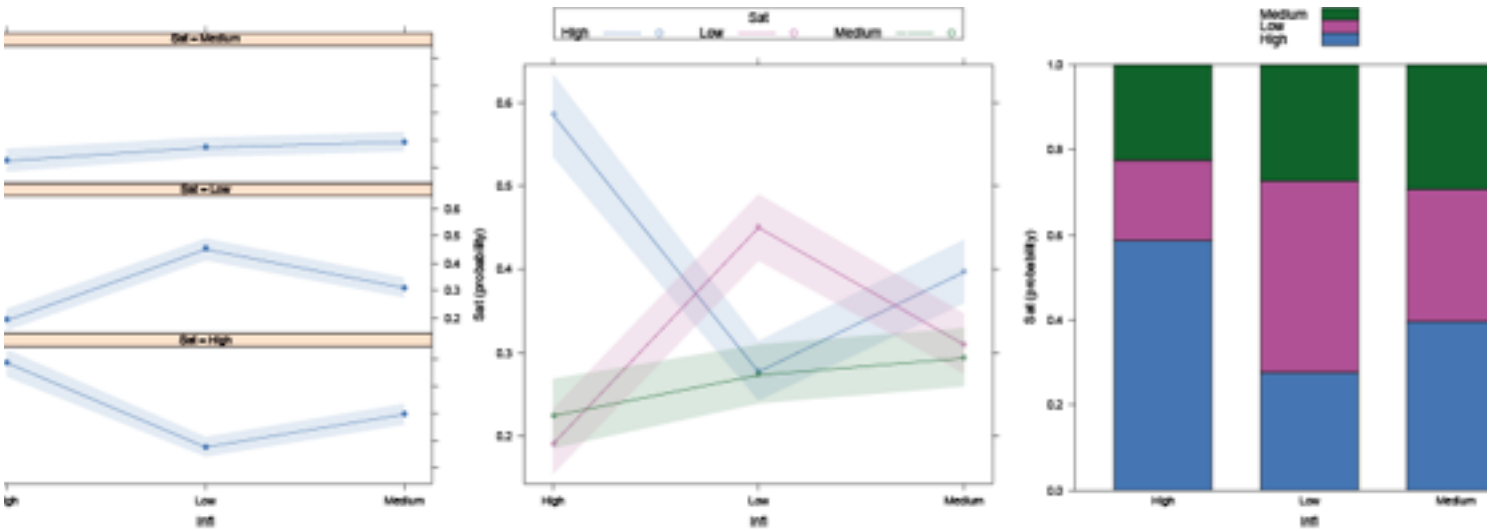

Figure 1: Three different versions of model predictions for model2.

While predictions for multinomial models are straightforward to produce using the `multinom` function in the `nnet` library, this becomes more tedious if other packages are employed. Hence, the results strongly depend on the package and generating predictions is an active area of research into statistical software development.

For models with random effects, the `clmm` function in package `ordinal` will work well with the `effects` package. Most other packages require predictions to be calculated by hand, as illustrated by the following examples:

- Prediction using multinom:

```
model2=multinom(Sat ~ Infl + Type + Cont, data = housing.long)
predict(model2,data.frame(Type="Apartment",Infl="Low",Cont="Hi
gh"))
```

- Prediction using vglm:

```
require(VGAM)
model3=vglm(Sat ~ Infl + Type + Cont, data =
housing.long,family=multinomial())
predict(model3,data.frame(Type="Apartment",Infl="Low",Cont="Hi
gh"))
```

- Prediction using bayesx:

```
require(R2BayesX)
model4=bayesx(Sat ~ Infl + Type + Cont, data =
housing.long,family="multinomial",reference=1)
summary(model4)
# predictions not possible
```

- Prediction using glmnet:

```
require(glmnet)
require(glmnetUtils) #required for formula-based interface

model5=glmnet(Sat ~ Infl + Type + Cont, data =
housing.long,family="multinomial")
predict(model5,data.frame(Type="Apartment",Infl="Low",Cont="Hi
gh"))
```

Real-world applications

Multinomial models are not only common across the sciences but also frequently employed when analyzing traditional “Big Data” problems. A recent application is text mining, where the response variable can be chunks of text grabbed from social media or from the internet. Whenever there is a categorical response variable, these models are the method of choice. If you are interested to read more, I recommend the books by Bilder & Loughin (2014) and Friendly & Meyer (2015).

Conclusion

In summary, a wide range of R functions for multinomial analysis is available. Predictions can (at current) be generated most easily using the `multinom` function in the `nnet` library. However, other packages are catching up and new developments, especially for models containing random effects, can be expected in the near future.

References

Belitz C, Brezger A, Kneib T, Lang S, Umlauf N (2017) BayesX: Software for Bayesian Inference in
Structured Additive Regression Models. Version 1.1. , http://www.BayesX.org/

Bilder CR, Loughin TM (2014) *Analysis of categorical data with R*. CRC Press. (see also the website http://www.chrisbilder.com/)

Christensen RHB (2018) ordinal - Regression Models for Ordinal Data. R package version 2018.4-19.
http://www.cran.r-project.org/package=ordinal/

Fox J (2003) Effect Displays in R for Generalised Linear Models. Journal of Statistical Software, 8(15), 1-27.
http://www.jstatsoft.org/v08/i15/.

Fox J, Hong J (2009). Effect Displays in R for Multinomial and Proportional-Odds Logit Models: Extensions to the effects Package. Journal of Statistical Software, 32(1), 1-24. http://www.jstatsoft.org/v32/i01/.

Friedman J, Haste T, Tibshirani R (2010) Regularization Paths for Generalized Linear Models via Coordinate Descent. Journal of Statistical Software, 33(1), 1-22, http://www.jstatsoft.org/v33/i01/.

Friendly M, Meyer D (2015) Discrete Data Analysis with R: Visualization and Modeling Techniques for Categorical and Count Data. Chapman and Hall/CRC Press.

Papageorgiou G, Hinde J (2012). mixcat: Mixed effects cumulative link and logistic regression models. R package version 1.0-3. https://CRAN.R-project.org/package=mixcat

Sarkar D (2008) Lattice: Multivariate Data Visualization with R. Springer, New York. ISBN 978-0-387-75968-5

Umlauf N, Adler D, Kneib T, Lang S, Zeileis A (2015) Structured Additive Regression Models: An R Interface
to BayesX. Journal of Statistical Software, 63(21), 1-46. http://www.jstatsoft.org/v63/i21/.

Venables WN, Ripley BD (2002) Modern Applied Statistics with S. Springer, New York
Yee TW (2015) Vector Generalized Linear and Additive Models: With an Implementation in R. New York, USA: Springer.

About the Author:

Christoph Scherber is a full professor of Animal Ecology at the Institute of Landscape Ecology, University of Münster, Germany. He received his statistical training with Michael J. Crawley at Imperial College London, Silwood Park, UK. Since then, he has taught statistics to undergraduate, graduate and Ph.D. students and companies in several countries. R code is posted on his personal website (http://www.christoph-scherber.de/statistics.html). He also has a YouTube channel with statistics tutorials with more than 500,000 views at https://www.youtube.com/user/phrygos