Efficient Bayesian Modeling of Binary and Categorical Data in R: The UPG Package

Gregor Zens
Vienna University of
Economics and Business

Sylvia Frühwirth-Schnatter
Vienna University of
Economics and Business

Helga Wagner
Johannes Kepler
University Linz

Abstract

We introduce the **UPG** package for highly efficient Bayesian inference in probit, logit, multinomial logit and binomial logit models. **UPG** offers a convenient estimation framework for balanced and imbalanced data settings where sampling efficiency is ensured through MCMC boosting methods. These convenient theoretical properties are combined with a speedy implementation using **Rcpp**. In addition, **UPG** offers several methods for fast production of output tables and summary plots that are easily accessible to a broad range of users.

1. Introduction

Modeling binary and categorical data is one of the most commonly encountered tasks of applied statisticians and econometricians. Probit and logit models are widely used to model binary outcomes while their extensions to multinomial and binomial regression models also have a permanent place in the toolbox of applied statistics. In this article, we present **UPG**, a new and comprehensive package for Bayesian analysis of well-known binary and categorical data models. **UPG** is based on a number of highly efficient '**U**ltimate **P**ólya **G**amma' MCMC algorithms that have been developed in Frühwirth-Schnatter, Zens, and Wagner (2020). The package features a speedy implementation as well as a number of 'plug&play' solutions to facilitate analysis and communicating results for researchers in a broad range of fields.

In principle, estimation of probit, logit and binomial regression models in R is readily available via **glm** in **stats** (R Core Team 2018). However, due to the immense popularity and broad range of applications of these models, it comes as no surprise that a large number of R packages offer several extensions and modifications. For instance, regularized logistic regression is implemented in several packages such as **CDLasso** (Grant, Lange, and Wu 2013), **clogitL1** (Reid and Tibshirani 2014), **clogitLasso** (Avalos, Pouyes, Kwemou, and Xu 2018), **enetLTS** (Kurnaz, Hoffmann, and Filzmoser 2018), **gren** (Münch, Peeters, van der Vaart, and van de Wiel 2018) or **stepPlr** (Park and Hastie 2018). In addition, some rather specialized applications of logistic regression have been released. These include high dimensional fixed effect models (**FEprovideR**; He and Wu 2019), logistic regression trees (**glmtree**; Ehrhardt 2019), logistic PCA (**logisticPCA**; Landgraf and Lee 2015) and mixtures of logistic regressions (**morpheus**; Auder and Loum 2020). Similarly, a variety of probit modifications may be

found, for instance in the context of instrumental variable analysis (**ivprobit**; Taha 2018) or spatial econometrics (**spatialprobit**; Wilhelm and de Matos 2015).

While most of these packages implement some variant of maximum likelihood estimation, several packages using Bayesian estimation algorithms are available. For example, **HTLR** (Li and Yao 2018) implements Bayesian logit models with heavy-tailed priors, **MHTrajecto-ryR** (Marbac and Sedki 2016) may be used for model selection in logit models, **OncoBayes2** (Weber, Widmer, and Bean 2020) implements Bayesian logit models suited for oncology research and **reglogit** (Gramacy 2018) uses Gibbs sampling to simulate from the posterior of regularized logistic regression models. A related, 'pseudo-Bayesian' approach is taken in the **arm** package (Gelman, Su, Yajima, Hill, Pittau, Kerman, Zheng, and Dorie 2020) that implements an approximate EM algorithm for several Bayesian generalized linear models in command `bayesglm`. **binomlogit** (Fussl 2014) allows the user to estimate a binomial logistic regression model using data augmentation (Fussl, Frühwirth-Schnatter, and Frühwirth 2013), offering one of the extremely few models dealing with binomial logits and the only Bayesian one available so far.

Contrary to probit, logit and binomial logit models, no estimation method for multinomial logit models is available in **stats**. Hence, the user has to rely on externally provided software packages to estimate multinomial logistic regression models in R. Naturally, this has given rise to a number of packages including standard multinomial logit analysis, such as **VGAM** (Yee 2010), **nnet** (Venables and Ripley 2002) or **mlogit** (Croissant 2020). In addition, certain specialized solutions are also available for dealing with multinomial outcomes. These include, among others, **mnlogit** (Hasan, Wang, and Mahani 2016) for fast estimation in large scale settings, **mlogitBMA** (Sevcikova and Raftery 2013) for Bayesian model averaging of multinomial logits as well as **gmnl** (Sarrias and Daziano 2017) for multinomial logits with random coefficients. A more robust version of the multinomial logit model that deals with overdispersed data can be found in **multinomRob** (Mebane, Jr., and Sekhon 2013).

This (certainly non-exhaustive) review of available packages is convincing evidence of the large demand for readily available software solutions when it comes to modeling of binary and categorical data. Aforementioned packages cover a broad set of applications and provide useful specialized tools for many settings. However, a number of shortcomings remain that we seek to tackle through releasing **UPG**. First, **UPG** allows for comprehensive analysis of the four most commonly encountered models for binary and categorical data in *one* simple, unified framework. This solves the problem that, at the moment, researchers may have to switch between packages for different tasks where functionality and usability may vary substantially. In addition, a comprehensive and streamlined coding framework allows to easily extend the basic Bayesian regression framework outlined in this article to include e.g. additional prior settings or more involved model setups in the future. Second, **UPG** is especially well suited for analysis of imbalanced data, where the implemented algorithms make extremely efficient posterior simulation possible. Bayesian analysis of imbalanced data has so far not been the focus of any package released in R (or any other programming language) while being a highly relevant problem in applied statistics (Johndrow, Smith, Pillai, and Dunson 2019; Frühwirth-Schnatter *et al.* 2020). Finally, no Bayesian multinomial logistic regression framework is currently available on CRAN[1], a gap that we are aiming to fill with **UPG**.

---

[1]The former package **bayesCL** has been recently removed from CRAN. **BayesLogit** (Polson, Scott, and Windle 2013a) included multinomial logistic regression in previous versions that have also been removed from CRAN. The updated, currently available version of **BayesLogit** has been reduced to a random number generator

In general, the Bayesian paradigm has a number of pronounced benefits when it comes to estimation of binary and categorical data models. Besides the usual conveniences of Bayesian uncertainty quantification and easy implementation of model extensions, it is well known that Bayesian methods are useful to resolve the issue of *perfect separation*. This phenomenon occurs when a given covariate (quasi-)perfectly separates the outcome variable of interest. To avoid that parameters drift off to $\pm\infty$ in such scenarios, frequentist statistics suggests for instance penalized likelihood methods (Heinze and Schemper 2002). In a Bayesian context, the combination of a potentially 'problematic' likelihood with a weakly informative prior with finite support automatically resolves the issue of perfect separation (Gelman, Jakulin, Pittau, Su *et al.* 2008; Rainey 2016). Tightly related to the occurence of perfect separation are scenarios where certain outcome categories are observed only very rarely or not at all. For similar reasons as outlined above, Bayesian inference helps researchers to avoid implausible parameter estimates in these cases. In fact, **UPG** is especially well suited for estimation in such imbalanced data scenarios, as the implemented algorithms will offer a high level of efficiency even for extremely imbalanced data sets (Frühwirth-Schnatter *et al.* 2020).

Apart from the practical and methodological benefits raised above, **UPG** aims to provide a wide range of functionality in order to be appealing to different groups of R users. First, researchers that are already familiar with Bayesian statistical analysis can easily introduce the underlying MCMC algorithms in **UPG** as an additional sampling block to pre-existing Gibbs sampling algorithms using a few lines of code. This may prove useful in several applications, including mixture-of-experts models (Gormley and Frühwirth-Schnatter 2019) or analysis of Markov switching models (Frühwirth-Schnatter 2006). Second, for a much broader group of users, the package implements several methods for easy and fast production of tables and plots from the estimation output provided. This facilitates analysis also for users that are not commonly working within the Bayesian paradigm.

UPG is licensed under the GNU General Public License 3 and is openly available on the Comprehensive R Archive Network (CRAN, https://cran.r-project.org/package=UPG).

The remainder of this article is structured as follows. Section 2 provides a short overview of the methodology behind **UPG**. Section 3 gives a brief introduction to the package, intended as a quick-start guide. Section 4 presents an extended illustration of the functionality of **UPG**. Finally, Section 5 concludes.

2. Brief methodological overview

This section provides a brief summary of the key insights relevant to understand the inner workings of the models implemented in **UPG**. Most of the contents and ideas are directly taken from Frühwirth-Schnatter *et al.* (2020) where the authors develop the methodology underlying **UPG**. This is also where the reader is refered to for more information and details.

2.1. Binary regression

Binary regression models for a set of N binary data $\mathbf{y} = (y_1, \ldots, y_N)$ are defined by

$$\Pr(y_i = 1|\mathbf{x}_i, \boldsymbol{\beta}) = F_\varepsilon(\mathbf{x}_i\boldsymbol{\beta}). \tag{1}$$

for Pólya Gamma variables.

Choosing the cdf $F_\varepsilon(\varepsilon) = \Phi(\varepsilon)$ of the standard normal distribution leads to the probit model $\Pr(y_i = 1|\mathbf{x}_i, \boldsymbol{\beta}) = \Phi(\mathbf{x}_i\boldsymbol{\beta})$, whereas the cdf $F_\varepsilon(\varepsilon) = e^\varepsilon/(1 + e^\varepsilon)$ of the logistic distribution leads to the logit model

$$\Pr(y_i = 1|\mathbf{x}_i, \boldsymbol{\beta}) = e^{\mathbf{x}_i\boldsymbol{\beta}}/(1 + e^{\mathbf{x}_i\boldsymbol{\beta}}).$$

A latent variable representation of model (1) involving the latent utility z_i is given by:

$$y_i = I\{z_i > 0\}, \qquad z_i = \mathbf{x}_i\boldsymbol{\beta} + \varepsilon_i, \quad \varepsilon_i \sim f_\varepsilon(\varepsilon_i), \tag{2}$$

where $f_\varepsilon(\varepsilon) = F'_\varepsilon(\varepsilon) = \phi(\varepsilon)$ is equal to the standard normal pdf for a probit model and equal to $f_\varepsilon(\varepsilon) = e^\varepsilon/(1 + e^\varepsilon)^2$ for a logit model.

While MCMC estimation based on (2) is straightforward for the probit model using one level of data augmentation involving the latent utilities $(z_1, \ldots, z_N)$ (Albert and Chib 1993), for the logit model a second level of data augmentation is required in addition to z_i, based on a mixture representation of the logistic distribution. In **UPG**, we apply the mixture representation of the logistic distribution from Frühwirth-Schnatter *et al.* (2020),

$$f_\varepsilon(\varepsilon) = e^\varepsilon/(1 + e^\varepsilon)^2 = \frac{1}{4} \int e^{-\omega \varepsilon^2/2} p(\omega) d\omega, \tag{3}$$

where $\omega \sim \mathcal{PG}(2, 0)$ follows a Pólya-Gamma distribution (Polson, Scott, and Windle 2013b) with parameter $b = 2$ and $\kappa = 0$. Most conveniently, $\omega_i|\varepsilon_i$ again follows a Pólya-Gamma distribution which is easy to sample from. Hence, setting up a Gibbs sampling scheme is straightforward.

2.2. Multinomial logistic regression

Let $\{y_i\}$ be a sequence of categorical data, $i = 1, \ldots, N$, where y_i is equal to one of at least three unordered categories. The categories are labeled by $L = \{0, \ldots, m\}$. We assume that the observations are mutually independent and that for each $k \in L$ the probability of y_i taking the value k depends on covariates $\mathbf{x}_i$ in the following way:

$$P(y_i = k|\boldsymbol{\beta}_0, \ldots, \boldsymbol{\beta}_m) = \pi_{ki}(\boldsymbol{\beta}_0, \ldots, \boldsymbol{\beta}_m) = \frac{\exp(\mathbf{x}_i\boldsymbol{\beta}_k)}{\displaystyle\sum_{l=0}^{m} \exp(\mathbf{x}_i\boldsymbol{\beta}_l)}, \tag{4}$$

where $\boldsymbol{\beta}_0, \ldots, \boldsymbol{\beta}_m$ are category specific unknown parameters of dimension d. To make the model identifiable, the parameter $\boldsymbol{\beta}_{k_0}$ of a baseline category k_0 is set equal to $\mathbf{0}$: $\boldsymbol{\beta}_{k_0} = \mathbf{0}$. Thus the parameter $\boldsymbol{\beta}_k$ is in terms of the change in log-odds relative to the baseline category k_0. In the following, we assume without loss of generality that $k_0 = 0$.

UPG uses the aggregated random utility model (aRUM) representation from Frühwirth-Schnatter *et al.* (2020) for estimation. To sample the category specific parameter $\boldsymbol{\beta}_k$, the data is reduced to three categories: category k, the baseline category, and a category which collapses the remaining categories in $A = \{l \in L|l \neq \{k, 0\}\}$. Let u_{ki} denote the latent utility of observation i when choosing category k. For all categories in A, we define an aggregated utility $u_{a,i} = \max_{l \in A} u_{li}$ as the maximal utility in A.

For each specific category $k = 1, \ldots, m$, it is possible to derive the following aRUM representation based on the latent utilities ($u_{ki}, u_{0i}, u_{a,i}$):

$$u_{ki} = \mathbf{x}_i \boldsymbol{\beta}_k + \epsilon_{ki}, \tag{5}$$

$$u_{0i} = \epsilon_{0i},$$

$$u_{a,i} = \mathbf{x}_i \boldsymbol{\beta}_a + \epsilon_{a,i}, \tag{6}$$

$$y_i = \begin{cases} k, & u_{ki} \geq \max(u_{0i}, u_{a,i}), \\ 0, & u_{0i} \geq \max(u_{ki}, u_{a,i}), \\ a \in A, & u_{a,i} \geq \max(u_{ki}, u_{0i}), \end{cases} \tag{7}$$

where the errors $\epsilon_{ki}, \epsilon_{0i}, \epsilon_{a,i}$ are iid from the extreme value distribution.

To derive an efficient sampler to estimate the category specific parameters $\boldsymbol{\beta}_k$ conditional on $u_{a,i}$, it is useful to rewrite (5) in the following way:

$$u_{ki} - u_{0i} = \mathbf{x}_i \boldsymbol{\beta}_k + \varepsilon_{ki}, \tag{8}$$

where $\varepsilon_{ki} \sim \mathcal{LO}$ follows a logistic distribution, independent of $\epsilon_{a,i}$, and the choice equation is the same as in (7). The mixture representation from (3) can then be used to represent the logistic distribution of ε_{ki}, which finally results in a Gibbs sampling scheme for the multinomial logit model that is easy to implement.

2.3. Binomial logistic regression

Finally, **UPG** can handle regression models with binomial outcomes, i.e. models of the form

$$y_i \sim \text{BiNom}(N_i, \pi_i), \qquad \text{logit } \pi_i = \mathbf{x}_i \boldsymbol{\beta}, \qquad i = 1, \ldots, N, \tag{9}$$

where y_i can be interpreted as the number of successes out of N_i trials of individual i. As shown in Frühwirth-Schnatter *et al.* (2020), the binomial model has following random utility representation for $0 < y_i < N_i$:

$$w_i = \mathbf{x}_i \boldsymbol{\beta} + \varepsilon_{w,i}, \quad \varepsilon_{w,i} \sim \mathcal{GL}_{\mathrm{II}}(k), \tag{10}$$

$$v_i = \mathbf{x}_i \boldsymbol{\beta} + \varepsilon_{v,i}, \quad \varepsilon_{v,i} \sim \mathcal{GL}_{\mathrm{I}}(N_i - k),$$

$$y_i = k \Leftrightarrow w_i > 0, v_i < 0,$$

where $\mathcal{GL}_{\mathrm{I}}(\nu)$ and $\mathcal{GL}_{\mathrm{II}}(\nu)$ are, respectively, the generalized logistic distributions of type I and type II. For $y_i = 0$, the model reduces to

$$v_i = \mathbf{x}_i \boldsymbol{\beta} + \varepsilon_{v,i}, \quad \varepsilon_{v,i} \sim \mathcal{GL}_{\mathrm{I}}(N_i), \quad y_i = 0 \Leftrightarrow v_i < 0.$$

For $y_i = N_i$, the model reduces to

$$w_i = \mathbf{x}_i \boldsymbol{\beta} + \varepsilon_{w,i}, \quad \varepsilon_{w,i} \sim \mathcal{GL}_{\mathrm{II}}(N_i), \quad y_i = N_i \Leftrightarrow w_i > 0.$$

For $N_i = 1$, the logistic model results, as both $\mathcal{GL}_{\mathrm{I}}(\nu)$ and $\mathcal{GL}_{\mathrm{II}}(\nu)$ reduce to a logistic distribution for $\nu = 1$. For $y_i = 0$, $z_i = v_i$, whereas for $y_i = 1$, $z_i = w_i$, and the choice equation reduces to $y_i = I\{z_i > 0\}$. To estimate $\boldsymbol{\beta}$ in this framework, it is possible to derive mixture representations similar to (3) for the $\mathcal{GL}_{\mathrm{I}}(\nu)$ and $\mathcal{GL}_{\mathrm{II}}(\nu)$ error distributions, see Frühwirth-Schnatter *et al.* (2020) for details.

2.4. Increasing sampling efficiency through MCMC boosting

It is well known that Bayesian estimation of binary and categorical data models using data augmentation may result in inefficient sampling behavior, especially in settings with imbalanced data (Johndrow *et al.* 2019). The samplers that are outlined in the previous subsections are, in principle, no exemption from this rule. To tackle this issue, **UPG** implements *boosted* MCMC algorithms that have been developed in Frühwirth-Schnatter *et al.* (2020) to enable highly efficient posterior sampling in a broad range of settings. These MCMC boosting methods are similar in spirit to previous work on MCMC sampling efficiency, see for instance Kastner and Frühwirth-Schnatter (2014) or Kastner, Frühwirth-Schnatter, and Lopes (2017) for MCMC boosting in the context of (factor) stochastic volatility models. A general intuition on MCMC boosting is provided in Yu and Meng (2011) for the case of Bayesian multi-level models that allow for competing parametrizations.[2]

The MCMC boosting methods implemented in **UPG** use marginal data augmentation (van Dyk and Meng 2001) to increase sampling efficiency. The boosting strategy involves expanding the latent variable representations from the previous subsections into an unidentified space where efficient sampling of the regression coefficients becomes possible. Applying this strategy resolves two issues that are directly related to sampling inefficiency in data augmentation based samplers. First, the posterior distribution of the coefficients and the posterior distribution of the latent utilities are heavily dependent on each other. In order to decrease this dependency, we re-scale the latent utilities using an unidentified working parameter in the spirit of Imai and van Dyk (2005). Second, when dealing with imbalanced data, the latent utilities will become large in absolute terms. As a result, they will often be located in regions of the respective link function that are extremely insensitive to changes in the latent utilities, furthermore increasing autocorrelation in the posterior samples. To tackle this issue, a second unidentified working parameter that re-centers the latent utilities is introduced during MCMC sampling. Re-scaling and re-centering the latent utilities allows for efficient simulation from the posterior distribution in a wide range of settings, making **UPG** the preferable choice for Bayesian estimation of binary and categorical data models in cases where data is imbalanced.

While this subsection serves as a first intuition of the mechanics behind our boosting strategy, the reader is referred to Frühwirth-Schnatter *et al.* (2020) for a proper methodological introduction to the implemented algorithms. In addition, the report features a number of large-scale simulation studies illustrating the gains in sampling efficiency that can be achieved by using **UPG** for estimation.

2.5. Further details

Estimation of above-mentioned models is based on simulating from conditional posterior distributions in a Gibbs sampling framework. Estimation of binary, multinomial and binomial logit models requires simulating from a Pólya Gamma distribution. This is accomplished using an implementation in C that is taken from **pgdraw** (Makalic and Schmidt 2016).

The (marginal) prior on $\boldsymbol{\beta}$ reads $\boldsymbol{\beta} \sim \mathcal{N}_d \left(0, A_0 \mathbf{I} + G_0 \boldsymbol{e}_d \boldsymbol{e}_d^\top \right)$ where $\boldsymbol{e}_d = \left(\begin{array}{cc} \mathbf{0}_{1 \times (d-1)} & 1 \end{array} \right)^\top$ and d is the index of the intercept term included in $\mathbf{x}_i$. That is, the prior variance for the intercept $G_0 + A_0$ and the prior variance for the remaining coefficients A_0 are specified separately.

---

[2]Note that *MCMC boosting* as outlined in this article is not to be confused with *boosting* in the sense of a meta-algorithm in supervised learning problems as pioneered in Schapire (1990).

Estimation Command	Model
`UPG(y, X, type = "probit")`	Probit
`UPG(y, X, type = "logit")`	Logit
`UPG(y, X, type = "mnl")`	Multinomial Logit
`UPG(y, X, Ni, type = "binomial")`	Binomial Logit

Table 1: Estimation commands for the models included in **UPG**

3. UPG Basics

The **UPG** package provides efficient sampling algorithms for Bayesian analysis of the probit, logit, multinomial logit and binomial logit model. This section covers the basics of the package, including data requirements, estimation as well as the methods included in **UPG**.

In terms of inputs, the minimum requirement for probit, logit and multinomial logit models is a suitable $N \times 1$ dependent vector `y` and a $N \times d$ design matrix `X`. An additional $N \times 1$ vector of total number of trials `Ni` is necessary to estimate a binomial logit model. For probit and logit models, `y` is supposed to be binary. For multinomial logit models, `y` is a categorical vector containing one realized category out of the set $L = \{0, \dots, m\}$ for each observation. The baseline category k_0 can be freely chosen by the user through parameter `baseline`. If no baseline is provided, the most frequently observed category is used as baseline. For binomial logits, `y` contains the number of successes of each observation. Inputs of class `integer`, `numeric`, `matrix` and `data.frame` are accepted. Estimation of multinomial logits additionally accepts `character` and `factor` types as dependent vector. Depending on the specified model type, **UPG** will use a variety of data checks to ensure proper estimation.

All necessary tools for efficient estimation of discrete choice models in a Gibbs sampling framework are wrapped into a single estimation function UPG() to facilitate use and keep the user interface as minimal and intuitive as possible. The samplers are written in C++, implemented via **Rcpp** (Eddelbuettel and François 2011) and **RcppArmadillo** (Eddelbuettel and Sanderson 2014). Hence, the provided estimation framework combines sampling efficiency through MCMC boosting with a speedy implementation. The four different models included in **UPG** can be called using the `type` parameter as shown in Table 1. An illustration of the estimation process and the most important posterior analysis methods using **UPG** are discussed in the next section.

In terms of output, UPG will return one out of four S3 objects, depending on the estimated model specified using `type`. The possible classes are UPG.Probit, UPG.Logit, UPG.MNL and UPG.Binomial. These objects hold the full posterior distribution for all parameters and latent variables involved during estimation. In addition, all user inputs are duplicated into the output object for further analysis. Several S3 methods that can be applied to any of these objects. The main task of these methods is to conveniently summarize the generated posterior samples. The methods themselves are summarized in Table 2 and will be discussed in further detail in the subsequent section using extensive examples.

4. Analyzing binary and categorical data using UPG

This section provides information on the data sets that are included in **UPG** and uses these data sets as running examples to demonstrate the functionality of the package.

S3 Method	Usage
print	Object overview
summary	Summary of posterior estimates as well table output
plot	Plot coefficient point estimates and credible intervals
predict	Predict probabilities for new data or input data
coef	Extract posterior means and credible intervals of coefficients
logLik	Extract log-likelihood based on posterior mean

Table 2: S3 methods included in **UPG**

4.1. Bayesian binary regression: Probit and Logit

To demonstrate how to estimate and analyze Bayesian probit and logit models using **UPG**, a microeconomic data set on female labor force particpation from the US *Panel Study of Income Dynamics* is included. It features a binary variable indicating labor force status as well as a number of additional covariates for 753 women.

```
R> data("lfp", package = "UPG")
R> head(lfp, 5)
```

```
  lfp intercept k5 k618       age wc hc       lwg   inc
1   1         1  1    0 -1.3053889  0  0 1.2101647 10.91
2   1         1  1    0 -1.5531414  0  0 0.3285041 19.50
3   1         1  1    1 -0.9337602  0  0 1.5141279 12.04
4   1         1  1    0 -1.0576365  0  0 0.0921151  6.80
5   1         1  1    2 -1.4292651  1  0 1.5242802 20.10
```

The binary dependent variable lfp takes the value of 1 if the woman is participating in the labor force. k5 gives the number of children under the age of 5, k618 indicates the number of children between 6 and 8 years, age is a standardized age index and wc as well as hc are binary indicators capturing whether a college degree was obtained by the wife and the husband, respectively. In addition, two income related predictors are included, where lwg describes the expected log wage of the woman and inc gives the logarithm of family income exclusive of the income of the woman. This data set comes from the **carData** package and has been originally analyzed in Mroz (1987).

Model estimation

To construct a suitable design matrix X and a binary dependent vector y for probit and logit models, it suffices to split the data set as follows:

```
R> y <- lfp[, 1]
R> X <- lfp[, -1]
```

In order to estimate a Bayesian logit model, one line of code is sufficient:

```
R> results.logit <- UPG(y = y, X = X, type = "logit")
```

```
Checking data & inputs ...
Initializing Gibbs Sampler ...
Simulating from posterior distribution ...
0%   10   20   30   40   50   60   70   80   90  100%
[----|----|----|----|----|----|----|----|----|----|
*************************************************|
Sampling succesful!
Saving output ...
Finished! Sampling took 2.52 CPU seconds.
```

In the remainder of this subsection, it is assumed that the goal is to estimate and analyze a logit model using `type = 'logit'`. Changing the `type` parameter to `type = 'probit'` allows to estimate a probit model. The discussion below holds for both types of models.

Tabulating results

Applying `summary` to the output object results in a quick overview of the regression results in the form of tabulated parameter estimates. Continuing the running example, it is easy to generate a table with posterior means and standard deviations as well as credible intervals:

```
R> summary(results.logit)


--- Bayesian Logit Results ---

N = 753
Analysis based on 1000 posterior draws after a burn-in period of 1000 iterations.
MCMC sampling took a total of 2.52 CPU seconds.
```

		Mean	SD	Q2.5	Q97.5	95% CI excl. 0
intercept	0.50	0.24	0.05	1.01	*
k5	-1.44	0.18	-1.80	-1.11	*
k618	-0.06	0.07	-0.19	0.07	
age	-0.50	0.10	-0.70	-0.30	*
wc	0.76	0.22	0.33	1.19	*
hc	0.13	0.21	-0.27	0.53	
lwg	0.60	0.15	0.31	0.90	*
inc	-0.03	0.01	-0.05	-0.02	*

In terms of interpretation, it is for instance visible that women with a college degree (`wc`) are more likely to participate in the labor force compared to women with no formal tertiary education, holding everything else constant. On the contrary, women who have small children under the age of 5 (`k5`) are *ceteris paribus* less likely to be active in the labor force compared to women without young children.

A number of possibilities for exporting summary tables to LaTeX, HTML or Microsoft Word

using `summary(obj, type = c("html","latex","pandoc"))` is available.[3] The user can choose from a number of different options to customize the table output directly, including the upper and lower bounds for the credible intervals based on posterior quantiles specified using `q`, the names of the variables using `names`, the number of significant digits using `digits`, the subset of variables to be used in the table using `include` and the table caption using `cap`. Further customizations are easy to implement as `summary` returns a `knitr_kable` object that can be further modified using the **kable** package. More details can be found in the package documentation.

Visualizing results

In case a more visual representation of the model output is desired, the `plot` function can be used to generate publication-ready coefficient plots for all four available models using **ggplot2** (Wickham 2016). Similar to the `summary` function, `plot` allows the user to customize a number of pre-specified parameters such as axis labels (`xlab`, `ylab`), coefficient names (`names`), the width of the credible intervals (`q`), and the set of included variables (`include`). `plot` will return a ggplot2 object that can be further modified using the complete arsenal of tools from the ggplot2 universe.

Continuing the logit example from above, we can generate a simple coefficient plot using

```
R> plot(results.logit)
```
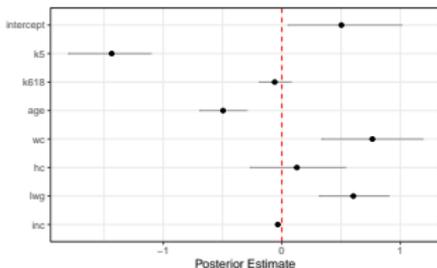

These plots provide point estimates as well as credible intervals for each covariate by default. The variables may be sorted by estimated effect size using `sort = TRUE`. Otherwise, they appear in the same order as in `X`.

Predicting probabilities

In several situations, applied researchers are not necessarily interested in examining the estimated coefficients, but in using these estimates to generate predictions. For these scenarios, `predict` may be used to produce point estimates and credible intervals of predicted probabilities based on the estimated model. These predictions can be generated using either the

---

[3]If a LaTeX table is desired, the 'booktabs' package has to be included in the preamble of the LaTeX document.

data provided for model estimation or new, external data provided by the user. Continuing the running example,

```
R> predict(results.logit)
```

will return a list containing the posterior mean as well as the 97.5% and 2.5% posterior quantile of the predicted probabilities for the data used to estimate the model. In case the user wants to predict probabilities from external data, a suitable explanatory matrix X.new with the same number of columns and same variable ordering must be provided to compute valid predictions. The syntax in that case is

```
R> predict(results.logit, newdata = X.new).
```

Similar to the other available S3 methods in **UPG**, the credible intervals can be specified by the user using the parameter q.

Log-likelihood

In case the user is interested in the log-likelihood of the data given the parameters, a logLik method is available. Applying this method to the output will extract the log-likelihood evaluated at the posterior mean of the parameters:

```
R> logLik(results.logit)
```

```
'log Lik.' -452.666 (df=8)
```

This log-likelihood object holds information on the number of observations as well as the number of estimated parameters.

4.2. Bayesian binomial logistic regression

To demonstrate how to estimate binomial logit model using **UPG**, aggregated individual passenger data of the *RMS Titanic* is included as an example data set:

```
R> data("titanic", package = "UPG")
R> head(titanic, 5)
```

```
  survived total intercept pclass female age.group
1        0     1         1      1      1         5
2        5     5         1      2      1         5
3       12    17         1      3      1         5
4        2     2         1      1      0         5
5        8     8         1      2      0         5
```

The passengers have been split into several groups that are based on passenger class (pclass), five year age groups (age.group) and gender (female). For each group, total passenger counts (total) and the number of passengers that survived the disaster (survived) are provided.

The data set is an aggregate version of the well-known titanic data set (Hilbe 2007, Table 6.11) that has for instance been previously analyzed in Frühwirth-Schnatter, Frühwirth, Held, and Rue (2009) and is often used in machine learning competitions.[4]

Model estimation

In this case, the dependent vector of successes is `survived` whereas the number of total trials corresponds to `total`. Both vectors have to be provided in addition to some explanatory variables to be able to estimate a binomial logit model using **UPG**. Hence, the data needs to be split into three parts parts prior to estimation:

```
R>  y  <- titanic[,1]
R>  Ni <- titanic[,2]
R>  X  <- titanic[,-c(1,2)]
R>  results.binomial <- UPG(y = y, X = X, Ni = Ni, type = "binomial")


Checking data & inputs ...
Initializing Gibbs Sampler ...
Simulating from posterior distribution ...
0%   10   20   30   40   50   60   70   80   90  100%
[----|----|----|----|----|----|----|----|----|----|
**************************************************|
Sampling succesful!
Saving output ...
Finished! Sampling took 2.06 CPU seconds.
```

All further steps of analysis are similar to the directions given in the previous subsection on probit and logit models. As an example, assume we would like to tabulate the data using a credible interval based on the 10% and 90% posterior quantiles:

```
R>  summary(results.binomial, q = c(0.1, 0.9))


--- Bayesian Binomial Logit Results ---

N = 78
Analysis based on 1000 posterior draws after a burn-in period of 1000 iterations.
MCMC sampling took a total of 2.06 CPU seconds.
```

	Mean	SD	Q10	Q90	80% CI excl. 0
intercept	2.30	0.40	1.77	2.81	*
pclass	-1.20	0.12	-1.35	-1.05	*
female	2.51	0.18	2.27	2.74	*
age.group	-0.03	0.01	-0.04	-0.03	*

---

[4]See https://www.kaggle.com/c/titanic/ for more details.

In terms of interpreting these results, we can for instance see that female passengers have had a much higher survival probability compared to their male counterparts on *MS Titanic*. A higher passenger class (corresponding to cheaper tickets) is associated with higher mortality. Finally, the log-odds of survival decrease with increasing age.

To demonstrate how to change the credible intervals that should be plotted, the estimation output is visualized using `q = c(0.1, 0.9)`. This results in a 80% credible interval based on the 0.1 and 0.9 quantiles of the posterior distribution. In addition, custom variable names are provided and `sort = TRUE` ensures that the variables are ordered based on estimated effect size:

```
R>  plot(results.binomial,
R>       sort = TRUE,
R>       q    = c(0.1, 0.9),
R>       names = c("Intercept", "Passenger Class", "Female", "Age Group"))
```
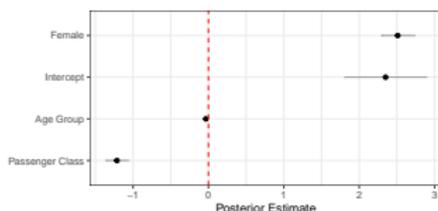

4.3. Bayesian multinomial logistic regression

For the multinomial logit model, a data set on 200 high school students and their program choice (general, vocational or academic) is included together with a binary variable taking the value of 1 for female students (`female`), a categorical variable indicating socio-economic status (`ses`) and standardized results of a writing test (`write`):

```
R>  data("program",package="UPG")
R>  head(program,5)
```

```
  program intercept female ses      write
1 vocation         1      1   1 -1.875280
2  general         1      0   2 -2.086282
3 vocation         1      0   3 -1.453276
4 vocation         1      0   1 -1.664278
5 vocation         1      0   2 -2.297284
```

This data set is also known as the `hsbdemo` data set and is provided online by the University of California, Los Angeles Statistical Consulting Group. It is widely used in several R packages and in other software tools as example data for multinomial logistic regression.[5]

---

[5]See for instance https://stats.idre.ucla.edu/stata/dae/multinomiallogistic-regression/ for usage of the data in Stata.

Model estimation

As mentioned above, dependent variables for multinomial logit estimation have to be provided as a categorical vector. By default, the category that occurs most often is chosen as baseline category. An alternative baseline category may be specified using `baseline`. In the example data set, `academic` is chosen 105 times out of 200 observations and will thus serve as baseline category. The code to create `y` and `X` is quite similar to the probit and logit case:

```
R> y <- program[,1]
R> X <- program[,-1]
```

To estimate a multinomial logit model, `type = 'mnl'` has to be specified when using the `UPG` command:

```
R>  results.mnl <- UPG(y = y, X = X, type = 'mnl', verbose = FALSE)
```

where we have set `verbose = FALSE` to suppress any output during estimation for illustration purposes. Handling the resulting `UPG.MNL` object is similar to the cases outlined above and is thus only discussed briefly. Tabulation of the results is based on a grouped representation of the model output:

```
R> summary(results.mnl,
R>         names = c("Intercept", "Female", "SES", "Writing Score"))
```

```
--- Bayesian Multinomial Logit Results ---

N = 200
Analysis based on 1000 posterior draws after a burn-in period of 1000 iterations.
MCMC sampling took a total of 1.58 CPU seconds.

Category 'academic' is the baseline category.
```

		Mean	SD	Q2.5	Q97.5	95% CI excl. 0
Category 'general'					
Intercept	0.32	0.58	-0.87	1.42	
Female	0.06	0.36	-0.65	0.76	
SES	-0.57	0.25	-1.06	-0.07	*
Writing Score	-0.57	0.19	-0.93	-0.19	*
Category 'vocation'					
Intercept	-0.42	0.65	-1.67	1.02	
Female	0.47	0.38	-0.24	1.23	
SES	-0.36	0.28	-0.96	0.16	
Writing Score	-1.14	0.23	-1.58	-0.68	*

From the output, it becomes obvious that in the observed sample higher scores on a writing test decrease the probability of choosing a general or vocational program compared to the baseline of choosing an academic program. Similar conclusions can be drawn from a coefficient plot that will automatically be grouped by outcome in case of the multinomial logit model:

```
plot(results.mnl,
     names = c("Intercept", "Female", "SES", "Writing Score"))
```
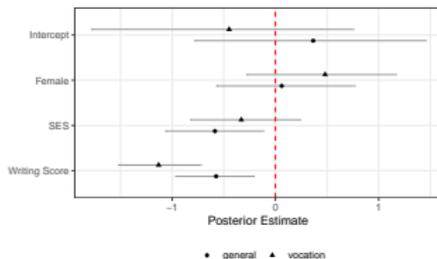

4.4. UPG-within-Gibbs

In certain applications, more advanced users might desire to use **UPG** as a single sampling block within a pre-existing Gibbs sampler. Examples where this might be useful include mixture-of-experts models, where a multinomial logit prior might be implemented (see e.g. Gormley and Frühwirth-Schnatter 2019). Similarly, probits, logits and multinomial logits do often serve as prior models in Bayesian Markov switching frameworks (Frühwirth-Schnatter 2006).

To implement 'UPG-within-Gibbs' it suffices to set up the estimation command UPG such that it generates only a single draw from the posterior distribution conditional on the inputs. Consider the example of a binary logit model. Assuming a suitable starting value for `beta.draw` is given, it suffices to use

```
results.logit  <- UPG(y          = y,
                       X          = X,
                       type       = "logit",
                       draws      = 1,
                       burnin     = 0,
                       beta.start = beta.draw,
                       verbose    = F)

beta.draw      <-  results.logit$posterior$beta.post
```

as sampling block in an existing Gibbs sampler. In this example, draws is set to 1 and burnin is set to 0 to generate exactly one posterior sample without burn-in period. verbose = F

suppresses all console output and parameter `beta.start` is used to provide the current value of `beta.draw` as starting value. Iterating over this code M times and saving the resulting posterior draws of `beta` gives equivalent results to generating M posterior draws from UPG directly.

However, note that this will, in general, be slower than generating M samples from UPG directly. This is due to the substantial overhead that results from repeated function calls in R. Specifically, generating M posterior samples from UPG directly results in exactly one function call, while 'UPG-within-Gibbs' will amount to M function calls. However, even in this 'slow' scenario, **UPG** allows for rather fast sampling. For instance, the 'within-Gibbs' logistic regression model outlined above samples more than 270 posterior draws per second using the example data with $N = 753$ and eight covariates on an Intel i7 CPU @ 2.40GHz. A more detailed picture on sampling efficiency and sampling speed is provided in the next subsection.

4.5. Sampling efficiency and sampling speed

In order to shed some light on the performance of the implemented models in specific applications, the user can compute several MCMC diagnostic measures using the command `UPG.Diag`. Specifically, a call to `UPG.Diag` will return the effective sample size (ESS) for each coefficient derived using `effectiveSize` from **coda** (Plummer, Best, Cowles, and Vines 2006).[6] In addition, inefficiency factors (IE; given by the number of saved draws divided by the effective sample size) and the effective sampling rate (ESR; given by the effective sample size divided by the running time of the sampler in seconds) are returned. To allow for a more convenient 'quick check' of the behavior of the Markov chain, `UPG.Diag` also returns the minimum, maximum and median across all coefficients as summary statistics of these three diagnostic measures.

To provide some performance measures and to give a sense of magnitude in terms of expected computation time, we summarize ESS, IE and ESR for probit as well as binary, binomial and multinomial logit models using the example data sets that come with **UPG**. For each model, 10,000 posterior draws are sampled after an initial burn-in period of 1,000 iterations. All simulations have been run on an Intel i7 CPU @ 2.40GHz. The results of this exercise are shown in Table 3. While the table shows that the MCMC algorithms in **UPG** exhibit rather efficient sampling behavior, a pronounced drop in sampling *speed* is visible when comparing the probit regression framework to the remaining models. This is due to the increased computational effort that results from sampling Pólya Gamma random variables. While these are not needed in the MCMC scheme of the probit model, they are necessary for all logit models in **UPG**, increasing computation time in each sweep of the sampler. Nevertheless, due to high levels of sampling *efficiency*, an effective posterior sample size that is sufficient for inference can be generated in a speedy fashion in the logit frameworks as well. In general, adding a few thousand more effective posterior samples in medium-sized data sets will be possible in significantly less than a minute.

5. Conclusion

In this article, the R package **UPG** is introduced as a software tool for Bayesian estimation of

---

[6]Effective sample sizes in **coda** are derived from the spectral density at 0 which is estimated based on fitting an autoregressive process to the posterior draws.

Table 3: Sampling efficiency and sampling speed of the implemented models. Results are based on 10,000 saved draws after an initial burn-in period of 1,000 iterations using example data sets as input.

		Probit	Binary Logit	Multinomial Logit	Binomial Logit
	N	753	753	200	78
	d	8	8	4	4
	$\sum_i N_i$				887
ESS	Min.	3051	2692	1719	3288
	Median	3755	3002	2068	3487
	Max.	4064	3239	2215	4379
IE	Min.	2.46	3.09	4.52	2.28
	Median	2.66	3.33	4.84	2.87
	Max.	3.28	3.71	5.82	3.04
ESR	Min.	1185	165	162	255
	Median	1458	184	195	271
	Max.	1578	198	209	340
Time (in sec.)		2.57	16.36	10.60	12.87

probit, logit, multinomial and binomial logit models. In addition to an implementation that enables highly efficient estimation through boosted MCMC algorithms, the package is designed to provide easy access to Bayesian models for binary and categorical data for researchers that might not be familiar with the Bayesian paradigm. For statisticians that are used to working with Bayesian models, the package allows for a range of customizations and for convenient integration into existing Gibbs samplers.

The package includes a variety of functions that may be used to quickly produce tables and plots that summarize the estimation output. These methods have been introduced and illustrated through applied examples using data sets that come with the package in order to provide some guidance on how to apply the package. In terms of implementation, the MCMC algorithms in **UPG** are written in C++, resulting in highly efficient and speedy Bayesian estimation methods for binary and categorical data models that are easily accessible for a broad range of users.

Acknowledgments

References

Albert JH, Chib S (1993). "Bayesian analysis of binary and polychotomous response data." *Journal of the American Statistical Association*, **88**, 669–679.

Auder B, Loum MA (2020). *morpheus: Estimate Parameters of Mixtures of Logistic Regressions*. R package version 1.0-1, URL https://CRAN.R-project.org/package=morpheus.

Avalos M, Pouyes H, Kwemou M, Xu B (2018). *clogitLasso: Sparse Conditional Logistic Regression for Matched Studies*. R package version 1.1, URL https://CRAN.R-project.org/package=clogitLasso.

Croissant Y (2020). *mlogit: Multinomial Logit Models*. R package version 1.0-3.1, URL https://CRAN.R-project.org/package=mlogit.

Eddelbuettel D, François R (2011). "Rcpp: Seamless R and C++ Integration." *Journal of Statistical Software*, **40**(8), 1–18. doi:10.18637/jss.v040.i08. URL http://www.jstatsoft.org/v40/i08/.

Eddelbuettel D, Sanderson C (2014). "RcppArmadillo: Accelerating R with high-performance C++ linear algebra." *Computational Statistics and Data Analysis*, **71**, 1054–1063. URL http://dx.doi.org/10.1016/j.csda.2013.02.005.

Ehrhardt A (2019). *glmtree: Logistic Regression Trees*. R package version 0.1, URL https://CRAN.R-project.org/package=glmtree.

Frühwirth-Schnatter S (2006). *Finite mixture and Markov switching models*. Springer Science & Business Media.

Frühwirth-Schnatter S, Frühwirth R, Held L, Rue H (2009). "Improved auxiliary mixture sampling for hierarchical models of non-Gaussian data." *Statistics and Computing*, **19**(4), 479.

Frühwirth-Schnatter S, Zens G, Wagner H (2020). "Ultimate Pólya Gamma Samplers – Efficient MCMC for possibly imbalanced binary and categorical data." *arXiv Preprint*. 2011.06898.

Fussl A (2014). *binomlogit: Efficient MCMC for Binomial Logit Models*. R package version 1.2, URL https://CRAN.R-project.org/package=binomlogit.

Fussl A, Frühwirth-Schnatter S, Frühwirth R (2013). "Efficient MCMC for binomial logit models." *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, **23**(1), 1–21.

Gelman A, Jakulin A, Pittau MG, Su YS, *et al.* (2008). "A weakly informative default prior distribution for logistic and other regression models." *The Annals of Applied Statistics*, **2**(4), 1360–1383.

Gelman A, Su YS, Yajima M, Hill J, Pittau MG, Kerman J, Zheng T, Dorie V (2020). "Package 'arm'."

Gormley IC, Frühwirth-Schnatter S (2019). "Mixture of experts models." In S Frühwirth-Schnatter, G Celeux, CP Robert (eds.), *Handbook of Mixture Analysis*, chapter 12, pp. 271–307. CRC Press, Boca Raton, FL.

Gramacy RB (2018). *reglogit: Simulation-Based Regularized Logistic Regression*. R package version 1.2-6, URL https://CRAN.R-project.org/package=reglogit.

Grant E, Lange K, Wu TT (2013). *CDLasso: Coordinate Descent Algorithms for Lasso Penalized L1, L2, and Logistic Regression*. R package version 1.1, URL https://CRAN.R-project.org/package=CDLasso.

Hasan A, Wang Z, Mahani AS (2016). "Fast Estimation of Multinomial Logit Models: R Package mnlogit." *Journal of Statistical Software*, **75**(3), 1–24. doi:10.18637/jss.v075.i03.

He K, Wu W (2019). *FEprovideR: Fixed Effects Logistic Model with High-Dimensional Parameters*. R package version 1.1, URL https://CRAN.R-project.org/package=FEprovideR.

Heinze G, Schemper M (2002). "A solution to the problem of separation in logistic regression." *Statistics in Medicine*, **21**, 2409–2419.

Hilbe JM (2007). *Negative binomial regression*. Cambridge University Press.

Imai K, van Dyk DA (2005). "A Bayesian Analysis of the Multinomial Probit Model Using Marginal Data Augmentation." *Journal of Econometrics*, **124**, 311–334.

Johndrow JE, Smith A, Pillai N, Dunson DB (2019). "MCMC for imbalanced categorical data." *Journal of the American Statistical Association*, **114**(527), 1394–1403.

Kastner G, Frühwirth-Schnatter S (2014). "Ancillarity-sufficiency interweaving strategy (ASIS) for boosting MCMC estimation of stochastic volatility models." *Computational Statistics & Data Analysis*, **76**, 408–423.

Kastner G, Frühwirth-Schnatter S, Lopes HF (2017). "Efficient Bayesian inference for multivariate factor stochastic volatility models." *Journal of Computational and Graphical Statistics*, **26**(4), 905–917.

Kurnaz FS, Hoffmann I, Filzmoser P (2018). *enetLTS: Robust and Sparse Methods for High Dimensional Linear and Logistic Regression*. R package version 0.1.0, URL https://CRAN.R-project.org/package=enetLTS.

Landgraf AJ, Lee Y (2015). "Dimensionality Reduction for Binary Data through the Projection of Natural Parameters." *Technical Report 890*, Department of Statistics, The Ohio State University. URL http://arxiv.org/abs/1510.06112.

Li L, Yao W (2018). "Fully Bayesian logistic regression with hyper-LASSO priors for high-dimensional feature selection." *Journal of Statistical Computation and Simulation*, **88**(14), 2827–2851.

Makalic E, Schmidt DF (2016). "High-dimensional Bayesian regularised regression with the BayesReg package." *arXiv preprint arXiv:1611.06649*.

Marbac M, Sedki M (2016). *MHTrajectoryR: Bayesian Model Selection in Logistic Regression for the Detection of Adverse Drug Reactions*. R package version 1.0.1, URL https://CRAN.R-project.org/package=MHTrajectoryR.

Mebane WR, Jr, Sekhon JS (2013). *multinomRob: Robust Estimation of Overdispersed Multinomial Regression Models*. R package version 1.8-6.1, URL https://CRAN.R-project.org/package=multinomRob.

Mroz TA (1987). "The sensitivity of an empirical model of married women's hours of work to economic and statistical assumptions." *Econometrica*, **55**, 765–799.

Münch M, Peeters C, van der Vaart A, van de Wiel M (2018). "The Spectral Condition Number Plot for Regularization Parameter Determination." *arXiv:1805.00389v1 [stat.ME]*.

Park MY, Hastie T (2018). *stepPlr: L2 Penalized Logistic Regression with Stepwise Variable Selection*. R package version 0.93, URL https://CRAN.R-project.org/package=stepPlr.

Plummer M, Best N, Cowles K, Vines K (2006). "CODA: Convergence Diagnosis and Output Analysis for MCMC." *R News*, **6**(1), 7–11. URL https://journal.r-project.org/archive/.

Polson NG, Scott JG, Windle J (2013a). "Bayesian inference for logistic models using Polya-Gamma latent variables." Most recent version: Feb. 2013., URL http://arxiv.org/abs/1205.0310.

Polson NG, Scott JG, Windle J (2013b). "Bayesian inference for logistic models using Pólya-Gamma latent variables." *Journal of the American Statistical Association*, **108**, 1339–49.

R Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Rainey C (2016). "Dealing with Separation in Logistic Regression Models." *Political Analysis*, **24**, 339–355. doi:10.1093/pan/mpw014.

Reid S, Tibshirani R (2014). "Regularization Paths for Conditional Logistic Regression: The clogitL1 Package." *Journal of Statistical Software*, **58**(12), 1–23. URL http://www.jstatsoft.org/v58/i12/.

Sarrias M, Daziano R (2017). "Multinomial Logit Models with Continuous and Discrete Individual Heterogeneity in R: The gmnl Package." *Journal of Statistical Software*, **79**(2), 1–46. doi:10.18637/jss.v079.i02.

Schapire RE (1990). "The strength of weak learnability." *Machine learning*, **5**(2), 197–227.

Sevcikova H, Raftery A (2013). *mlogitBMA: Bayesian Model Averaging for Multinomial Logit Models*. R package version 0.1-6, URL https://CRAN.R-project.org/package=mlogitBMA.

Taha Z (2018). *ivprobit: Instrumental Variables Probit Model*. R package version 1.1, URL https://CRAN.R-project.org/package=ivprobit.

van Dyk D, Meng XL (2001). "The art of data augmentation." *Journal of Computational and Graphical Statistics*, **10**, 1–50.

Venables WN, Ripley BD (2002). *Modern Applied Statistics with S*. Fourth edition. Springer, New York. ISBN 0-387-95457-0, URL http://www.stats.ox.ac.uk/pub/MASS4.

Weber S, Widmer LA, Bean A (2020). *OncoBayes2: Bayesian Logistic Regression for Oncology Dose-Escalation Trials*. R package version 0.6-5, URL https://CRAN.R-project.org/package=OncoBayes2.

Wickham H (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. ISBN 978-3-319-24277-4. URL http://ggplot2.org.

Wilhelm S, de Matos MG (2015). *spatialprobit: Spatial Probit Models*. R package version 0.9-11, URL https://CRAN.R-project.org/package=spatialprobit.

Yee TW (2010). "The VGAM Package for Categorical Data Analysis." *Journal of Statistical Software*, **32**(10), 1–34. URL http://www.jstatsoft.org/v32/i10/.

Yu Y, Meng XL (2011). "To center or not to center: That is not the question – An Ancillarity–Sufficiency Interweaving Strategy (ASIS) for boosting MCMC efficiency." *Journal of Computational and Graphical Statistics*, **20**(3), 531–570.

Affiliation:

Gregor Zens
Department of Economics
Vienna University of Economics and Business
1020 Vienna, Austria
E-mail: gzens@wu.ac.at
URL: https://www.wu.ac.at/vw1/m/gregor-zens-msc