Journal of Statistical Software

May 2015, Volume 65, Issue 5. *http://www.jstatsoft.org/*

The VGAM Package for Capture-Recapture Data Using the Conditional Likelihood

Thomas W. Yee
The University of Auckland

Jakub Stoklosa
The University of New South Wales

Richard M. Huggins
The University of Melbourne

Abstract

It is well known that using individual covariate information (such as body weight or gender) to model heterogeneity in capture-recapture (CR) experiments can greatly enhance inferences on the size of a closed population. Since individual covariates are only observable for captured individuals, complex conditional likelihood methods are usually required and these do not constitute a standard generalized linear model (GLM) family. Modern statistical techniques such as generalized additive models (GAMs), which allow a relaxing of the linearity assumptions on the covariates, are readily available for many standard GLM families. Fortunately, a natural statistical framework for maximizing conditional likelihoods is available in the Vector GLM and Vector GAM classes of models. We present several new R functions (implemented within the **VGAM** package) specifically developed to allow the incorporation of individual covariates in the analysis of closed population CR data using a GLM/GAM-like approach and the conditional likelihood. As a result, a wide variety of practical tools are now readily available in the **VGAM** object oriented framework. We discuss and demonstrate their advantages, features and flexibility using the new **VGAM** CR functions on several examples.

Keywords: closed population size estimation, conditional likelihood, mark-capture-recapture, vector generalized additive model, **VGAM**.

1. Introduction

Capture-recapture (CR) surveys are widely used in ecology and epidemiology to estimate population sizes. In essence they are sampling schemes that allow the estimation of both n and p in a Binomial(n, p) experiment (Huggins and Hwang 2011). The simplest CR sampling design consists of units or individuals in some population that are captured or tagged across several sampling occasions, e.g., trapping a nocturnal mammal species on seven consecutive

nights. In these experiments, when an individual is captured for the first time then it is marked or tagged so that it can be identified upon subsequent recapture. On each occasion recaptures of individuals which have been previously marked are also noted. Thus each observed individual has a capture history: a vector of 1s and 0s denoting capture/recapture and noncapture respectively. The unknown population size is then estimated using the observed capture histories and any other additional information collected on captured individuals, such as weight or sex, along with environmental information such as rainfall or temperature. For further details on CR experiments as a whole see McCrea and Morgan (2014).

We consider closed populations, where there are no births, deaths, emigration or immigration throughout the sampling period (Amstrup, McDonald, and Manly 2005). Such an assumption is often reasonable when the overall time period is relatively short. Otis, Burnham, White, and Anderson (1978) provided eight specific closed population CR models (see also Pollock 1991), which permit the individual capture probabilities to depend on time and behavioral response, and be heterogeneous between individuals. The use of covariate information (or explanatory variables) to explain heterogeneous capture probabilities in CR experiments has received considerable attention over the last 30 years (Pollock 2002). Population size estimates that ignore this heterogeneity typically result in biased population estimates (Amstrup *et al.* 2005).

Since individual covariate information (such as gender or body weight) can only be collected on observed individuals, conditional likelihood models are employed (Pollock, Hines, and Nichols 1984; Huggins 1989; Alho 1990; Lebreton, Burnham, Clobert, and Anderson 1992). That is, one conditions on the individuals seen at least once through-out the experiment, hence they allow for individual covariates to be considered in the analysis. The capture probabilities are typically modeled as logistic regression functions of the covariates, and parameters are estimated using maximum likelihood. Importantly, these CR models are generalized linear models (GLMs; McCullagh and Nelder 1989; Huggins and Hwang 2011).

Here, we maximize the conditional likelihood (or more formally the positive-Bernoulli distribution) models of Huggins (1989). This approach has become standard practice to carry out inferences when considering individual covariates, with several different software packages currently using this methodology, including: **MARK** (Cooch and White 2012), **CARE-2** (Hwang and Chao 2003), and the R packages (R Core Team 2015): **mra** (McDonald 2012), **RMark** (Laake 2013) and **Rcapture** (Baillargeon and Rivest 2007, 2014), the latter package uses a log-linear approach, which can be shown to be equivalent to the conditional likelihood (Cormack 1989; Huggins and Hwang 2011). For a more detailed review of CR software see Bunge (2013). These programs are quite user friendly, and specifically, allow modeling capture probabilities as linear functions of the covariates. So an obvious question is to ask why develop yet another implementation for closed population CR modeling?

Firstly, nonlinearity arises quite naturally in many ecological applications, (Schluter 1988; Yee and Mitchell 1991; Crawley 1993; Gimenez, Covas, Brown, Anderson, Brown, and Lenormand 2006; Bolker 2008). In the CR context, capture probabilities may depend nonlinearly on individual covariates, e.g., mountain pygmy possums with lighter or heavier body weights may have lower capture probabilities compared with those having mid-ranged body weights (e.g., Huggins and Hwang 2007; Stoklosa and Huggins 2012). However, in our experience, a majority of CR software does not handle nonlinearity well in regard to both estimation and in the plotting of the smooth functions. Since GAMs (Hastie and Tibshirani 1990; Wood 2006) were developed in the mid-1980s they have become a standard tool for data analysis in

regression. The nonlinear relationship between the response and covariate is flexibly modeled, and few assumptions are made on the functional relationship. The drawback in applying these models to CR data has been the difficult programming required to implement the approach.

Secondly, we have found several implementations of conditional likelihood slow, and in some instances unreliable and difficult to use. We believe that our implementation has superior capabilities, and has good speed and reliability. The results of Section 6.5 contrast our software with some others. Moreover, the incorporation of these methods in a general, maintained statistical package will result in them being updated as the package is updated.

Standard GLM and GAM methodologies are unable to cope with the CR models considered in this article because they are largely restricted to one linear/additive predictor η. Fortunately however, a natural extension in the form of the vector generalized linear and additive model (VGLM/VGAM) classes do allow for multiple ηs. VGAMs and VGLMs are described in Yee and Wild (1996) and Yee and Hastie (2003). Their implementation in the **VGAM** package (Yee 2008, 2010, 2014) has become increasingly popular and practical over the last few years, due to the large number of exponential families available for discrete/multinomial response data. Package **VGAM** is available from the Comprehensive R Archive Network (CRAN) at `http://CRAN.R-project.org/package=VGAM`. In addition to flexible modeling of both VGLMs and VGAMs, a wide range of useful features are also available:

- smoothing capabilities;
- model selection using, say, AIC or BIC (Burnham and Anderson 1999);
- regression diagnostics and goodness-of-fit tools;
- reduced-rank regression (Yee and Hastie 2003) for dimension reduction;
- computational speed and robustness;
- choice of link functions;
- offsets and prior weights; and
- (specifically) when using R: generic functions based on object oriented programming, e.g., `fitted()`, `coef()`, `vcov()`, `summary()`, `predict()`, `AIC()`, etc.

Our goal is to provide users with an easy-to-use object-oriented **VGAM** structure, where four `family`-type functions based on the conditional likelihood are available to fit the eight models of Otis *et al.* (1978). We aim to give the user additional tools and features, such as those listed above, to carry out a more informative and broader analysis of CR data; particularly when considering more than one covariate. Finally, this article primarily focuses on the technical aspects of the proposed package, and less so on the biological interpretation for CR experiments.

An outline of this article is as follows. In Section 2 we present the conditional likelihood for CR models and a description of the eight Otis *et al.* (1978) models. Section 3 summarizes pertinent details of VGLMs and VGAMs. Their connection to the CR models is made in Section 4. Software details are given in Section 5, and examples on real and simulated data using the new software are demonstrated in Section 6. Some final remarks are given in

Symbol	Explanation
N	(Closed) population size to be estimated.
n	Total number of distinct individuals caught in the trapping experiment.
τ	Number of sampling occasions, where $\tau \geq 2$.
$\boldsymbol{y}_i$	Vector of capture histories for individual i ($i = 1, \ldots, n$) with observed values 1 (captured) and 0 (noncaptured). Each $\boldsymbol{y}_i$ has at least one observed 1.
"h"	Model $\mathcal{M}$ subscript, for heterogeneity.
"b"	Model $\mathcal{M}$ subscript, for behavioral effects.
"t"	Model $\mathcal{M}$ subscript, for temporal effects.
p_{ij}	Probability that individual i is captured at sampling occasion j ($j = 1, \ldots, \tau$).
z_{ij}	$= 1$ if individual i has been captured before occasion j, else $= 0$.
$\boldsymbol{\theta}$	Vector of regression coefficients to be estimated related to p_{ij}.
$\boldsymbol{\eta}$	Vector of linear predictors (see Table 3 for further details).
g	Link function applied to, e.g., p_{ij}. Logit by default.

Table 1: Short summary of the notation used for the positive-Bernoulli distribution for capture-recapture (CR) experiments. Additional details are in the text.

Section 7. The two appendices give some technical details relating to the first and second derivatives of the conditional log-likelihood, and the means.

2. Capture-recapture models

In this section we give an outline for closed population CR models under the conditional likelihood/GLM approach. For further details we recommend Huggins (1991) and Huggins and Hwang (2011). The notation of Table 1 is used throughout this article.

2.1. Conditional likelihood

Suppose we have a closed population of N individuals, labeled $i = 1, \ldots, N$ and τ capture occasions labeled $j = 1, \ldots, \tau$. We make the usual assumptions that individuals in the population behave independently of each other, individuals do not lose their tags, and tags are recorded correctly. Let $y_{ij} = 1$ if the ith individual was caught on the jth occasion and be zero otherwise, and let n be the number of distinct individuals captured.

Let p_{ij} denote the probability of capturing individual i on occasion j. As noted in Section 1, Otis *et al.* (1978) describe eight models for the capture probabilities, see Section 2.2 for further details. Label the individuals captured in the experiment by $i = 1, \ldots, n$ and those never captured by $i = n+1, \ldots, N$. The full likelihood is given by

$$\begin{aligned} L_f &= K \prod_{i=1}^{N} \prod_{j=1}^{\tau} p_{ij}^{y_{ij}} (1 - p_{ij})^{1-y_{ij}} \\ &= K \left\{ \prod_{i=1}^{n} \prod_{j=1}^{\tau} p_{ij}^{y_{ij}} (1 - p_{ij})^{1-y_{ij}} \right\} \cdot \left\{ \prod_{i=n+1}^{N} \prod_{j=1}^{\tau} (1 - p_{ij}) \right\} \end{aligned} \tag{1}$$

where K is independent of the p_{ij} but may depend on N. The RHS of (1) requires knowledge of the uncaptured individuals and in general cannot be computed. Consequently no MLE

Capture history	Joint probability			
	$\mathcal{M}_0/\mathcal{M}_h$	$\mathcal{M}_b/\mathcal{M}_{bh}$	$\mathcal{M}_t/\mathcal{M}_{th}$	$\mathcal{M}_{tb}/\mathcal{M}_{tbh}$
01	$(1-p)p$	$(1-p_c)\,p_c$	$(1-p_1)p_2$	$(1-p_{c1})\,p_{c2}$
10	$p(1-p)$	$p_c(1-p_r)$	$p_1(1-p_2)$	$p_{c1}(1-p_{r2})$
11	p^2	$p_c\,p_r$	$p_1\,p_2$	$p_{c1}\,p_{r2}$
00	$(1-p)^2$	$(1-p_c)^2$	$(1-p_1)(1-p_2)$	$(1-p_{c1})(1-p_{c2})$
$M \equiv \dim(\boldsymbol{\eta})$	1	2	2 $(=\tau)$	3 $(=2\tau-1)$

Table 2: Capture history sample space and corresponding probabilities for the eight models of Otis *et al.* (1978), with $\tau = 2$ capture occasions in a closed population CR experiment. Here, p_{cj} = capture probability for unmarked individuals at sampling period j, p_{rj} = recapture probability for marked individuals at sampling period j, and p = constant capture probability across $\tau = 2$. Note that the "00" row is never realized in sample data.

of N will be available unless some homogeneity assumption is made about the noncaptured individuals. Instead, a conditional likelihood function based only on the individuals observed at least once is

$$L_c \quad \propto \quad \prod_{i=1}^{n} \frac{\prod_{j=1}^{\tau} p_{ij}^{y_{ij}} (1-p_{ij})^{1-y_{ij}}}{1-\prod_{s=1}^{\tau}(1-p_{is}^{\dagger})}. \tag{2}$$

Here $p_{is}^{\dagger}$ are the p_{ij} computed as if the individual had not been captured prior to j so that the denominator is the probability individual i is captured at least once. This conditional likelihood (2) is a modified version of the likelihood corresponding to a positive-Bernoulli distribution (Patil 1962).

2.2. The eight models

Models which allow capture probabilities to depend on one or a combination of time, heterogeneity or behavioral effects are defined using appropriate subscripts, e.g., $\mathcal{M}_{th}$ depends on time and heterogeneity. These eight models have a nested structure of which $\mathcal{M}_{tbh}$ is the most general. The homogeneous model $\mathcal{M}_0$ is the simplest (but most unrealistic) and has equal capture probabilities for each individual $H_0 : p_{ij} = p$, regardless of the sampling occasion. All eight models are GLMs, since the conditional likelihood (2) belongs to the exponential family (Huggins and Hwang 2011).

To illustrate the approach, we use the following toy example throughout: Consider a CR experiment with two occasions – morning and evening (i.e., $\tau = 2$), with capture probabilities varying between the two occasions. Furthermore, suppose we have collected some individual covariates – weight and gender. The joint probabilities of all the eight models are listed in Table 2. It can be seen that all but the positive-Binomial model ($\mathcal{M}_0/\mathcal{M}_h$) require more than one probability and hence more than one linear predictor, so that the original Nelder and Wedderburn (1972) GLM framework is inadequate. Further, there are two noteworthy points from Table 2 which apply for any $\tau \geq 2$:

- first, for $\mathcal{M}_t$-type models, as τ increases so will the number of linear predictors and hence the potential number of parameters;

- secondly, it is evident that there are four main categories consisting of non-heterogeneity models ($\mathcal{M}_0$, $\mathcal{M}_b$, $\mathcal{M}_t$ and $\mathcal{M}_{tb}$), which are paired with a heterogeneity sub-model (respectively $\mathcal{M}_h$, $\mathcal{M}_{bh}$, $\mathcal{M}_{th}$ and $\mathcal{M}_{tbh}$).

The four heterogeneity models allow for each individual to have their own probability of capture/recapture. In our toy example, the capture probabilities are dependent on an individual's weight and gender. We discuss these models further in Section 3.1. It is natural to consider individual covariates such as weight and gender as linear/additive predictors. Let x_i denote a covariate (either continuous or discrete) for the ith individual, which is constant across the capture occasions $j = 1, \ldots, \tau$, e.g., for continuous covariates one could use the first observed value or the mean across all j. If there are $d-1$ covariates, we write $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{id})^\top$ with $x_{i1} = 1$ if there is an intercept. Also, let $g^{-1}(\eta) = \exp(\eta)/\{1+\exp(\eta)\}$ be the inverse logit function. Consider model $\mathcal{M}_{tbh}$, then the capture/recapture probabilities are given as (notation follows Section 3.3)

$$\begin{aligned} p_{ij}^\dagger &= g^{-1}\Big(\beta_{(j+1)1}^* + \boldsymbol{x}_{i[-1]}^\top \boldsymbol{\beta}_{1[-1]} \Big), \quad j = 1, \ldots, \tau, \\ p_{ij} &= g^{-1}\Big(\beta_{(1)1}^* + \beta_{(j+1)1}^* + \boldsymbol{x}_{i[-1]}^\top \boldsymbol{\beta}_{1[-1]} \Big), \quad j = 2, \ldots, \tau, \end{aligned}$$

where $\beta_{(1)1}^*$ is the behavioral effect of prior capture, $\beta_{(j+1)1}^*$ for $j = 1, \ldots, \tau$ are time effects, and $\boldsymbol{\beta}_{1[-1]}$ are the remaining regression parameters associated with the covariates. Computationally, the conditional likelihood (2) is maximized with respect to all the parameters (denoted by $\boldsymbol{\theta}$) by the Fisher scoring algorithm using the derivatives given in Appendix A.

2.3. Estimation of N

In the above linear models, to estimate N let $\pi_i(\boldsymbol{\theta}) = 1 - \prod_{s=1}^{\tau} (1 - p_{is}^\dagger)$ be the probability that individual i is captured at least once in the course of the study. Then, if $\boldsymbol{\theta}$ is known, the Horvitz-Thompson (HT) estimator (Horvitz and Thompson 1952) is given by

$$\widehat{N}(\boldsymbol{\theta}) = \sum_{i=1}^{n} \pi_i(\boldsymbol{\theta})^{-1}, \tag{3}$$

which is an unbiased estimator for the population size N and an associated estimate of the variance of $\widehat{N}(\boldsymbol{\theta})$ is $s^2(\boldsymbol{\theta}) = \sum_{i=1}^{n} \pi_i(\boldsymbol{\theta})^{-2} [1 - \pi_i(\boldsymbol{\theta})]$. If $\boldsymbol{\theta}$ is estimated by $\widehat{\boldsymbol{\theta}}$ then one can use

$$\widehat{\mathrm{VAR}}\left(\widehat{N}(\widehat{\boldsymbol{\theta}})\right) \approx s^2(\widehat{\boldsymbol{\theta}}) + \widehat{\boldsymbol{\mathcal{D}}}^\top \widehat{\mathrm{VAR}}(\widehat{\boldsymbol{\theta}})\, \widehat{\boldsymbol{\mathcal{D}}} \tag{4}$$

where, following from a Taylor series expansion of $\widehat{N}(\widehat{\boldsymbol{\theta}})$ about $\widehat{N}(\boldsymbol{\theta})$,

$$\begin{aligned} \boldsymbol{\mathcal{D}} = \frac{dN(\boldsymbol{\theta})}{d\boldsymbol{\theta}} &= \sum_{i=1}^{n} \pi_i(\boldsymbol{\theta})^{-2} \frac{d\pi_i(\boldsymbol{\theta})}{d\boldsymbol{\theta}} \\ &= \sum_{i=1}^{n} \frac{-1}{\pi_i(\boldsymbol{\theta})^2} \sum_{s=1}^{\tau} \left[\prod_{t=1,\ t\neq s}^{\tau} \left(1 - p_{it}^\dagger\right) \right] \frac{\partial p_{is}^\dagger}{\partial \boldsymbol{\theta}}. \end{aligned}$$

3. Vector generalized linear and additive models

To extend the above linear models, we use VGLMs and VGAMs which we briefly describe in this section. These models fit within a large statistical regression framework which is described in Yee (2015). The details here are purposely terse; readers are directed to Yee (2008, 2010) for accessible overviews and examples, and Yee and Wild (1996) and Yee and Hastie (2003) for technical details.

3.1. Basics

Consider observations on independent pairs $(\boldsymbol{x}_i, \boldsymbol{y}_i)$, $i = 1, \ldots, n$. We use "[−1]" to delete the first element, e.g., $\boldsymbol{x}_{i[-1]} = (x_{i2}, \ldots, x_{id})^\top$. For simplicity, we will occasionally drop the subscript i and simply write $\boldsymbol{x} = (x_1, \ldots, x_d)^\top$. Consider a single observation where $\boldsymbol{y}$ is a Q-dimensional vector. For the CR models of this paper, $Q = \tau$ when the response is entered as a matrix of 0s and 1s, the only exception is for the $\mathcal{M}_0/\mathcal{M}_h$ where the aggregated counts may be inputted, see Section 5.2. VGLMs are defined through the model for the conditional density

$$f(\boldsymbol{y}|\boldsymbol{x}; \mathbf{B}) \;=\; f(\boldsymbol{y}, \eta_1, \ldots, \eta_M)$$

for some known function $f(\cdot)$, where $\mathbf{B} = (\boldsymbol{\beta}_1\, \boldsymbol{\beta}_2\, \cdots\, \boldsymbol{\beta}_M)$ is a $d \times M$ matrix of regression coefficients to be estimated. We may also write $\mathbf{B}^\top = (\boldsymbol{\beta}_{(1)}\, \boldsymbol{\beta}_{(2)}\, \cdots\, \boldsymbol{\beta}_{(d)})$ so that $\boldsymbol{\beta}_j$ is the jth column of $\mathbf{B}$ and $\boldsymbol{\beta}_{(k)}$ is the kth row.

The jth linear predictor is then

$$\eta_j \;=\; \boldsymbol{\beta}_j^\top \boldsymbol{x} \;=\; \sum_{k=1}^{d} \beta_{(j)k}\, x_k, \quad j = 1, \ldots, M, \tag{5}$$

where $\beta_{(j)k}$ is the kth component of $\boldsymbol{\beta}_j$. In the CR context, we remind the reader that, as in Table 2, we have $M = 2$ for $\mathcal{M}_{bh}$, $M = \tau$ for $\mathcal{M}_{th}$ and $M = 2\tau - 1$ for $\mathcal{M}_{tbh}$.

In GLMs the linear predictors are used to model the means. The η_j of VGLMs model the parameters of a model. In general, for a parameter θ_j we take

$$\eta_j \;=\; g_j(\theta_j), \quad j = 1, \ldots, M,$$

and we say g_j is a parameter link function. Write

$$\boldsymbol{\eta}_i \;=\; \begin{pmatrix} \eta_1(\boldsymbol{x}_i) \\ \vdots \\ \eta_M(\boldsymbol{x}_i) \end{pmatrix} \;=\; \mathbf{B}^\top \boldsymbol{x}_i \;=\; \begin{pmatrix} \boldsymbol{\beta}_1^\top \boldsymbol{x}_i \\ \vdots \\ \boldsymbol{\beta}_M^\top \boldsymbol{x}_i \end{pmatrix}. \tag{6}$$

In practice we may wish to constrain the effect of a covariate to be the same for some of the η_j and to have no effect for others. In our toy example, model $\mathcal{M}_{th}$ with $\tau = M = 2$, $d = 3$, we have

$$\begin{aligned} \eta_1(\boldsymbol{x}_i) &= \beta_{(1)1} + \beta_{(1)2}\, x_{i2} + \beta_{(1)3}\, x_{i3}, \\ \eta_2(\boldsymbol{x}_i) &= \beta_{(2)1} + \beta_{(2)2}\, x_{i2} + \beta_{(2)3}\, x_{i3}. \end{aligned}$$

If x_{i2} corresponds to being the individual's weight and x_{i3} is an indicator of gender and we have the constraints $\beta_{(1)2} \equiv \beta_{(2)2}$ and $\beta_{(1)3} \equiv \beta_{(2)3}$, then with "*" denoting the parameters that are estimated,

$$\begin{aligned}\eta_1(\boldsymbol{x}_i) &= \beta^*_{(1)1} + \beta^*_{(1)2}\, x_{i2} + \beta^*_{(1)3}\, x_{i3},\\ \eta_2(\boldsymbol{x}_i) &= \beta^*_{(2)1} + \beta^*_{(1)2}\, x_{i2} + \beta^*_{(1)3}\, x_{i3},\end{aligned}$$

and we may write

$$\begin{aligned}\boldsymbol{\eta}(\boldsymbol{x}_i) = \begin{pmatrix}\eta_1(\boldsymbol{x}_i)\\ \eta_2(\boldsymbol{x}_i)\end{pmatrix} &= \sum_{k=1}^{3} \boldsymbol{\beta}_{(k)}\, x_{ik}\\ &= \begin{pmatrix}\beta_{(1)1} & \beta_{(1)2} & \beta_{(1)3}\\ \beta_{(2)1} & \beta_{(2)2} & \beta_{(2)3}\end{pmatrix}\begin{pmatrix}x_{i1}\\ x_{i2}\\ x_{i3}\end{pmatrix}\\ &= \begin{pmatrix}\beta^*_{(1)1} & \beta^*_{(1)2} & \beta^*_{(1)3}\\ \beta^*_{(2)1} & \beta^*_{(1)2} & \beta^*_{(1)3}\end{pmatrix}\begin{pmatrix}x_{i1}\\ x_{i2}\\ x_{i3}\end{pmatrix}\\ &= \begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}\beta^*_{(1)1}\\ \beta^*_{(2)1}\end{pmatrix} x_{i1} + \begin{pmatrix}1\\ 1\end{pmatrix}\beta^*_{(1)2}\, x_{i2} + \begin{pmatrix}1\\ 1\end{pmatrix}\beta^*_{(1)3}\, x_{i3}\\ &= \sum_{k=1}^{3} \mathbf{H}_k\, \boldsymbol{\beta}^*_{(k)}\, x_{ik}.\end{aligned}$$

We can also write this as (noting that $x_{i1} = 1$)

$$\begin{aligned}\boldsymbol{\eta}(\boldsymbol{x}_i) &= \begin{pmatrix}x_{i1} & 0\\ 0 & x_{i1}\end{pmatrix}\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}\beta^*_{(1)1}\\ \beta^*_{(2)1}\end{pmatrix} + \begin{pmatrix}x_{i2} & 0\\ 0 & x_{i2}\end{pmatrix}\begin{pmatrix}1\\ 1\end{pmatrix}\beta^*_{(1)2} + \begin{pmatrix}x_{i3} & 0\\ 0 & x_{i3}\end{pmatrix}\begin{pmatrix}1\\ 1\end{pmatrix}\beta^*_{(1)3}\\ &= \sum_{k=1}^{3} \mathrm{diag}(x_{ik}, x_{ik})\, \mathbf{H}_k\, \boldsymbol{\beta}^*_{(k)}.\end{aligned}$$

In general, for VGLMs, we represent the models as

$$\begin{aligned}\boldsymbol{\eta}(\boldsymbol{x}_i) &= \sum_{k=1}^{d} \boldsymbol{\beta}_{(k)}\, x_{ik}\\ &= \sum_{k=1}^{d} \mathbf{H}_k\, \boldsymbol{\beta}^*_{(k)}\, x_{ik} \qquad (7)\\ &= \sum_{k=1}^{d} \mathrm{diag}(x_{ik}, \ldots, x_{ik})\, \mathbf{H}_k\, \boldsymbol{\beta}^*_{(k)}\end{aligned}$$

where $\mathbf{H}_1, \mathbf{H}_2, \ldots, \mathbf{H}_d$ are known constraint matrices of full column-rank (i.e., the rank equals $\texttt{ncol}(\mathbf{H}_k)$), and $\boldsymbol{\beta}^*_{(k)}$ is a vector containing a possibly reduced set of regression coefficients. Then we may write

$$\mathbf{B}^\top = \begin{pmatrix}\mathbf{H}_1\boldsymbol{\beta}^*_{(1)} & \mathbf{H}_2\boldsymbol{\beta}^*_{(2)} & \cdots & \mathbf{H}_d\boldsymbol{\beta}^*_{(d)}\end{pmatrix} \qquad (8)$$

as an expression of (6) concentrating on columns rather than rows. Note that with no constraints at all, all $\mathbf{H}_k = \mathbf{I}_M$ and $\boldsymbol{\beta}^*_{(k)} = \boldsymbol{\beta}_{(k)}$. We need both (6) and (8) since we focus on the η_j and at other times on the variables x_k. The constraint matrices for common models are pre-programmed in **VGAM** and can be set up by using arguments such as `parallel` and `zero` found in **VGAM** family functions. Alternatively, there is the argument `constraints` where they may be explicitly inputted. Using `constraints` is less convenient but provides the full generality of its power.

3.2. Handling time-varying covariates

Often, the covariates may be time-varying, e.g., when using temperature as a covariate, then a different value is observed and measured for each occasion j for $j = 1, \ldots, \tau$. Again, using our toy example with $M = 2$, $d = 3$, and $\tau = 2$, suppose we have time-dependent covariates $\boldsymbol{x}_{ij}$, $j = 1, 2$. We may have the model

$$\begin{aligned} \eta_1(\boldsymbol{x}_{i1}) &= \beta^*_{(1)1} + \beta^*_{(1)2}\, x_{i21} + \beta^*_{(1)3}\, x_{i31}, \\ \eta_2(\boldsymbol{x}_{i2}) &= \beta^*_{(2)1} + \beta^*_{(1)2}\, x_{i22} + \beta^*_{(1)3}\, x_{i32}, \end{aligned}$$

for the linear predictor on the two occasions. Here, x_{ikt} is for the ith animal, kth explanatory variable and tth time. We write this model as

$$\begin{aligned} \boldsymbol{\eta}(\boldsymbol{x}_{ij}) &= \begin{pmatrix} x_{i11} & 0 \\ 0 & x_{i12} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \beta^*_{(1)1} \\ \beta^*_{(2)1} \end{pmatrix} + \begin{pmatrix} x_{i21} & 0 \\ 0 & x_{i22} \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \beta^*_{(1)2} + \begin{pmatrix} x_{i31} & 0 \\ 0 & x_{i32} \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \beta^*_{(1)3} \\ &= \sum_{k=1}^{3} \operatorname{diag}(x_{ik1}, x_{ik2})\, \mathbf{H}_k\, \boldsymbol{\beta}^*_{(k)}. \end{aligned}$$

Thus to handle time-varying covariates one needs the `xij` facility of **VGAM** (e.g., see Section 6.3), which allows a covariate to have different values for different η_j through the general formula

$$\boldsymbol{\eta}(\boldsymbol{x}_{ij}) = \sum_{k=1}^{d} \operatorname{diag}(x_{ik1}, \ldots, x_{ikM})\, \mathbf{H}_k\, \boldsymbol{\beta}^*_{(k)} = \sum_{k=1}^{d} \mathbf{X}^{\#}_{(ik)} \mathbf{H}_k\, \boldsymbol{\beta}^*_{(k)} \tag{9}$$

where x_{ikj} is the value of variable x_k for unit i for η_j. The derivation of (9), followed by some examples are given in Yee (2010). Implementing this model requires specification of the diagonal elements of the matrices $\mathbf{X}^*_{ik}$ and we see its use in Section 6.3. Clearly, a model may include a mixture of time-dependent and time-independent covariates. The model is then specified through the constraint matrices $\mathbf{H}_k$ and the covariate matrices $\mathbf{X}^{\#}_{(ik)}$. Typically in CR experiments, the time-varying covariates will be environmental effects. Fitting time-varying individual covariates requires some interpolation when an individual is not captured and is beyond the scope of the present work.

3.3. VGAMs

VGAMs replace the linear functions in (7) by smoothers such as splines. Hence, the central formula is

$$\boldsymbol{\eta}_i \;=\; \sum_{k=1}^{d} \mathbf{H}_k \, \boldsymbol{f}_k^*(x_{ik}) \tag{10}$$

where $\boldsymbol{f}_k^*(x_k) = (f_{k(1)}^*(x_k), \ldots, f_{k(M_k)}^*(x_k))^\top$ is a vector of M_k smooth functions of x_k, where $M_k = \texttt{ncol}(\mathbf{H}_k)$ is the rank of the constraint matrix for x_k. Note that standard error bands are available upon plotting the estimated component functions (for details see Yee and Wild 1996), e.g., see Figure 1.

4. VGLMs and VGAMs applied to CR data

In this section we merge the results of Sections 2 and 3 to show how the eight models of Otis *et al.* (1978) can be fitted naturally within the VGLM/VGAM framework.

4.1. Linear predictors and constraint matrices

As in Section 3.1, we now write $\boldsymbol{y}_i$ as the capture history vector for individual i. Written technically, $\boldsymbol{y}_i \in (\{0,1\})^\tau \backslash \{\mathbf{0}_\tau\}$ so that there is at least one 1 (capture). For simplicity let p_c and p_r be the capture and recapture probabilities. Recall that the value for M will depend on the CR model type and the number of capture occasions considered in the experiment, for example, consider model $\mathcal{M}_b$ as in Table 2, then $(\eta_1, \eta_2) = (g(p_c), g(p_r))$ for some link function g, thus $M = 2$. The upper half of Table 3 gives these for the eight Otis *et al.* (1978) models. The lower half of Table 3 gives the names of the **VGAM** family function that fits those models. They work very similarly to the `family` argument of `glm()`, e.g.,

```
vglm(cbind(y1, y2, y3, y4, y5, y6) ~ weight + sex + age,
  family = posbernoulli.t, data = pdata)
```

is a simple call to fit a $\mathcal{M}_{th}$ model. The response is a matrix containing 0 and 1 values only, and three individual covariates are used here. The argument name `family` was chosen for not necessitating `glm()` users learning a new argument name; and the concept of error distributions as for the GLM class does not carry over for VGLMs. Indeed, `family` denotes some full-likelihood specified statistical model worth fitting in its own right regardless of an 'error distribution' which may not make sense. Each family function has `logit()` as their default link, however, alternatives such as `probit()` and `cloglog()` are also permissible. Section 5 discusses the software side of **VGAM** in detail, and Section 6 gives more examples.

As noted above, constraint matrices are used to simplify complex models, e.g., model $\mathcal{M}_{tbh}$ into model $\mathcal{M}_{th}$. The default constraint matrices for the $\mathcal{M}_{tbh}(\tau)$ model are given in Table 4. These are easily constructed using the `drop.b`, `parallel.b` and `parallel.t` arguments in the family function. More generally, the $\mathbf{H}_k$ may be inputted using the `constraints` argument – see Yee (2008) and Yee (2010) for examples. It can be seen that the building blocks of the $\mathbf{H}_k$ are $\mathbf{1}$, $\mathbf{0}$, $\mathbf{I}$ and $\mathbf{O}$. This is because one wishes to constrain the effect of x_k to be the same for capture and recapture probabilities. In general, we believe the $\mathbf{H}_k$ in conjunction with (9) can accommodate all linear constraints between the estimated regression coefficients $\widehat{\beta}_{(j)k}$.

For time-varying covariate models, the M diagonal elements x_{ikj} in (9) correspond to the value of covariate x_k at time j for individual i. These are inputted successively in order using the `xij` argument, e.g., as in Section 6.3.

4.2. Penalized likelihood and smoothing parameters

For each covariate x_k, the smoothness of each component function $f^*_{(j)k}$ in (10) can be controlled by the non-negative smoothing parameters $\lambda_{(j)k}$. Yee and Wild (1994) show that, when vector splines are used as the smoother, the penalized conditional log-likelihood

$$\ell_p \equiv \log L_p = \ell_c - \frac{1}{2}\sum_{k=1}^{d}\sum_{j=1}^{\texttt{ncol}(\mathbf{H}_k)} \lambda_{(j)k}\int_{a_k}^{b_k}\left\{f^{*\prime\prime}_{(j)k}(t)\right\}^2 \mathrm{d}t \tag{11}$$

is maximized. Here, ℓ_c is the logarithm of the conditional likelihood function (2). The penalized conditional likelihood (11) is a natural extension of the penalty approach described in Green and Silverman (1994) to models with multiple η_j.

An important practical issue is to control for overfitting and smoothness in the model. The `s()` function used within `vgam()` signifies the smooth functions $f^*_{(j)k}$ estimated by vector splines. It contains the argument `spar` for the smoothing parameters, where a relatively small (positive) value will give much flexibility and wiggliness. As `spar` increases the solution converges to the least squares estimate. More commonly, the argument `df` (which is known as the equivalent degrees of freedom (EDF)) has also been implemented and can be alternatively used. A value of unity for `df` gives a linear fit, and the default (which is set to `df=4`) affords a reasonable amount of flexibility.

5. Software details for CR models in VGAM

Having presented the conditional likelihood (2) and VGLMs/VGAMs for CR models, we further discuss the fitting in **VGAM**. It is assumed that users are somewhat familiar with modeling in R and using `glm()` class objects. **VGAM**, authored by TWY, uses S4 classes. In order to present the new `family` functions developed for `vglm()` and `vgam()`, some additional preliminaries for **VGAM** are given below. Version 0.9-4 or later is assumed, and the latest prerelease version is available at `http://www.stat.auckland.ac.nz/~yee/VGAM/prerelease`.

In `vglm()`/`vgam()`, both $\mathcal{M}_0$ and $\mathcal{M}_h$ are serviced by `family = posbinomial()`, i.e., the positive-binomial family. For models $\mathcal{M}_b$, $\mathcal{M}_t$ and $\mathcal{M}_{tb}$, each of these are serviced by their corresponding `family = posbernoulli.`-type functions as in Table 3. Formulas given in the form `~ 1` correspond to $\mathcal{M}_0$, $\mathcal{M}_b$, $\mathcal{M}_t$ and $\mathcal{M}_{tb}$; otherwise they are $\mathcal{M}_h$, $\mathcal{M}_{bh}$,$\mathcal{M}_{th}$ and $\mathcal{M}_{tbh}$.

Below we describe each of the eight models with their **VGAM** representation and their default values, we also give additional remarks. All eight models can of course be fitted using `posbernoulli.tb()`, however it is generally not recommended as it is less efficient in terms of memory requirements and speed.

5.1. Basic software details

All family functions except `posbinomial()` should have a $n \times \tau$ capture history matrix as the response, preferably with column names. Indicators of the past capture of individual i,

Model	$\boldsymbol{\eta}^\top$
$\mathcal{M}_0/\mathcal{M}_h$	$g(p)$
$\mathcal{M}_b/\mathcal{M}_{bh}$	$(g(p_c), g(p_r))$
$\mathcal{M}_t/\mathcal{M}_{th}$	$(g(p_1), \ldots, g(p_\tau))$
$\mathcal{M}_{tb}/\mathcal{M}_{tbh}$	$(g(p_{c1}), \ldots, g(p_{c\tau}), g(p_{r2}), \ldots, g(p_{r\tau}))$

Model	`family =`
$\mathcal{M}_0/\mathcal{M}_h$	`posbinomial(omit.constant = TRUE)`
	`posbernoulli.b(drop.b = FALSE ~ 0)`
	`posbernoulli.t(parallel.t = FALSE ~ 0)`
	`posbernoulli.tb(drop.b = FALSE ~ 0, parallel.t = FALSE ~ 0)`
$\mathcal{M}_b/\mathcal{M}_{bh}$	`posbernoulli.b()`
	`posbernoulli.tb(drop.b = FALSE ~ 1, parallel.t = FALSE ~ 0)`
$\mathcal{M}_t/\mathcal{M}_{th}$	`posbernoulli.t()`
	`posbernoulli.tb(drop.b = FALSE ~ 0, parallel.t = FALSE ~ 1)`
$\mathcal{M}_{tb}/\mathcal{M}_{tbh}$	`posbernoulli.tb()`

Table 3: Upper table gives the $\boldsymbol{\eta}$ for the eight Otis *et al.* (1978) models, see Table 2 for definitions. Lower table gives the relationships between the eight models and function calls. The g = logit link is default for all models.

defined as z_{ij}, are stored on **VGAM** objects as the `cap.hist1` component in the `extra` slot. Also, there is a component called `cap1` which indicates on which sampling occasion the first capture occurred.

As will be illustrated in Section 6.3, a fitted CR object stores the point estimate for the HT population size estimator (3) in the `extra` slot with component name `N.hat`. Likewise, its standard error (4) has component name `SE.N.hat`. By default all the family functions return fitted values corresponding to the probabilities in the conditional likelihood function (2), however, Appendix B describes an alternative type of fitted value; the choice is made by the argument `type.fitted`, and the fitted values are returned by the `fitted()` methods function.

Notice that in Table 3, the **VGAM** family functions have arguments such as `parallel.b` which may be assigned a logical or else a formula with a logical as the response. If it is a single logical then the function may or may not apply that constraint to the intercept. The formula is the most general and some care must be taken with the intercept term. Here are some examples of the syntax:

- `parallel.b = TRUE ~ x2` means a parallelism assumption is applied to variables x_2 and the intercept, since formulas include the intercept by default.
- `parallel.b = TRUE ~ x2-1` means a parallelism assumption is applied to variable x_2 only.
- `parallel.b = FALSE ~ 0` means a parallelism assumption is applied to every variable including the intercept.

	`parallel.t`	`!parallel.t`
`parallel.b`	$\begin{pmatrix} \mathbf{0}_\tau & \mathbf{1}_\tau \\ \mathbf{1}_{\tau-1} & \mathbf{1}_{\tau-1} \end{pmatrix}, \begin{pmatrix} \mathbf{1}_\tau \\ \mathbf{1}_{\tau-1} \end{pmatrix}$	$\begin{pmatrix} \mathbf{0}_\tau & \mathbf{I}_\tau \\ \mathbf{1}_{\tau-1} & \mathbf{I}_{\tau[-1,]} \end{pmatrix}, \begin{pmatrix} \mathbf{I}_\tau \\ \mathbf{I}_{\tau[-1,]} \end{pmatrix}$
`!parallel.b`	$\begin{pmatrix} \mathbf{O}_{\tau\times(\tau-1)} & \mathbf{1}_\tau \\ \mathbf{I}_{\tau-1} & \mathbf{1}_{\tau-1} \end{pmatrix}, \begin{pmatrix} \mathbf{1}_\tau \\ \mathbf{1}_{\tau-1} \end{pmatrix}$	$\begin{pmatrix} \mathbf{O}_{\tau\times(\tau-1)} & \mathbf{I}_\tau \\ \mathbf{I}_{\tau-1} & \mathbf{I}_{\tau[-1,]} \end{pmatrix}, \begin{pmatrix} \mathbf{I}_\tau \\ \mathbf{I}_{\tau[-1,]} \end{pmatrix}$

Table 4: For the general $\mathcal{M}_{tbh}(\tau)$ family `posbernoulli.tb()`, the constraint matrices corresponding to the arguments `parallel.t`, `parallel.b` and `drop.b`. In each cell the LHS matrix is $\mathbf{H}_k$ when `drop.b` is `FALSE` for x_k. The RHS matrix is when `drop.b` is `TRUE` for x_k; it simply deletes the left submatrix of $\mathbf{H}_k$. These $\mathbf{H}_k$ should be seen in light of Table 3. Notes: (i) the default for `posbernoulli.tb()` is $\mathbf{H}_1$ = the LHS matrix of the top-right cell and $\mathbf{H}_k$ = the RHS matrix of the top-left cell; and (ii) $\mathbf{I}_{\tau[-1,]} = (\mathbf{0}_{\tau-1}|\mathbf{I}_{\tau-1})$.

5.2. Models $\mathcal{M}_0/\mathcal{M}_h$

For $\mathcal{M}_0/\mathcal{M}_h$, the defaults are given as:

```
posbinomial(link = "logit", mv = FALSE, parallel = FALSE,
  omit.constant = FALSE, p.small = 1e-04, no.warning = FALSE, zero = NULL)
```

Both models can alternatively be fitted using `posbernoulli.t()`, `posbernoulli.b()` and `posbernoulli.tb()` by setting the appropriate constrains/arguments (Table 3). For example, setting `posbernoulli.t(parallel.t = FALSE ~ 0)` constrains all the p_j to be equal.

If comparing all eight models using `AIC()` or `BIC()` then setting `omit.constant = TRUE` will allow for comparisons to be made with the positive-Bernoulli functions given below. The reason is that this omits the log-normalizing constant $\log \binom{\tau}{\tau y_i^*}$ from its conditional log-likelihood so that it is comparable with the logarithm of (2).

An extreme case for $\mathcal{M}_h$ is where $p_{ij} = p_i$ with p_i being parameters in their own right (Otis *et al.* 1978). While this could possibly be fitted by creating a covariate of the form `factor(1:n)` there would be far too many parameters for comfort. Such an extreme case is not recommended to avoid over-parameterization.

5.3. Models $\mathcal{M}_t/\mathcal{M}_{th}$

```
posbernoulli.t(link = "logit", parallel.t = FALSE ~ 1, iprob = NULL,
  p.small = 1e-04, no.warning = FALSE)
```

Note that for $\mathcal{M}_t$, capture probabilities are the same for each individual but may vary with time, i.e., $H_0 : p_{ij} = p_j$. One might wish to constrain the probabilities of a subset of sampling occasions to be equal by forming the appropriate constraint matrices.

Argument `iprob` is for an optional initial value for the probability, however all **VGAM** family functions are self-starting and usually do not need such input.

5.4. Models $\mathcal{M}_b/\mathcal{M}_{bh}$

```
posbernoulli.b(link = "logit", drop.b = FALSE ~ 1,
  type.fitted = c("likelihood.cond", "mean.uncond"), I2 = FALSE,
  ipcapture = NULL, iprecapture = NULL, p.small = 1e-04,
  no.warning = FALSE)
```

Setting `drop.b = FALSE ~ 0` assumes there is no behavioral effect and this reduces to $\mathcal{M}_0/\mathcal{M}_h$. The default constraint matrices are

$$\mathbf{H}_1 = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}, \quad \mathbf{H}_2 = \cdots = \mathbf{H}_d = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

so that the first coefficient $\beta^*_{(1)1}$ corresponds to the behavioral effect. Section 6.4 illustrates how the VGLM/VGAM framework can handle short-term and long-term behavioral effects.

5.5. Models $\mathcal{M}_{tb}/\mathcal{M}_{tbh}$

There are three arguments which determine whether there are behavioral effects and/or time effects: `parallel.b`, `parallel.t` and `drop.b`. The last two are as above. The defaults are:

```
posbernoulli.tb(link = "logit", parallel.t = FALSE ~ 1,
  parallel.b = FALSE ~ 0, drop.b = FALSE ~ 1,
  type.fitted = c("likelihood.cond", "mean.uncond"), imethod = 1,
  iprob = NULL, p.small = 1e-04, no.warning = FALSE,
  ridge.constant = 0.01, ridge.power = -4)
```

One would usually want to keep the behavioral effect to be equal over different sampling occasions, therefore `parallel.b` should be normally left to its default. Allowing it to be `FALSE` for a covariate x_k means an additional $\tau - 1$ parameters, something that is not warranted unless the data set is very large and/or the behavioral effect varies greatly over time.

Arguments `ridge.constant` and `ridge.power` concern the working weight matrices and are explained in further detail in Appendix A.

Finally, we note that using:

```
vglm(..., family = posbernoulli.tb(parallel.b = TRUE ~ 0,
  parallel.t = TRUE ~ 0, drop.b = TRUE ~ 0))
```

fits the most general model. Its formula is effectively (5) for $M = 2\tau - 1$, hence there are $(2\tau - 1)d$ regression coefficients in total—far too many for most data sets.

6. Examples

We present several examples using package **VGAM** on both real and simulated CR data.

6.1. Deer mice

Our first example uses a well-known data set analyzed in both Huggins (1991) and Amstrup *et al.* (2005). The CR data was collected on the deer mouse (*Peromyscus maniculatus*), a small rodent native to North America, which is about 8 to 10 cm long, not counting the length of the tail. There were $n = 38$ individual mice caught over $\tau = 6$ trapping occasions. Individual body weight, sex and age (young or adult) were also recorded, which were used as covariates to model heterogeneity. The data are given in the following data frame `deermice`:

```
R> head(deermice, 4)

  y1 y2 y3 y4 y5 y6 sex age weight
1  1  1  1  1  1  1   0   y     12
2  1  0  0  1  1  1   1   y     15
3  1  1  0  0  1  1   0   y     15
4  1  1  0  1  1  1   0   y     15
```

Each row represents the capture history followed by the corresponding covariate values for each observed individual. We compared our results with those given in Huggins (1991), who reported an analysis which involved fitting all eight model variations. Prior to this we relabeled the age and sex covariates to match those given in Huggins (1991):

```
R> deermice <- within(deermice, {
+    age <- 2 - as.numeric(age)
+    sex <- 1 - as.numeric(sex)
+  })
```

Below we demonstrate model fitting for each model in **VGAM**:

```
R> M.0 <- vglm(cbind(y1, y2, y3, y4, y5, y6) ~ 1,
+    posbernoulli.t(parallel = TRUE ~ 1), data = deermice)
R> M.b <- vglm(cbind(y1, y2, y3, y4, y5, y6) ~ 1,
+    posbernoulli.b, data = deermice)
R> M.t <- vglm(cbind(y1, y2, y3, y4, y5, y6) ~ 1,
+    posbernoulli.t, data = deermice)
R> M.h <- vglm(cbind(y1, y2, y3, y4, y5, y6) ~ weight + sex + age,
+    posbernoulli.t(parallel = TRUE ~ weight + sex + age), data = deermice)
R> M.th <- vglm(cbind(y1, y2, y3, y4, y5, y6) ~ weight + sex + age,
+    posbernoulli.t, data = deermice)
R> M.tb <- vglm(cbind(y1, y2, y3, y4, y5, y6) ~ 1,
+    posbernoulli.tb, data = deermice)
R> M.bh <- vglm(cbind(y1, y2, y3, y4, y5, y6) ~ weight + sex + age,
+    posbernoulli.b, data = deermice)
R> M.tbh <- vglm(cbind(y1, y2, y3, y4, y5, y6) ~ weight + sex + age,
+    posbernoulli.tb, data = deermice)
```

Notice that `parallel = TRUE` was used for models $\mathcal{M}_0/\mathcal{M}_h$. Population size estimates with standard errors (SEs), log-likelihood and AIC values, can all be easily obtained using the following code. For example, consider model $\mathcal{M}_{bh}$:

```
R> c(M.bh@extra$N.hat, M.bh@extra$SE.N.hat)

[1] 47.1442  7.3219

R> c(logLik(M.bh), AIC(M.bh))

[1] -139.54  289.09
```

We did this for each model, and obtained the following:

```
R> Table

         M.tbh   M.bh   M.tb   M.th    M.h    M.b    M.t    M.0
N.hat    47.14  47.14  46.48  39.66  39.85  42.26  38.40  38.47
SE        9.70   7.32  12.64   1.61   1.72   3.75   0.66   0.72
-2ln(L) 274.66 279.08 296.36 279.10 289.74 300.86 304.84 314.54
AIC     294.66 289.09 310.36 297.10 297.75 304.87 316.84 316.54
```

Based on the AIC, it was concluded that $\mathcal{M}_{bh}$ was superior (although other criteria can also be considered), yielding the following coefficients estimates (as well as their SEs):

```
R> round(coef(M.bh), 2)

(Intercept):1 (Intercept):2        weight           sex           age
         1.18         -2.91          0.16          0.92         -1.88

R> round(sqrt(diag(vcov(M.bh))), 2)

(Intercept):1 (Intercept):2        weight           sex           age
         0.41          0.90          0.06          0.35          0.63
```

which, along with the estimates for the population size, agree with the results of Huggins (1991). The first coefficient, 1.18, is positive and hence implies a possible trap-happy effect.

Now to illustrate the utility of fitting VGAMs, we performed some model checking on $\mathcal{M}_{bh}$ by confirming that the component function of `weight` is indeed linear. To do this, we smoothed this covariate but did not allow it to be too flexible due to the size of the data set:

```
R> fit.bh <- vgam(cbind(y1, y2, y3, y4, y5, y6) ~ s(weight, df = 3) +
+    sex + age, posbernoulli.b, data = deermice)
R> plot(fit.bh, se = TRUE, las = 1, lcol = "blue", scol = "orange",
+    rcol = "purple", scale = 5)
```

Notice that the `s()` function was used to smooth over the weight covariate with the equivalent degrees of freedom set to 3. Plots of the estimated component functions against each covariate are given in Figure 1. In general, `weight` does seem to have a (positive) linear effect on the logit scale. Young deer mice appear more easily caught compared to adults, and gender seems to have a smaller effect than weight. A more formal test of linearity is:

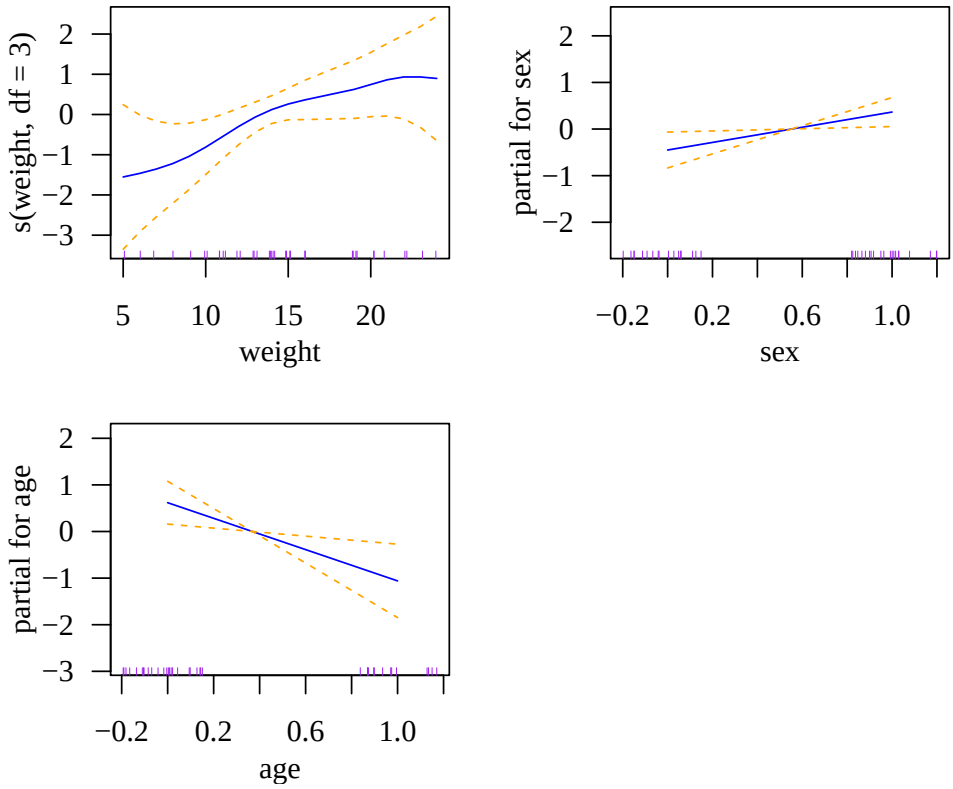

Figure 1: Estimated component functions with approximate ±2 pointwise SE bands fitting a $\mathcal{M}_{bh}$-VGAM using the `deermice` data (see Section 6.1). The rugplot gives jittered values of each covariate value x_{ik}.

```
R> summary(fit.bh)

Call:
vgam(formula = cbind(y1, y2, y3, y4, y5, y6) ~ s(weight, df = 3) +
    sex + age, family = posbernoulli.b, data = deermice)

Number of linear predictors:    2

Names of linear predictors: logit(pcapture), logit(precapture)

Dispersion Parameter for posbernoulli.b family:   1

Log-likelihood: -137.93 on 69.041 degrees of freedom

Number of iterations:  12

DF for Terms and Approximate Chi-squares for Nonparametric Effects

                  Df Npar Df Npar Chisq P(Chi)
(Intercept):1      1
```

```
(Intercept):2       1
s(weight, df = 3)   1       2       3.22 0.1943
sex                 1
age                 1
```

and not surprisingly, this suggests there is no significant nonlinearity. This is in agreement with Section 6.1 of Hwang and Huggins (2011) who used kernel smoothing.

Section 6.4 reports a further analysis of the `deermice` data using a behavioral effect comprising of long-term and short-term memory.

6.2. Yellow-bellied Prinia

Our second example also uses a well-known and well-studied data set collected on the Yellow-bellied Prinia (*Prinia flaviventris*), a common bird species located in Southeast Asia. A CR experiment was conducted at the Mai Po Nature Reserve in Hong Kong during 1991, where captured individuals had their wing lengths measured and fat index recorded. A total of $\tau = 19$ weekly capture occasions were considered, where $n = 151$ distinct birds were captured. In previous studies, models $\mathcal{M}_h$ and $\mathcal{M}_{th}$ have both been fitted to these data, where both wing length and fat index were used as covariates. We focus our attention on the former model, and considered the `posbinomial()` function, with some further emphasis on demonstrating smoothing on covariates. The `prinia` data consists of four columns and rows corresponding to each observed individual:

```
R> head(prinia, 4)[, 1:4]

     length fat cap noncap
1  1.006504   1   5     14
2  1.264626   1   3     16
3 -0.025983   1   6     13
4  3.071478   0   1     18
```

The first two columns give the observed covariate values for each individual, followed by the number of times each individual was captured/not captured respectively (columns 3–4). Notice that the wing length (`length`) was standardized here. We considered smoothing over the wing length, and now plotted the fitted capture probabilities with and without fat content against wing length present, see Figure 2.

```
R> M.h.GAM <- vgam(cbind(cap, noncap) ~ s(length, df = 3) + fat,
+    posbinomial(omit.constant = TRUE,
+    parallel = TRUE ~ s(length, df = 3) + fat), data = prinia)
R> M.h.GAM@extra$N.hat

[1] 447.63

R> M.h.GAM@extra$SE.N.hat

[1] 108.89
```

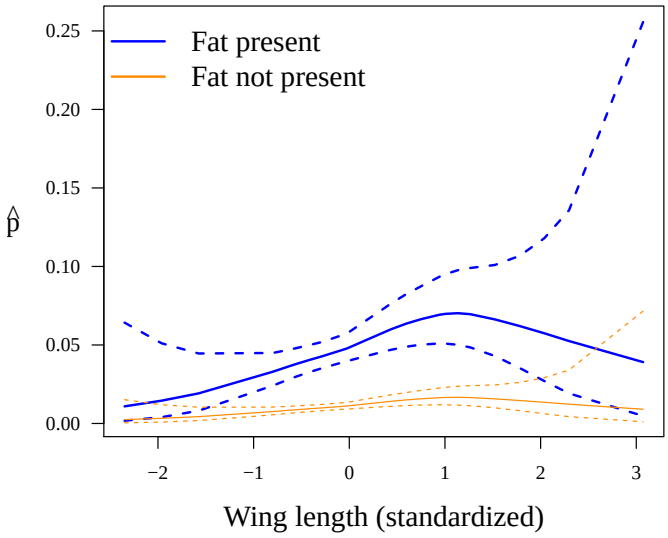

Figure 2: Capture probability estimates with approximate ±2 pointwise SEs, versus wing length with (blue) and without (orange) fat content present fitting a $\mathcal{M}_h$-VGAM using the `prinia` data (see Section 6.2). Notice that the SEs are wider at the boundaries.

Both the estimates for the population size and shape of the fitted capture probabilities with smoothing (Figure 2) matched those in previous studies, e.g., see Figure 1 of Hwang and Huggins (2007). Notice that capture probabilities are larger for individuals with fat content present, also the approximate ±2 pointwise SEs become wider at the boundaries – this feature is commonly seen in smooths.

6.3. A time-varying covariate example

To illustrate time-varying covariates in the $\mathcal{M}_{th}$ and $\mathcal{M}_{tbh}$ models via the `xij` argument, we mimicked the results of Huggins (1989) who fitted the $\mathcal{M}_{tbh}$ model to a small simulated data set of $n = 18$ observed individuals and $\tau = 10$ trapping occasions. To help illustrate the procedure we also fitted model $\mathcal{M}_{th}$. The true population was $N = 20$. For the ith individual, model $\mathcal{M}_{th}$ will be written as ($i = 1, \ldots, 18$, $j = 1, \ldots, 10$)

$$\text{logit}\, p_{ij} \quad = \quad \beta^*_{(1)1} + \beta^*_{(1)2} \cdot \texttt{x2}_i + \beta^*_{(1)3} \cdot \texttt{x3}_j, \tag{12}$$

and model $\mathcal{M}_{tbh}$ will be written as ($i = 1, \ldots, 18$, $j = 1, \ldots, 10$)

$$\text{logit}\, p_{ij} \quad = \quad \beta^*_{(1)1}\, z_{ij} + \beta^*_{(2)1} + \beta^*_{(1)2} \cdot \texttt{x2}_i + \beta^*_{(1)3} \cdot \texttt{x3}_j, \tag{13}$$

where $\beta^*_{(1)1}$ in (13) is the behavioral effect, and z_{ij} is defined in Table 1. Variable `x2` is an ordinary individual covariate such as weight, as in the previous examples. The variable `x3` is a time-varying or occasion-specific covariate such as temperature or daily rainfall that is handled using the `xij` argument described in Section 3.2. Note that the environmental covariates are involved in the η_j for individuals that have not been and have been previously captured so that if behavioral response is included in the model (e.g., $\mathcal{M}_{tbh}$) these must be repeated to construct the overall model matrix. Also, note that there can be no recaptures on the first occasion so that the environmental variable for this occasion need not be repeated. We first examined the data:

```
R> head(Huggins89table1, 4)

   x2 y01 y02 y03 y04 y05 y06 y07 y08 y09 y10 t01 t02 t03 t04 t05 t06
1 6.3   0   1   0   1   1   0   0   1   0   0 5.8 2.9 3.7 2.2 3.5 3.6
2 4.0   0   1   1   1   1   0   0   1   0   0 5.8 2.9 3.7 2.2 3.5 3.6
3 4.7   0   1   0   0   1   1   1   1   0   0 5.8 2.9 3.7 2.2 3.5 3.6
4 4.8   0   0   1   1   1   0   1   1   0   0 5.8 2.9 3.7 2.2 3.5 3.6
  t07 t08 t09 t10
1 1.9   3 4.8 4.7
2 1.9   3 4.8 4.7
3 1.9   3 4.8 4.7
4 1.9   3 4.8 4.7
```

The time-varying/occasion-specific covariate variable `x3` is represented by variables `t01`–`t10`. As noted above, we need to construct `T02`–`T10` to model the recapture probabilities through η_j for $j = 11, \ldots, 19$:

```
R> Hdata <- transform(Huggins89table1, x3.tij = t01, T02 = t02, T03 = t03,
+    T04 = t04, T05 = t05, T06 = t06, T07 = t07, T08 = t08, T09 = t09,
+    T10 = t10)
R> Hdata <- subset(Hdata, y01 + y02 + y03 + y04 + y05 + y06 + y07 + y08 +
+    y09 + y10 > 0)
```

The last step deletes the two observations which were never caught, such that $n = 18$. Thus model (12) can be fitted by:

```
R> fit.th <- vglm(cbind(y01, y02, y03, y04, y05, y06, y07, y08, y09, y10) ~
+    x2 + x3.tij,
+    xij = list(x3.tij ~ t01 + t02 + t03 + t04 + t05 + t06 + t07 + t08 +
+    t09 + t10 - 1),
+    posbernoulli.t(parallel.t = TRUE ~ x2 + x3.tij), data = Hdata,
+    trace = FALSE, form2 = ~ x2 + x3.tij + t01 + t02 + t03 + t04 + t05 +
+    t06 + t07 + t08 + t09 + t10)
```

The `form2` argument is required if `xij` is used and it needs to include all the variables in the model. It is from this formula that a very large model matrix is constructed, from which the relevant columns are extracted to construct the diagonal matrix in (9) in the specified order of diagonal elements given by `xij`. Their names need to be uniquely specified. To check the constraint matrices we can use:

```
R> constraints(fit.th, matrix = TRUE)

     (Intercept) x2 x3.tij
[1,]           1  1      1
[2,]           1  1      1
[3,]           1  1      1
[4,]           1  1      1
```

```
 [5,]           1  1      1
 [6,]           1  1      1
 [7,]           1  1      1
 [8,]           1  1      1
 [9,]           1  1      1
[10,]           1  1      1
```

Model (13) can be fitted by:

```
R> fit.tbh <- vglm(cbind(y01, y02, y03, y04, y05, y06, y07, y08, y09, y10) ~
+    x2 + x3.tij,
+    xij = list(x3.tij ~ t01 + t02 + t03 + t04 + t05 + t06 + t07 + t08 +
+    t09 + t10 + T02 + T03 + T04 + T05 + T06 + T07 + T08 + T09 + T10 - 1),
+    posbernoulli.tb(parallel.t = TRUE ~ x2 + x3.tij), data = Hdata,
+    trace = FALSE, form2 = ~ x2 + x3.tij + t01 + t02 + t03 + t04 + t05 +
+    t06 + t07 + t08 + t09 + t10 + T02 + T03 + T04 + T05 + T06 + T07 +
+    T08 + T09 + T10)
```

To compare with model (12) we have:

```
R> c(logLik(fit.th), AIC(fit.th))

[1] -97.08 200.16

R> c(logLik(fit.tbh), AIC(fit.tbh))

[1] -94.15 196.30
```

so that the behavioral response model does indeed give a better fit. To check, the constraint matrices are (cf., Table 4):

```
R> head(constraints(fit.tbh, matrix = TRUE), 4)

     (Intercept):1 (Intercept):2 x2 x3.tij
[1,]             0             1  1      1
[2,]             0             1  1      1
[3,]             0             1  1      1
[4,]             0             1  1      1

R> tail(constraints(fit.tbh, matrix = TRUE), 4)

      (Intercept):1 (Intercept):2 x2 x3.tij
[16,]             1             1  1      1
[17,]             1             1  1      1
[18,]             1             1  1      1
[19,]             1             1  1      1
```

The coefficients $\widehat{\beta}^{*}_{(j)k}$ and their SEs are:

```
R> coef(fit.tbh)

(Intercept):1 (Intercept):2            x2        x3.tij
      1.09376      -0.63056       0.38449      -0.83692

R> sqrt(diag(vcov(fit.tbh)))

(Intercept):1 (Intercept):2            x2        x3.tij
      0.53648       1.36080       0.22291       0.18567
```

The first coefficient, 1.09, is the behavioral effect estimate. Since this is positive and the Wald statistic is 2.04, this suggests that there may be a trap-happy effect to behavioral response.

Estimates of the population size can be obtained from:

```
R> fit.tbh@extra$N.hat

[1] 20.935

R> fit.tbh@extra$SE.N.hat

[1] 3.4559
```

This compares with $\widehat{N} = 20.86$ with a SE of 4.51 (Huggins 1989).

In closing, we refit model `fit.tbh` using `Select()` to illustrate the avoidance of manual specification of cumbersome formulas and response matrices with many columns. For example, suppose `pdata` is a data frame with columns `y01`, `y02`, ..., `y30`. Then `Select(pdata, "y")` will return the matrix `cbind(y01, y02, ..., y30)` if there are no other variables beginning with `"y"`.

Starting with `Huggins89table1`, the following code works quite generally provided the original variables are labeled as `y01`, `y02`, ..., and `t01`, `t02`, ... The code makes a copy of `cbind(t01, ..., t10)` for the capture probabilities and calls the variables `cbind(T01, ..., T10)` for the recapture probabilities. Also, `Form2` contains more variables than what is needed:

```
R> Hdata <- subset(Huggins89table1,
+    rowSums(Select(Huggins89table1, "y")) > 0)
R> Hdata.T <- Select(Hdata, "t")
R> colnames(Hdata.T) <- gsub("t", "T", colnames(Hdata.T))
R> Hdata <- data.frame(Hdata, Hdata.T)
R> Hdata <- transform(Hdata, x3.tij = y01)
R> Form2 <- Select(Hdata, prefix = TRUE, as.formula = TRUE)
R> Xij <- Select(Hdata, c("t", "T"), as.formula = TRUE, sort = FALSE,
+    rhs = "0", lhs = "x3.tij", exclude = "T01")
R> fit.tbh <- vglm(Select(Hdata, "y") ~ x2 + x3.tij, form2 = Form2,
+    xij = list(Xij), posbernoulli.tb(parallel.t = TRUE ~ x2 + x3.tij),
+    data = Hdata, trace = FALSE)
R> coef(fit.tbh)
```

```
(Intercept):1 (Intercept):2            x2        x3.tij
      1.09376      -0.63056       0.38449      -0.83692
```

Note that this illustrates the ability to enter a matrix response without an explicit `cbind()`, e.g., `Y <- Select(Hdata, "y")` and the invocation `vglm(Y ~ ...)` would work as well. However, the utility of `cbind()` encourages the use of column names, which is good style and avoids potential coding errors.

6.4. Ephemeral and enduring memory

Yang and Chao (2005) consider modeling the behavioral effect with both enduring (long-term) and ephemeral (short-term) memory components. For example, the short-term component depends on whether or not the animal was caught on the most recent sampling occasion. We call this a lag-1 effect. In the example of this section, which combines aspects of Sections 6.1 and 6.3, we illustrate how this may be easily achieved within the VGLM framework; it is another case of using the `xij` argument. We retain the enduring component as with the $\mathcal{M}_{tbh}$: $\mathbf{H}_1$ contains a column that applies to all the recapture probabilities. For simplicity, we first consider a lag-1 effect only (as in Yang and Chao 2005) for the short-term component.

In the following, we fit a $\mathcal{M}_{tbh}$ model to `deermice` with both long-term and short-term effects

$$\begin{aligned} \text{logit}\, p_{cs} &= \beta^*_{(2)1} + \beta^*_{(1)2}\, \texttt{sex} + \beta^*_{(1)3}\, \texttt{weight}, \\ \text{logit}\, p_{rt} &= \beta^*_{(1)1} + \beta^*_{(2)1} + \beta^*_{(1)2}\, \texttt{sex} + \beta^*_{(1)3}\, \texttt{weight} + \beta^*_{(1)4}\, y_{t-1}, \end{aligned}$$

where $s = 2, \ldots, \tau$, $t = 1, \ldots, \tau$ and $\tau = 6$.

```
R> deermice <- transform(deermice, Lag1 = y1)
R> M.tbh.lag1 <- vglm(cbind(y1, y2, y3, y4, y5, y6) ~ sex + weight + Lag1,
+    posbernoulli.tb(parallel.t = FALSE ~ 0, parallel.b = FALSE ~ 0,
+    drop.b = FALSE ~ 1),
+    xij = list(Lag1 ~ fill(y1) + fill(y2) + fill(y3) + fill(y4) +
+    fill(y5) + fill(y6) + y1 + y2 + y3 + y4 + y5),
+    form2 = ~ sex + weight + Lag1 + fill(y1) + fill(y2) + fill(y3) +
+    fill(y4) + fill(y5) + fill(y6) + y1 + y2 + y3 + y4 + y5 + y6,
+    data = deermice)
R> coef(M.tbh.lag1)
```

```
(Intercept):1 (Intercept):2          sex       weight         Lag1
    1.2776930    -1.5782178    1.1513794    0.0058372    0.0230575
```

The coefficient of `Lag1`, 0.0231, is the estimated ephemeral effect $\widehat{\beta}^*_{(1)4}$. The estimated enduring effect $\widehat{\beta}^*_{(1)1}$ has value 1.2777. Note that the `fill()` function is used to create 6 variables having 0 values, i.e., $\mathbf{0}_n$.

There is an alternative method to fit the above model; here we set $\mathbf{H}_{\texttt{Lag1}} = (\mathbf{0}_\tau^\top, \mathbf{1}_{\tau-1}^\top)^\top$ and the variables `fill(y1)`, ..., `fill(y6)` can be replaced by variables that do not need to be 0. Importantly, the two methods have $\mathbf{X}^{\#}_{(ik)}\mathbf{H}_k$ in (9) being the same. The second alternative method requires constraint matrices to be inputted using the `constraints` argument, e.g.,

```
R> deermice <- transform(deermice, Lag1 = y1)
R> deermice <- transform(deermice, f1 = y1, f2 = y1, f3 = y1, f4 = y1,
+    f5 = y1, f6 = y1)
R> tau <- 6
R> H2 <- H3 <- cbind(rep(1, 2 * tau - 1))
R> H4 <- cbind(c(rep(0, tau), rep(1, tau - 1)))
R> M.tbh.lag1.method2 <- vglm(cbind(y1, y2, y3, y4, y5, y6) ~ sex + weight +
+    Lag1, posbernoulli.tb(parallel.b = TRUE ~ 0, parallel.t = TRUE ~ 0),
+    constraints = list("(Intercept)" = cbind(H4, 1), sex = H2, weight = H3,
+    Lag1 = H4), xij = list(Lag1 ~ f1 + f2 + f3 + f4 + f5 + f6 + y1 + y2 +
+    y3 + y4 + y5), form2 = Select(deermice, prefix = TRUE,
+    as.formula = TRUE), data = deermice)
R> coef(M.tbh.lag1.method2)

(Intercept):1 (Intercept):2           sex        weight          Lag1
    1.2776930    -1.5782178     1.1513794     0.0058372     0.0230575
```

is identical. In closing, it can be noted that more complicated models can be handled. For example, the use of `pmax()` to handle lag -2 effects as follows:

```
R> deermice <- transform(deermice, Lag2 = y1)
R> M.bh.lag2 <- vglm(cbind(y1, y2, y3, y4, y5, y6) ~ sex + weight + Lag2,
+    posbernoulli.tb(parallel.t = FALSE ~ 0, parallel.b = FALSE ~ 0,
+    drop.b = FALSE ~ 1),
+    xij = list(Lag2 ~ fill(y1) + fill(y2) + fill(y3) + fill(y4) +
+    fill(y5) + fill(y6) + y1 + pmax(y1, y2) + pmax(y2, y3) +
+    pmax(y3, y4) + pmax(y4, y5)),
+    form2 = ~ sex + weight + Lag2 + fill(y1) + fill(y2) + fill(y3) +
+    fill(y4) + fill(y5) + fill(y6) + y1 + pmax(y1, y2) + pmax(y2, y3) +
+    pmax(y3, y4) + pmax(y4, y5) + y6, data = deermice)
R> coef(M.bh.lag2)

(Intercept):1 (Intercept):2           sex        weight          Lag2
    1.9676972    -1.6589767     1.2172470     0.0074673    -0.7222657
```

Models with separate lag-1 and lag-2 effects may also be similarly estimated as above.

6.5. Some timing and reliability tests

We investigated computational times for the **VGAM** CR functions and compared these with the **RMark** and **mra** packages. Once again we used the `prinia` data as above, but now considered all the positive-Bernoulli models. These data have a large number of capture occasions which allowed us to detect significant computational times. Models $\mathcal{M}_{bh}$ and $\mathcal{M}_{tbh}$ were excluded from **mra** as we encountered several convergence issues. For all other heterogeneity models, we considered the fat content covariate only. The **Rcapture** package was also initially considered, however computational times for individual models were not obtainable here, since several different models are collectively fitted in the `closedp()` function and we encountered

convergence issues when fitting model $\mathcal{M}_{tb}$. Also, note that **RMark** requires external software which is not available for Linux. Below we give the computational times for each model on a Windows machine with quad processor (860 @ 2.80 GHz, 2.93 GHz CPU) and 4GB memory and running R 3.2.0:

```
R> table_time
```

	VGAM comp. times	mra comp. times	RMark comp. times
M_0	0.260	0.021	0.307
M_t	0.230	1.700	0.381
M_h	0.280	0.062	0.289
M_b	0.090	0.130	0.303
M_tb	1.600	2.800	0.403
M_th	0.300	1.700	0.517
M_bh	0.034	NA	0.319
M_tbh	1.300	NA	0.669
Total	4.000	6.500	3.188

Overall, **RMark** gave the fastest computational times followed by **VGAM**, however for some non-time-dependent models, the **mra** package was quickest. It should also be noted that these computational times were demonstrated on only one example data set, so the timing may vary for different data sets. Also, **mra** crashed on one of the authors' Linux machines when attempting to compute models $\mathcal{M}_b$ and $\mathcal{M}_{tb}$.

To further examine **VGAM**'s smoothing capabilities and SE performance, we undertook a simulation study following a similar set-up as in Section 5.2 of Stoklosa and Huggins (2012). Here, we considered model $\mathcal{M}_{th}$ using three (individual) covariates: two continuous covariates ($x_{i1} \sim \text{Unif}(-3, 0)$ and $x_{i2} \sim \text{Unif}(-3, 0)$) and a binary covariate ($x_{i3} \sim \text{Binom}(1, 0.6)$) along with an intercept set to -1.2. We then set $f(x_{i1}) = 2\sin(2x_{i1} - 1)$ and $f(x_{i2}) = \cos(2x_{i2})$ to represent the nonlinear relationship functions. The true population was set to $N = 100$ with $\tau = 8$ capture occasions. A VGLM which falsely assumes a linear relationship between the response and covariates was also included to demonstrate the improvements when using smoothing. For each fitted model, we investigated: (1) the mean, median, standard deviation (SD) and median absolute deviation (MAD) for n and the population size estimate; (2) the mean and median of the SE estimates for $\widehat{N}$, and (3) the mean square error (MSE), the median absolute error (MAE) and the coverage of the nominal 95% confidence interval for N. We present the results in Table 5 and compared these with Stoklosa and Huggins (2012)'s code for 500 simulations.

From Table 5 it is evident there is some bias present in the populations size estimates when assuming a linear model, whereas **VGAM** performs well and compares with the results of Stoklosa and Huggins (2012). Both the SEs and coverage probabilities were somewhat reasonable, with the MSE and MAE being similar to those of Stoklosa and Huggins (2012).

7. Discussion

We have presented how the VGLM/VGAM framework naturally handles the conditional-likelihood and closed population CR models in a GLM-like manner. Recently, Stoklosa,

	n	Linear $\widehat{N}$	GAM $\widehat{N}$	**VGAM** $\widehat{N}$
Mean	77.07	78.72	102.91	102.33
Median	77.00	78.74	98.84	97.17
SD	4.24	4.38	18.04	18.45
MAD	4.44	4.22	13.08	12.33
Mean SE	–	1.41	14.71	13.15
Median SE	–	1.34	11.30	10.12
MSE	–	471.92	333.36	345.26
MAE	–	21.25	9.57	9.57
Coverage	–	0.00	0.85	0.84

Table 5: We compared a linear VGLM and two GAMs: Stoklosa and Huggins (2012)'s code and **VGAM**. For each model we give: (1) the mean, median, standard deviation (SD) and median absolute deviation (MAD) for n and each population size estimate; (2) the mean and median of the SE estimates for $\widehat{N}$; and (3) the mean square error (MSE), the median absolute error (MAE) and the coverage of the nominal 95% confidence interval for N.

Hwang, Wu, and Huggins (2011) proposed a partial likelihood approach for heterogeneous models with covariates. There, the recaptures of the observed individuals were modeled, which yielded a binomial distribution, and hence a GLM/GAM framework in R is also possible. However, some efficiency is lost, as any individuals captured only once on the last occasion are excluded. The advantage of partial likelihood is that the full range of GLM based techniques, which includes more than GAMs, are readily applicable. Huggins and Hwang (2007); Hwang and Huggins (2011) and Stoklosa and Huggins (2012) implemented smoothing on covariates for more general models, however these methods required implementing sophisticated coding for estimating the model parameters. Zwane and van der Heijden (2004) also used the **VGAM** package for smoothing and CR data, but considered multinomial logit models as an alternative to the conditional likelihood. We believe the methods presented here, based on spline smoothing and classical GAM, are an improvement in terms of ease of use, capability and efficiency.

When using any statistical software, the user must take a careful approach when analyzing and interpreting their output data. In our case, one must be careful when estimating the population via the HT estimator. Notice that (3) is a sum of the reciprocal of the estimated capture probabilities seen at least once $\widehat{\pi}_i(\boldsymbol{\theta})$. Hence, for very small $\widehat{\pi}_i(\boldsymbol{\theta})$, the population size estimate may give a large and unrealistic value (this is also apparent when using the **mra** package and the **Rcapture** package which gives the warning message: "`The abundance estimation for this model can be unstable`"). To avoid this, Stoklosa and Huggins (2012) proposed a robust HT estimator which places a lower bound on $\widehat{\pi}_i(\boldsymbol{\theta})$ to prevent it from giving unrealistically large values. In package **VGAM**, a warning similar to the one in package **Rcapture** is also implemented, and there are arguments to control how close to 0 "very small" is and to suppress the warning entirely.

There are limitations for $\mathcal{M}_h$-type models, in that they rely on the very strong assumption that all the heterogeneity is explained by the unit-level covariates. This assumption is often not true, see, e.g., Rivest and Baillargeon (2014). To this end, a proposal is to add random-effects to the VGLM class. This would result in the VGLMM class ("M" for mixed) which would be

potentially very useful if developed successfully. Of course, VGLMMs would contain GLMMs (McCulloch, Searle, and Neuhaus 2008) as a special case. Further future implementations also include: automatic smoothing parameter selection (via, say, generalized cross validation or AIC); including a bootstrap procedure as an alternative for SEs.

GAMs are now a standard statistical tool in the modern data analyst's toolbox. With the exception of the above references, CR analysis has since been largely confined to a few regression coefficients (at most), and devoid of any data-driven exploratory analyses involving graphics. This work has sought to rectify this need by introducing GAM-like analyses using a unified statistical framework. Furthermore, the functions are easy to use and often can be invoked by a single line of code. Finally, we believe this work is a substantial improvement over other existing software for closed population estimation, and we have shown **VGAM**'s favorable speed and reliability over other closed population CR R packages.

Acknowledgments

We thank the reviewers for their helpful feedback that led to substantial improvements in the manuscript. TWY thanks Anne Chao for a helpful conversation, and the Department of Mathematics and Statistics at the University of Melbourne for hospitality, during a sabbatical visit to Taiwan and a workshop, respectively. Part of his work was also conducted while as a visitor to the Institute of Statistical Science, Academia Sinica, Taipei, during October 2012. JS visited TWY on the Tweedle-funded Melbourne Abroad Travelling Scholarship, the University of Melbourne, during September 2011. All authors would also like to thank Paul Yip for providing and giving permission for use of the `prinia` data set, and Zachary Kurtz for some helpful comments.

References

Alho JM (1990). "Logistic Regression in Capture-Recapture Models." *Biometrics*, **54**(3), 623–635.

Amstrup SC, McDonald TL, Manly BFJ (2005). *Handbook of Capture-Recapture Analysis*. Princeton University Press, Princeton.

Baillargeon S, Rivest LP (2007). "**Rcapture**: Loglinear Models for Capture-Recapture in R." *Journal of Statistical Software*, **19**(5), 1–31. URL http://www.jstatsoft.org/v19/i05/.

Baillargeon S, Rivest LP (2014). ***Rcapture**: Loglinear Models for Capture-Recapture Experiments*. R package version 1.4-2, URL http://CRAN.R-project.org/package=Rcapture.

Bolker BM (2008). *Ecological Models and Data in R*. Princeton University Press, Princeton.

Bunge JA (2013). "A Survey of Software for Fitting Capture-Recapture Models." *Wiley Interdisciplinary Reviews: Computational Statistics*, **5**(2), 114–120.

Burnham KP, Anderson DR (1999). *Model Selection and Inference: A Practical Information-Theoretic Approach*. Springer-Verlag, New York.

Cooch EG, White GC (2012). *Program **MARK**: A Gentle Introduction, 11th Edition*. URL http://www.phidot.org/software/mark/docs/book/.

Cormack RM (1989). "Log-Linear Models for Capture-Recapture." *Biometrics*, **45**(2), 395–413.

Crawley MJ (1993). *GLIM for Ecologists*. Blackwell Scientific Publications, Boston.

Gimenez O, Covas R, Brown CR, Anderson MD, Brown MB, Lenormand T (2006). "Nonparametric Estimation of Natural Selection on a Quantitative Trait Using Mark-Recapture Data." *Evolution*, **60**(3), 460–466.

Green PJ, Silverman BW (1994). *Nonparametric Regression and Generalized Linear Models: A Roughness Penalty Approach*. Chapman & Hall/CRC, London.

Hastie T, Tibshirani R (1990). *Generalized Additive Models*. Chapman & Hall/CRC, London.

Horvitz DG, Thompson DJ (1952). "A Generalization of Sampling Without Replacement from a Finite Universe." *Journal of the American Statistical Association*, **47**(260), 663–685.

Huggins RM (1989). "On the Statistical Analysis of Capture Experiments." *Biometrika*, **76**(1), 133–140.

Huggins RM (1991). "Some Practical Aspects of a Conditional Likelihood Approach to Capture Experiments." *Biometrics*, **47**(2), 725–732.

Huggins RM, Hwang WH (2007). "Non-Parametric Estimation of Population Size from Capture-Recapture Data when the Capture Probability Depends on a Covariate." *Journal of the Royal Statistical Society C*, **56**(4), 429–443.

Huggins RM, Hwang WH (2011). "A Review of the Use of Conditional Likelihood in Capture-Recapture Experiments." *International Statistical Review*, **79**(3), 385–400.

Hwang WH, Chao A (2003). *Brief User Guide for Program **CARE-3**: Analyzing Continuous-Time Capture-Recapture Data*. URL http://chao.stat.nthu.edu.tw/blog/software-download/care/.

Hwang WH, Huggins RM (2007). "Application of Semiparametric Regression Models in the Analysis of Capture-Recapture Experiments." *Australian & New Zealand Journal of Statistics*, **49**(2), 191–202.

Hwang WH, Huggins RM (2011). "A Semiparametric Model for a Functional Behavioural Response to Capture in Capture-Recapture Experiments." *Australian & New Zealand Journal of Statistics*, **53**(4), 403–421.

Laake JL (2013). "**RMark**: An R Interface for Analysis of Capture-Recapture Data with **MARK**." *AFSC Processed Report 2013-01*, Alaska Fisheries Science Center, NOAA, National Marine Fisheries Service, Seattle. URL http://www.afsc.noaa.gov/Publications/ProcRpt/PR2013-01.pdf.

Lebreton JD, Burnham KP, Clobert J, Anderson DR (1992). "Modelling Survival and Testing Biological Hypothesis Using Marked Animals: Unified Approach with Case Studies." *Biometrics*, **62**(1), 67–118.

McCrea RS, Morgan BJT (2014). *Analysis of Capture-Recapture Data*. Chapman & Hall/CRC, London.

McCullagh P, Nelder JA (1989). *Generalized Linear Models*. 2nd edition. Chapman & Hall/CRC, London.

McCulloch CE, Searle SR, Neuhaus JM (2008). *Generalized, Linear, and Mixed Models*. 2nd edition. John Wiley & Sons, Hoboken.

McDonald TL (2012). ***mra**: Analysis of Mark-Recapture Data*. R package version 2.13, URL http://CRAN.R-project.org/package=mra.

Nelder JA, Wedderburn RWM (1972). "Generalized Linear Models." *Journal of the Royal Statistical Society A*, **135**(3), 370–384.

Otis DL, Burnham KP, White GC, Anderson DR (1978). "Statistical Inference From Capture Data on Closed Animal Populations." *Wildlife Monographs*, **62**, 3–135.

Patil GP (1962). "Maximum Likelihood Estimation for Generalized Power Series Distributions and its Application to a Truncated Binomial Distribution." *Biometrika*, **49**(1), 227–237.

Pollock KH (1991). "Modeling Capture, Recapture, and Removal Statistics for Estimation of Demographic Parameters for Fish and Wildlife Populations: Past, Present and Future." *Journal of the American Statistical Association*, **86**(413), 225–238.

Pollock KH (2002). "The Use of Auxiliary Variables in Capture-Recapture Modelling: An Overview." *Journal of Applied Statistics*, **29**(1–4), 85–102.

Pollock KH, Hines J, Nichols J (1984). "The Use of Auxiliary Variables in Capture-Recapture and Removal Experiments." *Biometrics*, **40**(2), 329–340.

R Core Team (2015). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org.

Rivest LP, Baillargeon S (2014). "Capture-Recapture Methods for Estimating the Size of a Population: Dealing with Variable Capture Probabilities." In *Statistics in Action: A Canadian Outlook*, pp. 289–304.

Schluter D (1988). "Estimating the Form of Natural Selection on a Quantitative Trait." *Evolution*, **42**(5), 849–861.

Stoklosa J, Huggins RM (2012). "A Robust P-Spline Approach to Closed Population Capture-Recapture Models with Time Dependence and Heterogeneity." *Computational Statistics & Data Analysis*, **56**(2), 408–417.

Stoklosa J, Hwang WH, Wu SH, Huggins RM (2011). "Heterogeneous Capture-Recapture Models with Covariates: A Partial Likelihood Approach for Closed Populations." *Biometrics*, **67**(4), 1659–1665.

Wood SN (2006). *Generalized Additive Models: An Introduction with R*. Chapman & Hall/CRC, London.

Yang HC, Chao A (2005). "Modeling Animals' Behavioral Response by Markov Chain Models for Capture-Recapture Experiments." *Biometrics*, **61**(4), 1010–1017.

Yee TW (2008). "The **VGAM** Package." *R News*, **8**(2), 28–39.

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/.

Yee TW (2014). ***VGAM**: Vector Generalized Linear and Additive Models*. R package version 0.9-6, URL http://CRAN.R-project.org/package=VGAM.

Yee TW (2015). *Vector Generalized Linear and Additive Models: With an Implementation in R*. Springer-Verlag, New York.

Yee TW, Hastie TJ (2003). "Reduced-Rank Vector Generalized Linear Models." *Statistical Modelling*, **3**(1), 15–41.

Yee TW, Mitchell ND (1991). "Generalized Additive Models in Plant Ecology." *Journal of Vegetation Science*, **2**(5), 587–602.

Yee TW, Wild CJ (1994). "Vector Generalized Additive Models." *Technical Report STAT04*, Department of Statistics, University of Auckland, Auckland, New Zealand.

Yee TW, Wild CJ (1996). "Vector Generalized Additive Models." *Journal of the Royal Statistical Society B*, **58**(3), 481–493.

Zwane EN, van der Heijden PGM (2004). "Semiparametric Models for Capture-Recapture Studies with Covariates." *Computational Statistics & Data Analysis*, **47**(4), 729–743.

A. Derivatives

We give the first and (expected) second derivatives of the models. Let $z_{ij} = 1$ if individual i has been captured before occasion j, else $= 0$. Also, let p_{cj} and p_{rj} be the probability that an individual is captured/recaptured at sampling occasion j, and $Q_{s:t} = \prod_{j=s}^{t}(1 - p_{cj})$. Occasionally, subscripts i are omitted for clarity. Huggins (1989) gives a general form for the derivatives of the conditional likelihood (2).

For the $\mathcal{M}_{tbh}$, the score vector is

$$\begin{aligned}
\frac{\partial \ell_i}{\partial p_{cj}} &= (1 - z_{ij}) \left[\frac{y_{ij}}{p_{cj}} - \frac{1 - y_{ij}}{1 - p_{cj}} \right] - \frac{Q_{1:\tau}/(1 - p_{cj})}{1 - Q_{1:\tau}}, \quad j = 1, \ldots, \tau, \\
\frac{\partial \ell_i}{\partial p_{rj}} &= z_{ij} \left[\frac{y_{ij}}{p_{rj}} - \frac{1 - y_{ij}}{1 - p_{rj}} \right], \quad j = 2, \ldots, \tau,
\end{aligned}$$

and the non-zero elements of the expected information matrix (EIM) can be written

$$\begin{aligned}
-\mathsf{E}\left(\frac{\partial^2 \ell}{\partial p_{cj}^2} \right) &= \frac{Q_{1:(j-1)}}{1 - Q_{1:\tau}} \left\{ \frac{1}{p_{cj}} + \frac{1 - Q_{(j+1):\tau}}{1 - p_{cj}} \right\} - \left(\frac{\partial Q_{1:\tau}/\partial p_{cj}}{1 - Q_{1:\tau}} \right)^2 \\
&= \frac{1}{(1 - p_{cj})^2 (1 - Q_{1:\tau})} \left\{ \frac{Q_{1:j}}{p_{cj}} - \frac{Q_{1:\tau}}{1 - Q_{1:\tau}} \right\}, \\
-\mathsf{E}\left(\frac{\partial^2 \ell}{\partial p_{rj}^2} \right) &= \frac{1 - Q_{1:j}/(1 - p_{cj})}{p_{rj}(1 - p_{rj})(1 - Q_{1:\tau})}, \\
-\mathsf{E}\left(\frac{\partial^2 \ell}{\partial p_{cj}\, \partial p_{ck}} \right) &= \frac{-\dfrac{\partial Q_{1:\tau}}{\partial p_{cj}} \dfrac{\partial Q_{1:\tau}}{\partial p_{ck}}}{(1 - Q_{1:\tau})^2} - \frac{\dfrac{\partial^2 Q_{1:\tau}}{\partial p_{cj}\, \partial p_{ck}}}{(1 - Q_{1:\tau})}, \quad j \neq k,
\end{aligned}$$

where $\partial Q_{1:\tau}/\partial p_{cj} = -Q_{1:\tau}/(1 - p_{cj})$ and $\partial^2 Q_{1:\tau}/(\partial p_{cj}\, \partial p_{ck}) = Q_{1:\tau}/\{(1 - p_{cj})(1 - p_{ck})\}$.

Arguments `ridge.constant` and `ridge.power` in `posbernoulli.tb()` add a ridge parameter to the first τ EIM diagonal elements, i.e., those for p_{cj}. This ensures that the working weight matrices are positive-definite, and is needed particularly in the first few iteratively reweighted least squares iterations. Specifically, at iteration a a positive value $\omega K \times a^p$ is added, where K and p correspond to the two arguments, and ω is the mean of elements of such working weight matrices. The ridge factor decays to zero as iterations proceed and plays a negligible role upon convergence.

For individual i, let y_{0i} be the number of noncaptures before the first capture, y_{r0i} be the number of noncaptures after the first capture, and y_{r1i} be the number of recaptures after the first capture. For the $\mathcal{M}_{bh}$, the score vector is

$$\begin{aligned}
\frac{\partial \ell_i}{\partial p_c} &= \frac{1}{p_c} - \frac{y_{0i}}{1 - p_c} - \frac{\tau (1 - p_{ij})^{\tau - 1}}{1 - Q_{1:\tau}}, \\
\frac{\partial \ell_i}{\partial p_r} &= \frac{y_{r1i}}{p_r} - \frac{y_{r0i}}{1 - p_c}.
\end{aligned}$$

The non-zero elements of the EIM can be written

$$\begin{aligned}
-\mathsf{E}\left(\frac{\partial^2 \ell}{\partial p_c^2}\right) &= \frac{p_c}{1-Q_{1:\tau}} \sum_{j=1}^{\tau} (1-p_c)^{j-1} \left(\frac{j-1}{(1-p_c)^2} + \frac{1}{p_c^2}\right) - \frac{\partial}{\partial p_c}\left(\frac{\partial Q_{1:\tau}/\partial p_c}{1-Q_{1:\tau}}\right) \\
&= \frac{1-Q_{1:\tau} - p_c[1+(\tau-1)Q_{1:\tau}]}{p_c\,(1-p_c)^2\,(1-Q_{1:\tau})} + \frac{1}{p_c^2} - \\
&\quad \tau(\tau-1)\,\frac{(1-p_c)^{\tau-2}}{1-Q_{1:\tau}} + \tau^2\,\frac{(1-p_c)^{\tau-2}}{(1-Q_{1:\tau})^2}, \\
-\mathsf{E}\left(\frac{\partial^2 \ell}{\partial p_r^2}\right) &= \frac{1}{p_r\,(1-p_r)\,(1-Q_{1:\tau})} \sum_{j=1}^{\tau} \left\{1-(1-p_c)^{j-1}\right\} \\
&= \frac{\tau - (1-Q_{1:\tau})/p_c}{p_r(1-p_r)(1-Q_{1:\tau})}.
\end{aligned}$$

For the $\mathcal{M}_{th}$, the score vector is

$$\frac{\partial \ell_i}{\partial p_j} = \frac{y_{ij}}{p_{ij}} - \frac{1-y_{ij}}{1-p_{ij}} - \frac{Q_{1:\tau}/(1-p_{ij})}{1-Q_{1:\tau}}, \qquad j = 1, \ldots, \tau,$$

and the EIM elements are

$$\begin{aligned}
-\mathsf{E}\left(\frac{\partial^2 \ell}{\partial p_j^2}\right) &= \frac{1-p_j-Q_{1:\tau}}{p_j\,(1-p_j)^2\,(1-Q_{1:\tau})^2}, \\
-\mathsf{E}\left(\frac{\partial^2 \ell}{\partial p_j\,\partial p_k}\right) &= \frac{p_j\,p_k\,Q_{1:\tau}(1-Q_{1:\tau}) + Q_{1:\tau}^2}{(1-Q_{1:\tau})^2\,(1-p_j)\,(1-p_k)}, \qquad j \neq k.
\end{aligned}$$

B. Fitted values

By default all the family functions have fitted values corresponding to the probabilities in the conditional likelihood function (2), viz.

$$\widehat{p}_{ij}^{\,y_{ij}}\,(1-\widehat{p}_{ij})^{1-y_{ij}} \cdot \left[1 - \prod_{s=1}^{\tau}\left(1-\widehat{p}_{i,cs}\right)\right]^{-1}.$$

Alternatively, the unconditional means of the Y_j can be returned as the fitted values upon selecting `type.fitted = "mean"` argument. They are $\mu_1 = E(Y_1) = p_{c1}/(1-Q_{1:\tau})$, $\mu_2 = [(1-p_{c1})\,p_{c2} + p_{c1}\,p_{r2}]/(1-Q_{1:\tau})$, and for $j = 3, 4, \ldots, \tau$,

$$\mu_j = (1-Q_{1:\tau})^{-1}\left\{p_{cj}\,Q_{1:(j-1)} + p_{rj}\left[p_{c1} + \sum_{s=2}^{j-1} p_{cs}\,Q_{1:(s-1)}\right]\right\}.$$

Affiliation:

Jakub Stoklosa
School of Mathematics and Statistics
The University of New South Wales
New South Wales 2052, Australia
E-mail: j.stoklosa@unsw.edu.au

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 65, Issue 5 *Submitted:* 2013-12-23
May 2015 *Accepted:* 2014-08-19