

EM and component-wise boosting for Hidden Markov Models: a machine-learning approach to capture-recapture

Robert W. Rankin^{a,*}

^a*Cetacean Research Unit, School of Veterinary and Life Sciences, Murdoch University, Australia*

1 Abstract

2 This study introduces statistical boosting for capture-mark-recapture (CMR) models. It is a shrinkage
3 estimator that constrains the complexity of a CMR model in order to promote automatic variable-selection
4 and avoid over-fitting. I discuss the philosophical similarities between boosting and AIC model-selection,
5 and show through simulations that a boosted Cormack-Jolly-Seber model often out-performs AICc methods,
6 in terms of estimating survival and abundance, yet yields qualitatively similar estimates. This new boosted
7 CMR framework is highly extensible and could provide a rich, unified framework for addressing many topics
8 in CMR, such as non-linear effects (splines and CART-like trees), individual-heterogeneity, and spatial com-
9 ponents.

10
11 *Keywords: capture-recapture, boosting, machine-learning, model-selection, marked animals, high-dimensional*
12 *data*

13 1. Introduction

14 Multi-model inference (MMI) has become an integral part of the capture-mark-recapture (CMR) litera-
15 ture. By CMR, I refer to the survey design and statistical modelling of abundance and survival of marked
16 animals under imperfect detection, using individual time-series of recaptures. By MMI, I loosely refer to a
17 variety of strategies such as model-selection, model-averaging, and regularization techniques such as shrinkage
18 estimators (e.g. some random-effects models; Royle & Link, 2002) and sparse estimators. A good overview is
19 by Leeb & Pötscher (2009). These strategies may be used to address research goals such as: finding ecolog-
20 ically important covariates; deciding which model-cum-hypothesis has most support; incorporating “model
21 uncertainty” into estimates; or seeking parsimony in estimation, such as estimating survival across sex and
22 age classes, and doing so without over-fitting.

23 Among these related goals, we may categorize them into two distinct objectives: estimation/prediction
24 vs. selection of the “correct” model or “best approximating” model. Often, these two objectives cannot be
25 achieved by the same MMI procedure (Shao, 1993; Yang, 2005; Leeb & Pötscher, 2005; Vrieze, 2012; Aho

*Corresponding author. E-mail: robertw.rankin@gmail.com

26 et al., 2014). Estimation is generally heralded by shrinkage-estimators and the Akaike Information Criterion
27 (AIC; Akaike, 1998, 1974), whereas selection is championed by sparse-estimators and the Schwarz-Bayes
28 Criterion (BIC; Schwarz, 1978). This paper will introduce a new MMI technique for capture-mark-recapture
29 called “boosting”, and I will show how it fits into the two domains of MMI.

30 Boosting is a type of shrinkage estimator, a class of techniques that (crudely) achieve the goals of MMI
31 with a single smoothing model. Crucially, model complexity can “shrink” along a continuum, in contrast
32 to all-subsets model-selection where there is a discrete set of fixed-effect models with different numbers of
33 parameters. Shrinkage estimators were first motivated by Royle & Link (2002) for CMR, in which case they
34 advocated for a random-effects Bayesian model. In this paper, I present a new boosting algorithm, which
35 could be considered as the Frequentist answer to Royle & Link.

36 To understand shrinkage, consider the classic example of survival and its fixed-effects extremes: time-
37 varying survival vs. time-constant survival. In CMR notation, these are known as $\phi(t)$ and $\phi(\cdot)$, respectively.
38 The former is difficult to reliably estimate, whereas the latter is often a poor reflection of reality. Shrinkage
39 estimators will achieve an intermediate solution between the two extremes. In other words, $\phi(t)$ is shrunk
40 towards $\phi(\cdot)$.

41 The question then becomes, how much shrinkage? To Bayesians, like Royle & Link (2002), the answer is
42 to use prior distributions. To a Frequentist, the amount of shrinkage is decided by prediction error: we find a
43 model that can both explain the observed data and make good predictions on new data. CMR practitioners
44 may not think of themselves as seeking models with good predictive performance, but their tool of choice, the
45 AIC/c, is based on a predictive error called the KL-loss (Akaike, 1974, 1998). Likewise, boosting methods are
46 highly *efficient* at minimizing prediction error and estimation error (Bühlmann & Yu, 2003; Meir & Rätsch,
47 2003). This makes boosting very philosophically similar to model-selection by AIC (Leeb & Pötscher, 2009).
48 Therefore, boosting should be of great interest to CMR practitioners who are already using the AIC for
49 model building.

50 However, boosting can do things that AIC/c model-selection cannot. For example, it can include splines
51 for non-linear effects (e.g., a non-linear change in survival with age). It can include classification and regression
52 trees (CART; Hothorn et al., 2006) for automatic discovery of higher order interactions (such as a three way
53 interaction of sex, time, and age on capture-probability). It can include spatial effects (Kneib et al., 2009;
54 Tyne et al., 2015). It can deal with “high-dimensional” covariate data, such as sorting through dozens or
55 hundreds of potential environmental variables, even under small sample sizes. It also does a better job of
56 handling “model uncertainty” under the scourge of multi-collinearity (Mayr et al., 2014), which troubles the
57 model-averaging approach (Cade, 2015). Boosting is also related to many other types of popular techniques,
58 such as being a type of Generalized Additive Model (Schmid et al., 2010; Hofner et al., 2014) and ℓ_1 -
59 regularization (a.k.a. the Lasso; Bühlmann & Yu, 2003; Efron et al., 2004; Tibshirani, 2011). This versatility
60 has led some to call boosting the “unified framework for constrained regression” (Hofner et al., 2014). This
61 paper introduces this powerful framework to CMR.

62 Many of the above benefits should interest CMR practitioners (especially believers of the AIC approach).
63 Perhaps most importantly, boosting excels in one particular domain which is terribly onerous for all-subsets
64 model-selection: the scourge of high-dimensionality. Every additional covariate leads to an exponential
65 increase in the number of possible fixed-effect models. This is due to the multi-parameter nature of CMR
66 models: we must perform model-selection on both the survival parameter as well as the capture parameter. In
67 this paper, I will consider an example with just three covariates (sex, time, and an environmental covariate)
68 which results in 64 fixed-effects models. With a fourth and fifth covariate, the number of fixed-effect models
69 would explode to 196 and 900, respectively. This computational burden is quickly prohibitive for all-subsets
70 model-selection, with even a small number of covariates. Consequentially, some recent CMR studies using
71 AIC/c model-selection have taken computational shortcuts, such as step-wise selection (Pérez-Jorge et al.,
72 2016; Taylor et al., 2016), an out-dated procedure that is strongly discouraged for many reasons (Burnham
73 et al., 2011). In contrast, boosting can sort through all covariates and their interactions in just one model,
74 because covariate selection is integrated within the fitting procedure.

75 I will introduce CMR boosting for the two-parameter open-population Cormack-Jolly-Seber model (CJS;
76 Cormack, 1964; Jolly, 1965; Seber, 1965), for estimating survival and abundance under imperfect detection.
77 The simplicity of the CJS will suffice to prove the new boosting algorithm for CMR data; such data is
78 not possible to analyze using conventional boosting algorithms. Conventional boosting methods assume
79 independent data-points in order to perform gradient descent (i.e., step-wise minimization of a loss function),
80 whereas CMR capture-histories consist of serially-dependent observations. The key innovation of this paper is
81 to garner conditionally independent observations by imputing time-series of latent states, a routine trick from
82 Hidden Markov Models (HMM). In CJSboost, we alternate between boosting the parameters (conditional on
83 latent states) and imputing expectations of the latent states (conditional on the parameters), and repeating
84 *ad infinitum*. I will prove this framework on the simple and manageable CJS model, with the ultimate
85 goal to refine the method on more complex models, such as POPAN and the Robust Design and spatial
86 capture-recapture.

87 By focusing on a simple CJS model, I will also elucidate some of the technical challenges and limitations
88 of boosting. The most obvious challenge is the computational burden of multiple cross-validation steps.
89 Another less obvious limitation is that boosting is generally unsuitable for making inferences about the “true
90 model” or discriminating among truly influential covariates vs non-influential covariates, i.e., it is not model-
91 selection *consistent*. This is true for all procedures that are optimized for prediction/estimation, including
92 the AIC/c (Yang, 2005; Leeb & Pötscher, 2009; Vrieze, 2012; Aho et al., 2014). These loss-efficient procedures
93 have a well-known tendency to prefer slightly more-complex models (Shao, 1997) and they can result in false
94 discoveries when misused to find the “true model”. As a possible remedy, I suggest combining CJSboosting
95 with a new regularization-resampling technique called stability selection (Meinshausen & Bühlmann, 2010) to
96 make inferences about which covariates are truly influential. Therefore, CMR practitioners can use CJSboost
97 for either efficient estimation or consistent model-selection/model-identification.

98 First, I will provide some background theory about model-selection and shrinkage, as well as a brief
 99 introduction to conventional boosting algorithms. Then, I will use simulations and a classic dataset (Lebreton
 100 et al., 1992) to illustrate CJSboost and benchmark it to AICc model-selection and model-averaging. Finally,
 101 I will end with a simulation that is computationally impossible for AICc-based inference: model-selection of
 102 a CJS model with 21 covariates. This is unheard of in CMR, until now.

103 For R code (R Core Team, 2016) and a tutorial, see the online content at [http://github.com/
 104 faraway1nspace/HMMboost/](http://github.com/faraway1nspace/HMMboost/).

105 2. Methods

106 2.1. Background

107 2.1.1. Capture-Recapture and the Cormack-Jolly-Seber Model

108 Imagine that we wish to study the abundance and survival of an open-population of animals. At regular
 109 time-intervals $t \in \{1, 2, 3, \dots, T\}$, we randomly capture, mark, and release individual animals. In subsequent
 110 $t \geq 2$, we recapture some of these already-marked animals with probability $p_{i,t}$, conditional on an animal being
 111 alive at t . Animals may die between capture periods $t-1$ and t , or survive with probability $\phi_{i,t}$. Recaptures
 112 are scored as the binary outcome $y_{i,t} \in \{0, 1\}$ for $\{no\text{-capture}, re\text{-capture}\}$. \mathbf{y}_i is the time-series of captures
 113 for individual i , called a *capture-history*. The ragged matrix $\mathbf{Y}^{(n \times T)}$ includes the capture-histories of all n
 114 unique individuals who were observed.

115 Our goals are two-fold: i) to estimate the abundance of marked animals N_t for each capture period
 116 $t > 1$; and ii) estimate survival ϕ , including its sources of variation, such as temporal variation or individual
 117 variation. The above formulation is the Cormack-Jolly-Seber open population model (Cormack, 1964; Jolly,
 118 1965; Seber, 1965). We can estimate the parameters $p_{i,t}$ and $\phi_{i,t}$ by maximizing the CJS likelihood:

$$p(\mathbf{y}_i | \phi_i, \mathbf{p}_i, t_i^0) = \left(\prod_{t > t_i^0}^{t_i^*} \phi_{i,t} (p_{i,t})^{y_{i,t}} (1 - p_{i,t})^{1 - y_{i,t}} \right) \chi_i^{(t_i^* + 1)} \quad (1)$$

119 where t_i^0 is the capture-period in which individual i was first captured; t_i^* is the capture-period when
 120 individual i was last observed; and $\chi_i^{(t_i^* + 1)}$ is the probability of never being seen again after t_i^* until the
 121 end of the study, $\chi_i^{(t)} = (1 - \phi_{i,t}) + (1 - p_{i,t})\phi_{i,t}\chi_i^{(t+1)}$. Notice that $\chi_i^{(t)}$ is calculated recursively. Given
 122 $\hat{p}_{i,t}$, we can estimate the abundance of animals at time t using a Horvitz-Thompson-type estimator: $N_t =$
 123 $m_t^0 + \sum_i^n \frac{1[y_{i,t}=1 \ \& \ t_i^0 > t]}{\hat{p}_{i,t}}$, where m_t^0 is the number of animals whose first capture was at time t (McDonald &
 124 Amstrup, 2001).

125 A key point is that the captures are serially-dependent and cannot be considered independent; in other
 126 words, the CJS likelihood (1) is evaluated on an entire capture-history, *not* per capture. This is mathe-
 127 matically embodied by the recursive term $\chi_i^{(t)}$. For gradient descent algorithms, like boosting, we require
 128 independent data points. If we reformulate the CMR system as a HMM, we can garner conditional indepen-
 129 dence through the use of latent states $z_{i,t} \in \{0, 1\}$ to represent $\{dead, alive\}$. When $z_{i,t} = 1$, then individual i
 130 is alive and available for capture at time t , and the probability of a capture is simply $p(y_{i,t}=1 | z_{i,t}=1) = p_{i,t}$.

131 However, if $z_{i,t} = 0$ then individual i is dead and unavailable for capture at time t ; therefore the probability
132 of a capture is zero.

133 The use of latent states and boosting is not new (Ward et al., 2009; Hutchinson et al., 2011). The novelty
134 of the CJSboost approach is that the latent states obey Markovian transition rules and form a serially-
135 dependent time-series. For example, a trailing sequence of no-captures $\mathbf{y}_{t:T} = [0, \dots, 0]^T$ has many possible
136 state-sequences, but once $z_t = \text{dead}$ then also z_{t+1} must equal dead . Fortunately, we can utilize well-developed
137 HMM tools to estimate all the permissible state-sequences \mathbf{z} . This is a key point which will be developed
138 further when I describe the CJSboost algorithm.

139 2.1.2. Prediction, Estimation and Generalization Error

140 There are many types of MMI techniques that share an implicit property of making optimal predictions.
141 This is true for shrinkage estimators, like boosting, and the AIC and their cousins (i.e. what Aho et al.,
142 2014, called “A-type” thinking). Here, prediction has a more technical meaning than, e.g., the layman idea
143 of weather forecasting or predicting the next USA president. It means that if we collect a new sample of
144 data $y^{(\text{new})}$ from the population \mathbb{Y} , our predictive model should be able to accurately estimate the $y^{(\text{new})}$
145 values. More formally, we wish to minimize the error in predicting $y^{(\text{new})}$, for all theoretical data-sets that
146 we might randomly sample from the population distribution of \mathbb{Y} . Notice that this predictive framework is
147 not explicitly about testing hypotheses nor accurate estimation of parameters, but it nonetheless serves as
148 a principled means of model-building: we desire a model that is complex enough to fit to the observed data
149 and make good predictions on new data, but does not over-fit the observed data. This is one way to codify
150 *parsimony*.

151 We can formalize this intuition as the following. Consider that we have a family of models \mathbb{G} which map
152 covariate information \mathbb{X} to the response variable, i.e., $G: \mathbb{X} \rightarrow \mathbb{Y}$. Our sample of data $\{y_j, \mathbf{x}_j\}_{j=1}^n$ arises
153 from the unknown population distribution P . The optimal model G is that which minimizes the following
154 *generalization error*:

$$\mathcal{L}(y, G(\mathbf{x})) = \int \ell(y, G(\mathbf{x})) dP(y, \mathbf{x}) = \mathbb{E}_P \left[\ell(y, G(\mathbf{x})) \right] \quad (2)$$

155 where ℓ is a *loss* function: it scores how badly we are estimating y from $G(\mathbf{x})$. \mathcal{L} is the *expected loss*, a.k.a, the
156 *risk* (Bühlmann & Yu, 2003; Meir & Rätsch, 2003; Murphy, 2012a). Here, the integral is just a mathematical
157 way of saying that we are minimizing the loss over the entire theoretical population, and any new samples
158 from this population.

159 There are many types of loss-functions. Akaike (1998) makes the case for using the (negative) log-
160 Likelihood; in which case Eqn. 2 becomes equivalent to minimizing the Expected (negative) log-Likelihood
161 (which is not to be confused with Maximum Likelihood Estimation). In fact, the Expected log-Likelihood is
162 seen in Eqn 1.1 of Akaike’s seminal derivation of the AIC (Akaike, 1998). This emphasizes the fundamental
163 similarity between the AIC and any estimator that minimizes (2).

164 While minimizing the expected loss is ostensibly about predicting new values of the response variable y ,
165 it also has desirable properties for estimation. This is crucially important because CMR practitioners are
166 not interested in making predictions about new capture-histories. Instead, we want to minimize the error
167 of estimating abundance and survival. Fortunately, as noted by Akaike (1974), minimizing the Expected
168 (negative) log-Likelihood is *efficient*. This means that by minimizing the expected loss (2) we also minimize
169 the square-error between the estimated model parameters and their true values. This connection is straight-
170 forward in multiple linear regression (Copas, 1997), but may only be approximately true for capture-mark-
171 recapture. Through simulations, I will explore this estimation error for the CJS and the AICc (sections
172 2.4).

173 2.1.3. Regularization and shrinkage

174 One cannot measure the expected loss or generalization error (2); it requires having data for the entire
175 population. Instead, we are forced to work only with our sample of data, and proceed to minimize the
176 *empirical risk*:

$$L(\mathbf{y}, G(\mathbf{X})) = \frac{1}{n} \sum_{j=1}^n \ell(y_j, G(\mathbf{X}_j)) \quad (3)$$

177 The difference between (2) and (3) is that the former integrates the loss over the entire population, while the
178 latter only calculates the loss on the observed data. Minimizing the empirical risk is easy. In when the loss
179 function is the negative log-likelihood, it is the Maximum Likelihood solution. But, at finite sample sizes, it
180 tends to *over-fit* a sample, make bad predictions, and have higher estimation errors (Copas, 1983, 1997).

181 The question then becomes: how can we minimize something we cannot see (the generalization error),
182 when all we have to work with is the observed data and empirical risk? Akaike (1998, 1974) answered this
183 question with the AIC, which was to approximate the Expected (negative) log-Likelihood with $2L(\mathbf{y}, G(\mathbf{X})) +$
184 $2\|G\|_o$, where the second term is the number of parameters in G , a.k.a the ℓ_0 norm¹. The approximation
185 works well at large sample sizes for linear regression and auto-regressive models, but is less exact for CMR
186 models.

187 Another answer comes from Learning Theory, called regularization. The theory tells us that if we constrain
188 the complexity of our function space, we can use the same procedure that minimizes the empirical risk, but
189 still bound the generalization error (Bühlmann & Yu, 2003; Meir & Rätsch, 2003; Mukherjee et al., 2003).
190 Practically, this implies that we penalize the complexity of G and prevent the procedure from fully minimizing
191 L . Popular examples are the Lasso (Efron et al., 2004; Tibshirani, 2011) and Ridge regression, which have
192 penalties on the ℓ_1 - and ℓ_2 -norms, respectively; hence, they are known as ℓ_1 - and ℓ_2 -regularizers. Boosting
193 is generally equivalent to ℓ_1 -regularization (under certain circumstances; Efron et al., 2004; Bühlmann &
194 Hothorn, 2007).

¹In the standard AIC formula, the first term is negative. It is omitted here because I define L as the *negative* log-Likelihood.

195 In boosting, the principal means of regularization is by *functional gradient descent* and *early-stopping*.
196 Gradient descent means: i) we start with a very simple model $G^{(0)}$ that has a high empirical risk $L^{(0)}$; and
197 then ii) we take tiny steps that reduce L towards its global minimum, where each m^{th} step slightly increments
198 the complexity of the model $G^{(m)}$. If we run the gradient descent until $m \rightarrow \infty$, we would minimize the
199 empirical risk and get a *fully-saturated model* $G^{(m \rightarrow \infty)}$, which is generally equivalent to Maximum Likelihood
200 Estimation. But, we stop short at some $m_{\text{stop}} \ll \infty$. Figure 1 (*bottom panel*) shows the gradient of the
201 empirical risk.

202 Why would we want to stop-short and not maximize the model-fit to the data? It turns out that, at
203 finite sample sizes, the best predictors which minimize the generalization error have *shrinkage*: the estimates
204 are shrunk away from the MLEs of the fully-saturated model and are pushed towards the simple model
205 G^0 (Copas, 1983, 1997). Optimal predictors are never as extreme as the MLEs. This predictive principle
206 generally holds true for estimation as well; it was discovered as early as the 1950's by Stein (1956) and James
207 & Stein (1961). It was incendiary at the time because shrinkage estimators are *biased*. For example, Figure
208 2 compares true and estimated values from CJSboost, and I suspect most ecologists will find it alarming: it
209 clearly shows the bias of shrinkage. A simple way to understand the optimality of shrinkage is through the
210 idea of the “bias-variance trade-off”: we may be slightly biased but our estimates are likely to be closer to
211 the truth (low-variance), whereas the MLEs are unbiased but may vary wildly with a new sample of data
212 (high-variance). The Appendix E provides a primer about the bias-variance trade-off, and compares how
213 CJSboost and AIC methods each negotiate this trade-off to minimize an expected loss.

214 Of course, we cannot measure the expected loss, so we must approximate it with the *average holdout-*
215 *risk* using cross-validation or bootstrap-validation. We measure the empirical risk on out-of-sample subsets
216 of bootstrapped data. The goal is to tweak the complexity of the model, by varying the regularization
217 parameters, such that the average holdout-risk is minimized. Figure 1 (*top panel*) shows an example of
218 minimizing the average holdout-risk at $m = m_{CV}$. For a large number of bootstrap resamples, minimizing the
219 average holdout risk will also minimize the expected loss. The wondrous utility of AIC is that it is generally
220 equivalent to minimization of a leave-one-out cross-validation criteria (Stone, 1977; Shao, 1993, 1997).

221 2.1.4. Introduction to boosting

222 The previous sections pertained generally to shrinkage estimators and MMI. I will now tie these ideas
223 together with boosting before describing the CJSboost algorithm in section 2.2.1. This overview will focus
224 only on the statistical view of boosting, whereas its full history and origins in machine-learning can be found
225 in Meir & Rätsch (2003) and Mayr et al. (2014).

226 Statistical boosting can be thought of in two ways. One, it is an iterative method for obtaining a
227 statistical model, $G(X)$, via functional gradient descent (Breiman, 1998; Friedman et al., 2000; Friedman,
228 2001; Breiman, 1999; Schmid et al., 2010; Nikolay Robinzonov, 2013), where $G(X) = \hat{F}$ and \hat{F} is the *fit-*
229 *vector*, the expected values of Y based on covariate data X . Although boosting has origins in classification
230 algorithms, we now know that it is equivalent to regularized regression, such as the Lasso (Bühlmann & Yu,

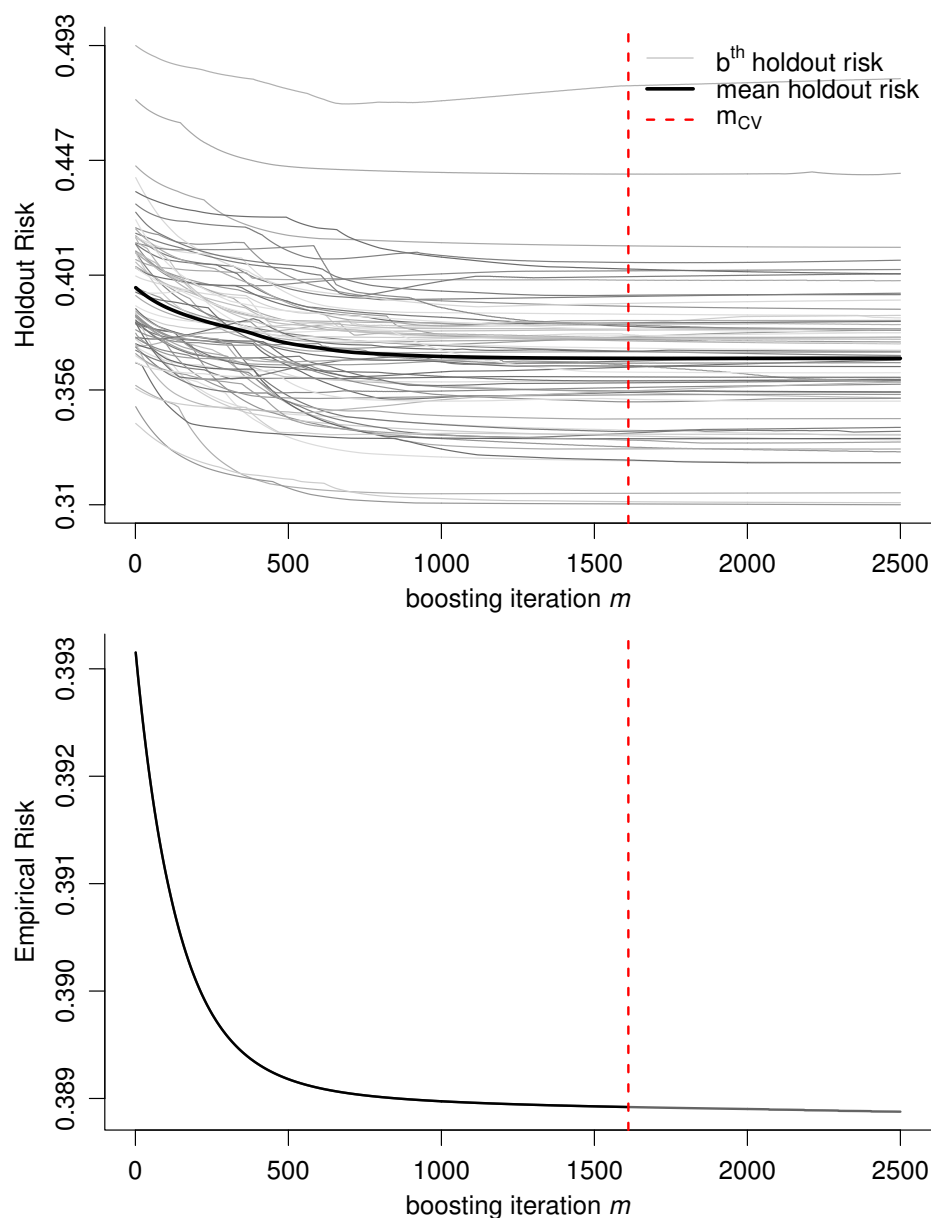


Figure 1: *Top*: Visualization of the step-wise minimization of *generalization error* by CJSboost (a.k.a the *expected loss* or *risk*), which is approximated by the mean holdout-risk (solid **black line**) from bootstrap-validation. Each m step along the x -axis is a boosting iteration, which adds one base-learner and increases the complexity of the model. At $m = m_{\text{cv}}$ (red dashed line), the mean holdout-risk is minimized; beyond m_{cv} the model is over-fitting. Each b^{th} gray line represents the holdout-risk predicted from one CJSboost model trained on a bootstrapped sample of capture-histories and then evaluating the holdout-risk on the out-of-sample data. *Bottom*: The empirical risk of the final statistical model using the full dataset. The model increases in complexity until it stops early at $m = m_{\text{cv}}$. The empirical risk is the negative log-Likelihood of the Cormack-Jolly-Seber model. Running the algorithm for $m \rightarrow \infty$ will result in the MLE solution. The difference between the MLE model and the model at $m = m_{\text{cv}}$ is *shrinkage*.

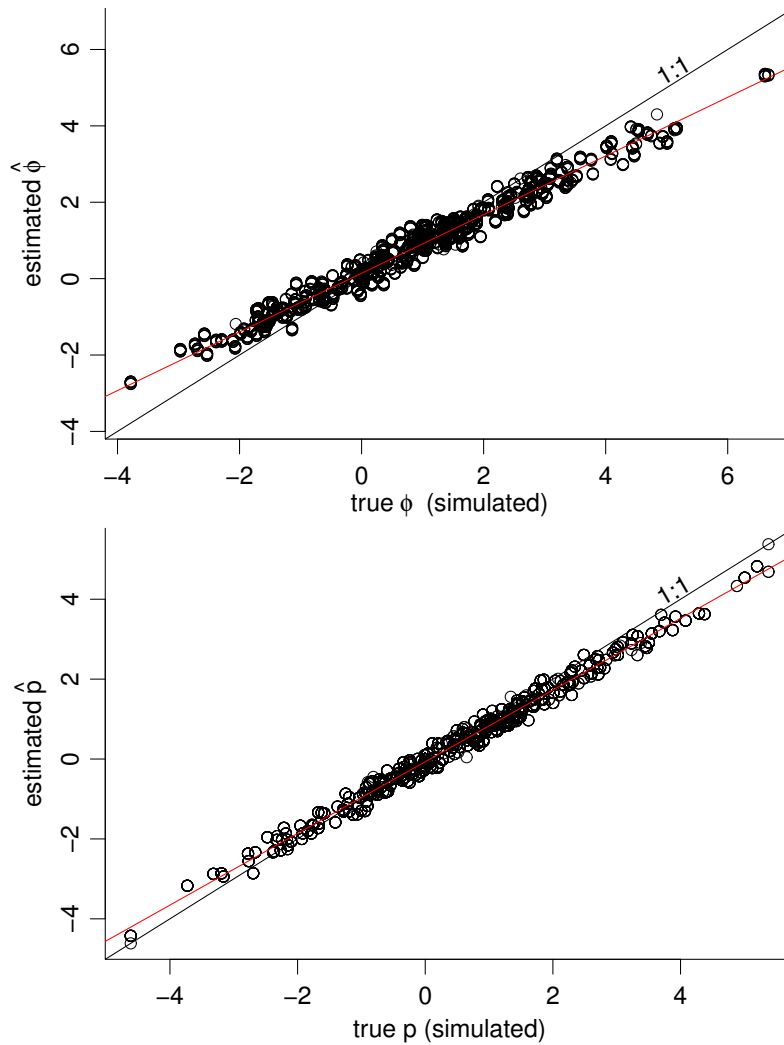


Figure 2: Visualization of shrinkage, by comparing the true (simulated) values of survival ($\phi_{i,t}$; *top*) and capture-probability ($p_{i,t}$; *bottom*) vs. the CJSboost-EM estimates. Each point is an individual i at capture-period t . The CJSboost estimates have some downward bias (evident in the difference between the 1:1 line and the estimates' red trend-line) due to shrinkage of coefficients to the intercept-only model. The amount of bias is our principle means of negotiating the “bias-variance trade-off” for optimal prediction.

231 2003; Efron et al., 2004, under certain conditions).

232 Second, boosting is the step-wise construction of an ensemble model $\mathcal{G} := \{g^{(1)}, g^{(2)}, \dots, g^{(m)}\}$, composed
233 of many weak prediction functions g , somewhat similar to model-averaging (Hand & Vinciotti, 2003). The
234 prediction functions arise from base-learners b , which are any function that can take data (x, y) and make a
235 predictor $g(x)$ to predict y from x , i.e. $b_k : (x, y) \Rightarrow g_k(x) = \hat{f}$. The fitting function b may be a Least-Squares
236 estimator b_{OLS} , or Penalized Least-Squares estimator b_{PLS} , or recursive-partitioning trees b_{trees} (a.k.a CART),
237 or low-rank splines b_{spline} , or many others. The variety of base-learners gives boosting more flexibility than
238 other shrinkage estimators or model-selection techniques. As an extreme example, if one uses Least-Squares
239 base-learners, b_{OLS} , and runs the boosting algorithm until $m \rightarrow \infty$, this unpenalized model will produce
240 regression coefficients that are nearly identical to a frequentist GLM.

241 Practically, we deliberately constrain the base-learners and keep them weak (Bühlmann & Yu, 2003).
242 Base-learners need only have a predictive performance of slightly better than random chance for the entire
243 ensemble to be strong (Schapire, 1990; Kearns & Valiant, 1994). The boosted ensemble results in a smooth
244 additive model of adaptive complexity:

$$G : (\mathbf{X}) \Rightarrow \sum_{m=1}^{m_{\text{stop}}} \nu \cdot g_k^{(m)}(\mathbf{X}_k) = \hat{F}^{(m_{\text{stop}})} \quad (4)$$

245 where each prediction function g_k is deliberately shrunk by the scalar parameter $\nu \in (0, 1)$, called the
246 *learning-rate*.

247 *Conventional boosting.* There are many flavours of boosting, but they all share a basic algorithm. The goals
248 are: i) to estimate the fit-vector $\hat{F} := \mathbb{E}[Y]$, which is the vector of our expected values of y ; and ii) to make
249 an ensemble of base-learners \mathcal{G} that can make predictions from new covariate data. Boosting is summarized
250 as: *i*) set the initial values of fit-vectors $F^{(0)}$ to the MLEs of the simplest model (such as the intercept-only
251 model); *ii*) increment m ; *iii*) use the current fit-vector $\hat{F}^{(m-1)}$ to estimate the negative-gradient of the loss-
252 function, $\hat{\mathbf{u}}^{(m)}$ (like the residual variation unexplained by the previous step); *iv*) make a prediction function
253 that maps X to $\hat{\mathbf{u}}^{(m)}$ and append the prediction function to the ensemble $\mathcal{G}^{(m)} \leftarrow g^*$; *v*) increment the
254 fit-vector with the predictions from g^* , shrunk by the scalar ν such that $\hat{F}^{(m)} = \hat{F}^{(m-1)} + \nu \hat{f}$; *vi*) repeat
255 steps *ii* to *v* until $m = m_{\text{stop}}$. The regularization parameters m_{stop} and ν govern the amount of shrinkage
256 (Bühlmann & Yu, 2003; Schmid & Hothorn, 2008a).

257 *Component-wise boosting.* The development of boosting from a classification algorithm into a statistical
258 modelling framework is credited to Bühlmann & Yu (2003). In their component-wise boosting framework,
259 the user specifies a large candidate set of base-learners, each representing a plausible set of sub-models for
260 different main effects and interactions and non-linear effects, etc. This is somewhat analogous to the way
261 in which a user would set-up a large candidate set of fixed-effect models for model-selection (but simpler).
262 Figure 3 shows a comparison of 64 different fixed-effect CJS models in Program Mark, and their equivalent
263 representations as base-learners for CJSboost. Variable selection is integrated internally to the descent

264 algorithm by selecting only one best-fitting base-learner per m iteration. In other words, base-learners
265 compete with each other to enter the ensemble, per m .

266 In component-wise boosting, the fitted ensemble \mathcal{G} contains the final selected base-learners, which can
267 be used to understand the functional relationships between covariate data and the response variable. For
268 example, if covariate x_1 has more predictive power than x_2 , we expect that the base-learner $b(x_1)$ to be
269 selected with greater frequency than $b(x_2)$. For least-square base-learners, we can retrieve the regression
270 coefficient of x_1 by adding up all the pertinent coefficients contained in \mathcal{G} , multiplied by ν . These have the
271 same meaning as the regression coefficients in a GLM (except they have shrinkage). More specifically, they
272 are almost equivalent to the regression coefficients of an ℓ_1 -regularizer like the Lasso (Bühlmann & Yu, 2003;
273 Efron et al., 2004).

274 *Multi-parameter boosting, or GAMLSS.* Another key development was the extension of boosting to include
275 multi-parameter likelihood functions (Schmid & Hothorn, 2008b; Schmid et al., 2010; Mayr et al., 2012),
276 sometimes called boosted-GAMLSS (or “GAMs for location, scale and shape”). This is a wide class of
277 interesting regression models such as Beta regression (Schmid et al., 2013) or Occupancy-Detection models
278 (Hutchinson et al., 2011) which have multiple parameters.

279 The multi-parameter problem is obvious in the CJS likelihood, where we have a parameter ϕ for survival
280 and a second parameter p for capture-probability. We must perform model-selection on both parameters.
281 The fit-vectors \mathbf{F} are no longer the expected values of the response variable Y (which does not interest us
282 in CMR); instead the fit-vectors $\mathbf{F} := \{\hat{F}_p, \hat{F}_\phi\}$ represent the expected values of the processes ϕ and p on
283 the logit scale, $\hat{\phi}_{i,t} = \frac{1}{1+e^{-\hat{F}_{\phi,i,t}}}$. Also, we now have different ensembles of base-learners for each parameter
284 $\mathcal{G} := \{\mathcal{G}_p, \mathcal{G}_\phi\}$.

285 The boosted-GAMLSS algorithm requires independent data-points, so it is not suitable for CMR. But, it
286 provides the mechanism to jointly boost the survival and capture processes. The key innovation of boosted-
287 GAMLSS was to estimate the negative gradient of the loss function by taking the partial derivatives of the
288 loss function with respect to each parameters’ fit-vector, $\hat{u}_{\theta,i} = -\frac{\partial \ell_i}{\partial F_\theta}$, conditional on the values of the other
289 fit vectors $F_{-\theta}$.

290 2.2. CJSboost

291 CJSboost combines all the aforementioned ideas of conventional boosting (functional gradient descent
292 by taking small regularized steps) and component-wise boosting (integrated variable selection) and multi-
293 parameter boosting (interweaving boosting steps for ϕ and p), but requires one more step to make boosting
294 applicable to CMR data. We must break the serial-dependence among individual captures within a capture-
295 history. In other words, we garner conditional independence of data-points, and then proceed with gradient
296 descent.

297 I developed two algorithms to achieve this conditional independence. CJSboost-MC uses stochastic impu-
298 tation of latent states; it is described in Appendix A. I will focus on another algorithm, CJSboost-EM, which

299 imputes and iteratively updates the expected values of latent states through an Expectation-Maximization
300 step.

301 2.2.1. The Expectation-Maximization Step

302 Expectation-Maximization (EM) is a common technique to adapt boosting to complex non-linear models,
303 such as Conditional Random Fields (Dietterich et al., 2004), presence-only species distribution models (see
304 the Appendix of Ward et al., 2009), and Generalized Additive Mixed-Models (Groll & Tutz, 2012).

305 The motivation is thus: our loss function, the negative CJS log-likelihood (1), can only be evaluated
306 *per capture-history*, and not per data-point/capture. Therefore, it cannot be boosted because there is no
307 point-wise evaluation of the negative gradient. As a technical remedy, we use a slightly different *surrogate*
308 loss function which can be evaluated per data-point. This surrogate loss function is derived from the negative
309 *Complete-Data log-Likelihood* (CDL). The CDL can be evaluated per capture because it assumes that we
310 know the latent states $(z_{i,t}, z_{i,t-1})$ at t and $t-1$. The negative CDL is:

$$\begin{aligned}
 -\text{CDL}(y_{i,t}, z_{i,t}, z_{i,t-1} | F_{\phi,i,t}, F_{p,i,t}) = & -\mathbf{1}[z_{i,t-1}=1, z_{i,t}=1] \left(\log \left(\frac{1}{1+e^{-F_{\phi,i,t}}} \right) + y_{i,t} \log \left(\frac{1}{1+e^{-F_{p,i,t}}} \right) \right. \\
 & \left. + (1-y_{i,t}) \log \left(\frac{1}{1+e^{F_{p,i,t}}} \right) \right) \\
 & -\mathbf{1}[z_{i,t-1}=1, z_{i,t}=0] \log \left(\frac{1}{1+e^{F_{\phi,i,t}}} \right) \\
 & -\mathbf{1}[z_{i,t-1}=0, z_{i,t}=0]
 \end{aligned} \tag{5}$$

311 where y and z are defined as above in (1) and $\hat{F}_{p,i,t}$ and $\hat{F}_{\phi,i,t}$ are the fit-vectors for the capture-probability
312 and survival parameters, respectively, on the logit scale.

313 Using the negative CDL, we derive the surrogate loss function for the EM-step. It is called a “Q-
314 function”. The idea is to replace the values of $(z_{i,t-1}, z_{i,t})$ in (5) with their *two-slice marginal* expectations:
315 $w_t(q, r) := p(z_{t-1}=q, z_t=r | \mathbf{y}, \mathbf{F})$. $w_t(q, r)$ is the joint probability of $z_{t-1}=q$ and $z_t=r$, conditional on the fit
316 vectors \mathbf{F} and the data \mathbf{y} . The two-slice marginals $\{w(1, 1), w(1, 0), w(0, 0)\}$ can easily be computed with a
317 standard “forwards-backwards” HMM algorithm (Rabiner, 1989; Murphy, 2012b), as detailed in Appendix
318 B. This must be done in-between boosting steps.

319 To simplify notation, we will index each capture $y_{i,t}$ of individual i at time t with the index $j := (i, t)$.
320 This also emphasizes how each capture is conditionally independent given z . The Q-function is:

$$\begin{aligned}
 q(y_j, \{F_{j,\phi}, F_{j,p}\}) = & -w_j(1, 1) \left(\log \left(\frac{1}{1+e^{-F_{j,\phi}}} \right) + y_j \log \left(\frac{1}{1+e^{-F_{j,p}}} \right) + (1-y_j) \log \left(\frac{1}{1+e^{F_{j,p}}} \right) \right) \\
 & - w_j(1, 0) \log \left(\frac{1}{1+e^{F_{j,\phi}}} \right) \\
 & - w_j(0, 0)
 \end{aligned} \tag{6}$$

321 The q formula has a clear intuition: we are *weighting* three conditional loss functions that represent the three
 322 plausible latent state transitions: *alive* \rightarrow *alive*, vs. *alive* \rightarrow *dead*, vs. *dead* \rightarrow *dead* (the fourth scenario of
 323 *dead* \rightarrow *alive* is not permissible).

324 According to the theory of EM, by minimizing the surrogate loss function q , we also minimize the target
 325 risk function: the negative CJS log-likelihood (1). The advantage of working with the surrogate loss function
 326 is that it is easy to calculate its point-wise gradient using partial derivatives: $\frac{\partial q}{\partial F}$ (7).

327 The two-slice marginal expectations $w(\cdot, \cdot)$ change with every update of $\hat{\phi}$ and \hat{p} . Therefore, we iteratively
 328 boost the parameters ϕ and p conditional on $w(\cdot, \cdot)$, and then update $w(\cdot, \cdot)$ conditional on $\hat{\phi}$ and \hat{p} . The
 329 expectations quickly converge and we fit a statistical CMR model that is optimal at prediction and has
 330 integrated variable selection.

331 2.2.2. CJSboost-EM algorithm

332 The formal CJSboost-EM algorithm is as followed. It is identical to the multi-parameter component-wise
 333 boosting algorithm of Schmid et al. (2010, §2), except for the additional EM-step (Step 5) and, of course,
 334 different loss and gradient functions (Step 6).

- 335 1. Specify the candidate set of plausible base-learners $\{b_k\}_{k=1}^K$, per ϕ and p .
- 336 2. Set the regularization parameters, m_{stop} , ν_ϕ and ν_p ; e.g. $m_{\text{stop}} = 10^3$; $\nu_\phi = 0.01$.
- 337 3. Initialize the fit vectors at the MLEs of a simple intercept-only model

$$\hat{\mathbf{F}}^{(0)} := \left\{ \hat{F}_\phi^{(0)} = \text{logit}(\hat{\phi}(\cdot)), \hat{F}_p^{(0)} = \text{logit}(\hat{p}(\cdot)) \right\}$$

- 337 4. Set $m = 1$.
- 338 5. Estimate the two-slice marginal probabilities $\{w_j(1, 1), w_j(1, 0), w_j(0, 0)\}_{j=1}^J$ for all individuals and
 339 capture-periods, using the forwards-backwards algorithm (see Appendix B.3).
- 340 6. Estimate the gradients of the surrogate loss function q w.r.t the fit vectors $\hat{\mathbf{F}}^{(m-1)}$:

$$\begin{aligned} \hat{u}_{j,\phi}^{(m)} &= -\frac{\partial q_j}{\partial F_\phi^{(m-1)}} = \frac{w_j(1, 1) - w_j(1, 0)e^{\hat{F}_{j,\phi}^{(m-1)}}}{\left(1 + e^{\hat{F}_{j,\phi}^{(m-1)}}\right)} \\ \hat{u}_{j,p}^{(m)} &= -\frac{\partial q_j}{\partial F_p^{(m-1)}} = \frac{w_j(1, 1) \left(1 + e^{\hat{F}_{j,p}^{(m-1)}}\right) y_j - w_j(1, 1)e^{\hat{F}_{j,p}^{(m-1)}}}{1 + e^{\hat{F}_{j,p}^{(m-1)}}} \end{aligned} \quad (7)$$

- 340 7. For each parameter θ in $\{\phi, p\}$, do:
 - 341 (a) for each k base-learner for θ , do:
 - 342 i. fit the base-learner to the gradient: $b_k(\hat{\mathbf{u}}_\theta^{(m)}, X_k) \Rightarrow g_k$;
 - 343 ii. make an estimate of the gradient, $\hat{f}_k = g_k(X_k)$;
 - (b) find the base-learner that best-fits the gradient:

$$k^* = \underset{k}{\operatorname{argmin}} (\hat{\mathbf{u}}_\theta^{(m)} - \hat{f}_k)^2 \quad (8)$$

- 344 (c) append the prediction function of k^* to the ensemble $\mathcal{G}_\theta \leftarrow g_{k^*}$;

- 345 (d) re-estimate the fit vector: $\hat{F}_\theta^{(m)} = \hat{F}_\theta^{(m-1)} + \nu_\theta \hat{f}_k^*$;
 346 8. Monitor the empirical risk on the full data $L(\mathbf{Y}, \hat{\mathbf{F}}^{(m)})$. Or, monitor the holdout-risk using an out-of-
 347 sample subset of the data $L(\mathbf{Y}_{\text{oos}}, \hat{\mathbf{F}}_{\text{oos}}^{(m)})$ s.t. $\hat{\mathbf{F}}_{\text{oos}}^{(m)} = \{G_\phi^{(m)}(\mathbf{X}_{\text{oos}}), G_p^{(m)}(\mathbf{X}_{\text{oos}})\}$ to use for bootstrap-
 348 validation.
 349 9. Update $m = m + 1$.
 350 10. Repeat steps 5 to 9 until $m = m_{\text{stop}}$.

351 The three regularization parameters m_{stop} , ν_ϕ , ν_p control the shrinkage, and must be tuned by minimizing
 352 the average holdout-risk. This is our estimate of the expected loss (see 2.2.3).

353 The outputs of the algorithm are the fit vectors $\hat{\mathbf{F}}$ and the ensemble of fitted base-learners \mathcal{G}_ϕ and \mathcal{G}_p . We
 354 can estimate the survival of individual i at time t by back-transforming the fit-vectors onto the probability
 355 scale: $\hat{\phi}_{i,t} = \text{logit}^{-1}(\hat{F}_{\phi,i,t})$. We do the same for capture-probability $\hat{p}_{i,t}$. For abundance, we use the Horvitz-
 356 Thompson-type estimator: $\hat{N}_t = m_t^0 + \sum_i^n \mathbf{1}[y_{i,t} = 1 \ \& \ t_i^0 > t] / \hat{p}_{i,t}$ (McDonald & Amstrup, 2001). For
 357 predicting ϕ^* and p^* on new covariate data \mathbf{X}^* , we merely process the data through the ensemble of fitted
 358 base-learners and shrink by ν , i.e., $\hat{F}_\theta^* = G_\theta(\mathbf{X}^*) = \nu_\theta \sum_{g_k \in \mathcal{G}_\theta} g_k(\mathbf{X}^*)$.

359 The second algorithm, CJSboost-MC, is described in Appendix A.

360 2.2.3. Regularization parameters

361 In multi-parameter boosting, the most important regularization parameters are m_{stop} , ν_ϕ , ν_p , which
 362 control the shrinkage. To guarantee a prediction optimal model, we must tune m_{stop} , ν_ϕ , ν_p with cross-
 363 validation or bootstrap-validation. As per Schmid et al. (2013), I suggest bootstrapping the individual
 364 capture-histories between 50 to 100 times, training a new model on each bootstrap sample. On average,
 365 each bootstrap leaves 36.5% of the capture-histories unused in the model fitting, which can then be used to
 366 estimate a holdout-risk.

367 Finding the optimal value of m_{stop} is straight-forward and routine in conventional boosting. See Fig-
 368 ure 1 for an example of bootstrap-validation used to estimate m_{cv} . Tuning the Real-valued ν_p and ν_ϕ
 369 computationally expensive and requires some careful consideration. This challenge is inherent to all multi-
 370 parameter boosting algorithms, including boosted-GAMLSS models (Schmid et al., 2013; Mayr et al., 2012)
 371 and CJSboost. Practitioners should see Appendix C for my proposed method and other ideas.

372 Finally, there are complexity parameters associated with individual base-learners that must be decided
 373 *a priori* and could be considered as regularization parameters, e.g., the effective-degrees-of-freedom of a
 374 Penalized Least-Squares base-learner, or the maximum tree-depth of a conditional inference tree. The effects
 375 of these parameters have been studied in conventional component-wise boosting (Bühlmann & Yu, 2003;
 376 Schmid & Hothorn, 2008a; Kneib et al., 2009). Practitioners should read Appendix D for best-practises, as
 377 well as the tutorial by Hofner et al. (2012).

378 2.3. Sparsity and Consistency

379 The previous discussions were predicated on prediction and minimizing the expected loss of estimation.
 380 There is another type of *model-identification* inference which is focused on finding the “correct” model, such

381 as declaring one covariate to be truly influential and another covariate to be spurious (what Aho et al., 2014
382 calls “B-type” thinking). If we declare a spurious covariate to be important, it is a False Discovery (FD).
383 If we declare a truly influential covariate to be unimportant, it is a False Rejection (FR). For this type of
384 inference, the chief desirable property is *consistency*: that we will, with high probability, find the correct
385 model as sample size increases. It turns out that prediction-optimal methods are generally not consistent
386 estimators and will tend to produce FDs (Shibata, 1980, 1986a; Shao, 1993, 1997; Yang, 2005; van Erven
387 et al., 2012)

388 Whether or not a procedure is consistent and/or efficient is mediated by one’s assumptions about the
389 dimensionality of the true generative process (i.e., the number of parameters in the true model). Consistent
390 procedures assume *sparsity*: the true generative model has a finite number of covariates, most covariates have
391 zero effect, and the dimensionality stays constant as sample size increases. The truth is the truth regardless
392 of sample size. This is a controversial assumption (Burnham & Anderson, 2004). For example, some authors
393 believe that the truth is never sparse: natural phenomena are complex with an infinite number of influences.
394 CMR theorists generally believe that as sample size increases, an MMI procedure should reveal more of these
395 small influences (Otis et al., 1978, pages 50-51). The AIC and prediction-optimal methods may be consistent
396 under this latter assumption, so long as one’s models are also approximately infinite-dimensional (Shibata,
397 1980). Conversely, consistent procedures, such as the BIC (Schwarz, 1978), Bayes Factors, the adaptive lasso
398 (Zou, 2006), and twin-boosting (Bühlmann & Hothorn, 2010), will severely under-fit if the truth is not sparse
399 (Sun et al., 2013) and will have unbounded maximum expected loss (Shibata, 1986b; Leeb & Pötscher, 2008).

400 These distinctions have been more-or-less ignored in the ecological literature (but see Burnham & Ander-
401 son, 2004; Link & Barker, 2006; Aho et al., 2014; Galipaud et al., 2014). In the CMR field, consistency and
402 model-identification has been much less important than estimating abundance, but some examples do exist
403 (e.g. Pérez-Jorge et al., 2016; Taylor et al., 2016).

404 Interestingly, two recent papers by Meinshausen & Bühlmann (2010) and Bach (2008) have proposed
405 similar ways to use ℓ_1 -regularizers, like the Lasso and boosting, in order to find truly influential covariates
406 under “high-dimensional” situations: when the sample size is small and there are many potential covariates.
407 The idea is to subsample/resample the data, and tally the frequency that each covariate is selected by an
408 ℓ_1 -regularizer, over the entire space of the regularization parameter (e.g., m in boosting). Some authors have
409 suggested that these are Frequentist approximations to Bayesian posterior inclusion probabilities (Richardson,
410 2010; Draper, 2010; Murphy, 2012c). I will loosely refer to these procedures as “stability selection”, although
411 there is a lot of subtle variation in this rapidly evolving field of research. In particular, its application
412 in multi-parameter boosting, like boosted-GAMLSS or CJSboost, is still unvalidated. See Appendix F for
413 clarifications.

414 There are two key points. First, this type of MMI is no longer about prediction nor estimation, but uses
415 prediction-optimal methods as an intermediate step for correct model-identification, i.e., which covariates are
416 part of the true model. Second, posterior inclusion probabilities lead to straight-forward inferences: covariates

417 with high inclusion probabilities are probably more important; covariates with low inclusion probabilities are
418 probably not that important.

419 Thus, CJSboost offers a choice to CMR practitioners. If one’s goals are to estimate abundance or survival,
420 then one can use the vanilla CJSboost model tuned for optimal prediction. Or, if one’s goals are to find
421 covariates that significantly effect survival, then one can use the stability-selection-enhanced CJSboost and
422 calculate inclusion probabilities. This choice is analogous to switching from the AIC to the BIC.

423 2.4. Simulation 1: Estimation

424 The first simulation investigated the ability of CJSboost to estimate abundance and survival, over different
425 sample sizes. Technically, I demonstrate that minimizing the average holdout-risk also minimizes the square-
426 error of estimating abundance and survival, as benchmarked against AICc model-selection and AICc model-
427 averaging. I used the AICc because it is supposed to excel at precisely this kind of task: minimizing
428 estimation error. I focused on metrics of *relative efficiency*, because this exemplifies the choice faced by
429 Frequentist practitioners: to choose among procedures based on their relative performance to get as close as
430 possible to the truth, over all theoretical data-sets.

431 I tested two CJSboost-EM models: i) a linear-model called b_{PLS} -CJSboost, which used least-square base-
432 learners, as listed in figure 3; and ii) a non-linear model, called b_{trees} -CJSboost, which used conditional
433 inference trees (Hothorn et al., 2006). The AICc-methods used 64 fixed-effects models listed in figure 3.

434 The simulated data-sets were inspired by the European Dipper dataset from Lebreton et al. (1992). There
435 were $T=10$ primary periods and two sexes of individuals ($\mathcal{X} \in \{1, 2\}$). Individuals’ first-capture periods (t_i^0)
436 were random. The true processes were time-varying effects plus an individual sex effect (\mathcal{X}). The true
437 data-generating processes ² were: $\phi(t, \mathcal{X}) = 0.91 - 0.01t - 0.05 \cdot \mathbf{1}[t = 5, 6] + 0.05 \cdot \mathbf{1}[t = 9, 10] - 0.05 \cdot \mathbf{1}[\mathcal{X} = 1]$
438 and $p(t, \mathcal{X}) = \text{logit}^{-1} \left(q + t \frac{\sin(t)}{17} \right) - 10 \cdot \mathbf{1}[\mathcal{X} = 1]$, where q controlled the mean capture-probability. Figure
439 5 graphs an example simulation. For analyses, there was an additional categorical variable, called *Flood*,
440 which grouped the captures periods $\{4, 5, 6\}$: it simulates an analyst’s hypothesis that dipper survival and
441 capture-probability are different in periods 4, 5 and 6, due to environmental degradation by flooding.

442 For each simulation and estimator, the mean standardized square error (MSE) was calculated for abun-
443 dance ($N_{t,\mathcal{X}}$) and survival ($\phi_{t,\mathcal{X}}$), e.g. $MSE-\hat{N} = \sum_{\mathcal{X} \in \mathcal{X}} \sum_{t=2}^T \frac{(\hat{N}_{t,\mathcal{X}} - N_{t,\mathcal{X}}^{(\text{true})})^2}{\text{Var}[N_{t,\mathcal{X}}]}$. A lower MSE is better. We
444 compared the estimators’ MSE values by two statistics: i) the *observed efficiency* of estimator i , which is
445 $\frac{\text{MSE}_{\min}}{\text{MSE}_i} \in (0, 1]$ (higher is better), where MSE_{\min} is the MSE of the best performing estimator; and ii) *rank*,
446 which is the rank-order of estimates by increasing values of MSE (rank 1 is best). These criteria were used
447 by early researchers of the AIC and BIC (Shibata, 1980; McQuarrie, 1999). Both criteria are empirical ways
448 of approximating the more fundamental Frequentist value of *relative efficiency*. Better values imply that a
449 procedure has, over repeated sampling, estimates that are closer to the truth (but not necessarily unbiased).

²Despite the existence of an implicit “true model”, the performance of the estimators were *not* judged on their ability to find it. Rather, the AIC and boosting are supposed to find/produce a model that minimizes the Expected negative log-Likelihood.

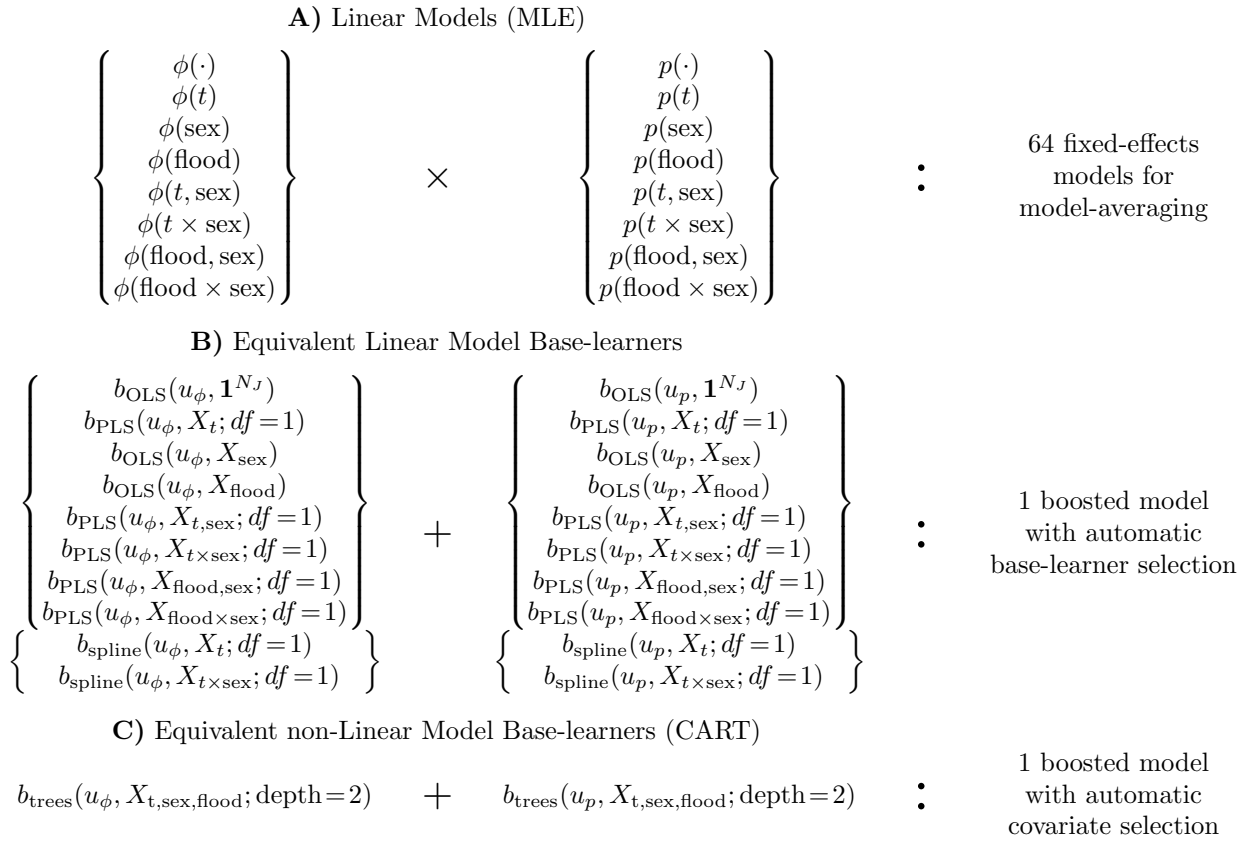


Figure 3: Different notation for multimodel inference of a Cormack-Jolly-Seber model, comparing fixed-effects model-averaging and boosting. **A)** Each fixed-effect model includes one term for ϕ (*left*) and one for p (*right*). $\theta(\cdot)$ is an intercept model; $\theta(t)$ has different coefficients per T capture periods (with appropriate constraints on $t=T$); $\theta(a, b)$ is a linear combination of covariate a and b on the logit scale; $\theta(a \times b)$ are main-effects plus the interaction between a and b on the logit scale. **B)** Equivalent linear base-learners (Ordinary and Penalized Least Squares from `mboost`; Bühlmann & Hothorn, 2007) with penalties to constrain their effective- df . All base-learners are available in one model; selection of base-learners is by component-wise boosting. **C)** A CJS model with CART-like trees, allowing non-linear effects and complex interactions. Selection of covariates is internal to the base-learners' `ctree` algorithm (Hothorn et al., 2006).

450 The observed efficiency and rank calculations were summarized according to sample size scenarios: differ-
 451 ent combinations of average capture-probabilities $p \in \{0.2, 0.4, 0.65\}$ and the number of captured individuals
 452 $n \in \{50, 100, 200, 400, 800\}$. I ran 20 simulations per combination of n and p .

453 All boosting models used 70-times bootstrap-validation to estimate optimal values of m_{stop} , ν_ϕ and ν_p .
 454 The base-learners were taken from the `mboost` R package (Bühlmann & Hothorn, 2007; Hofner et al., 2012).
 455 The AICc model-averaging analyses were conducted in Program MARK (White & Burnham, 1999) and
 456 RMark (Laake, 2013).

457 2.5. Analysis: Dipper Example

458 Using CJSboost-EM, I reanalyzed the European Dipper dataset from Lebreton et al. (1992). I compared
 459 the results to the MLEs of the fully-saturated model ($\phi(t \times \text{sex})p(t \times \text{sex})$) as well as to AICc model-averaged
 460 estimates. The dataset has 294 individuals in $T = 7$ capture periods. Covariates included time, sex, and

461 flood, similar to Section 2.4. The model-building framework was the same as in Figure 3. 100-fold bootstrap-
462 validation was used to optimize m_{stop} , ν_ϕ and ν_p .

463 Interested readers can repeat this analysis using the online tutorial at [http://github.com/farawayinspace/
464 HMMboost/](http://github.com/farawayinspace/HMMboost/).

465 2.6. Simulation 2: Sparsity and Consistency

466 The final simulation addressed the issue of high-dimensionality and the ability of CJSboost (EM) to find a
467 sparse set of important covariates out of many spurious covariates. This type of *model-identification* inference
468 is distinct from the estimation/prediction goals of shrinkage estimators and AIC approaches. The loss-
469 function is no longer about minimizing a square-estimation error, but is focused on limiting False Discoveries
470 (FD) and False Rejections (FR). For this task, one desires an estimator that is *model-selection consistent*;
471 which is to say, it will make zero FDs and FRs with probability 1 as sample size gets large.

472 Practically, this challenge is inappropriate for fixed-effect model-selection, because one must consider all
473 combinations of covariates for different parameters (ϕ, p) . In this section, I simulated 21 multi-collinear
474 covariates, resulting in more than 4 trillion different fixed-effects models (excluding two-way interactions). It
475 is clearly impossible for all-subsets model-selection (unless one takes ill-advised short-cuts).

476 2.6.1. Stability Selection and Inclusion Probabilities

477 Theoretically, this challenge is also inappropriate for the vanilla CJSboost or other shrinkage estimators.
478 Instead, I propose to use a bootstrapped-enhanced CJSboost to produce a consistent estimator. The crux of
479 this estimator is to approximate the Bayesian probability that a covariate is part of the “true model”, a.k.a.
480 posterior inclusion probabilities, $\pi(I_{\theta,k}|\mathbf{Y}, \mathbf{X})$. We desire such probabilities because they lead to inferences
481 about the significance of covariates.³ Influential covariates should have very high inclusion probabilities
482 that converge to 1 at high sample sizes, while spurious covariates should have low probabilities. In this
483 simulation, I will show the distribution of inclusion probabilities for truly-influential and spurious covariates,
484 as a function of different sample sizes.

485 Inclusion probabilities are a fundamentally Bayesian quantity, but Frequentist approximations are desir-
486 able for significance testing in a multi-model framework (Lee & Boone, 2011). Some authors (Richardson,
487 2010; Draper, 2010; Murphy, 2012c) noticed that such an approximation is possible through Stability Se-
488 lection plus ℓ_1 -regularization (Meinshausen & Bühlmann, 2010; Shah & Samworth, 2013). The idea is to
489 subsample/resample the data and tally the number of times that a covariate is selected by an ℓ_1 -regularizer,
490 over all values of the regularization parameter (m, ν_ϕ, ν_p) . To calculate the approximate inclusions probab-
491 ities, $\tilde{\Pi}_{\theta,k}$, I propose the following procedure: set the values of ν_ϕ and ν_p to their prediction-optimal values
492 $\hat{\nu}$; bootstrap the capture-histories B times; for each b bootstrap, run CJSboost for m_{stop} iterations, where
493 $m_{\text{stop}} \gg m_{\text{cv}}$. Stability selection probabilities, \hat{S} , are estimated by scoring whether a k^{th} covariate is selected
494 in a b bootstrap before m iterations (conditional on $\hat{\nu}$), $\hat{S}_{\theta,k}^{(m)|\hat{\nu}} = \frac{1}{B} \sum_{b=1}^B \mathbf{1}[k \in \mathcal{G}_\theta^{(b,m)}|\hat{\nu}]$. Notice that $\hat{S}_{\theta,k}^{(m)|\hat{\nu}}$

³This is not to be confused with classical Null Hypothesis Tests of the marginal effect of regression coefficients.

495 is evaluated per m and per covariate k and per parameter $\theta \in \{\phi, p\}$. $\hat{S}_{\theta,k}^{(m)|\nu}$ will always increase with m
 496 (i.e., weaker ℓ_1 -regularization will always increase the chance of selecting a covariate; see Figure 8). Call
 497 $I_{\theta,k}^{(\text{true})}$ the indicator of whether the k^{th} covariate is part of the true model, then the inclusion probability is
 498 approximated by $\pi(I_{\theta,k}|\mathbf{Y}, \mathbf{X}) \approx \tilde{\Pi}_{\theta,k}^{(m_{\max})|\nu} = \frac{1}{m_{\max}} \sum_{m=1}^{m_{\max}} S_{\theta,k}^{(m)|\nu}$.

499 From a Bayesian perspective, it is like we have a prior distribution on the model-coefficients that is the
 500 exponential of the negative regularization parameter (m) (Geman et al., 1992), and we are crudely integrating
 501 over the prior to score selection indicators. Technically, we should integrate over ν_ϕ and ν_p as well as m .
 502 I propose focusing on m strictly for computational convenience, but this short-cut needs further validation.
 503 Readers should refer to Appendix F to see how the above formulation relates to the existing literature on
 504 stability selection (Bach, 2008; Meinshausen & Bühlmann, 2010; Schmid et al., 2012; Shah & Samworth,
 505 2013; Hofner et al., 2015).

506 2.6.2. Simulating Data

507 In 240 simulations, I use the following generative model for survival and capture-probability:

$$\text{logit}(\theta_{i,t}) = \beta_{\theta,0} + \underbrace{\sum_{k=1}^{21} \beta_{\theta,k}^T x_{i,k}}_{\text{individual effects}} + \underbrace{\sum_{\tau=2}^T \beta_{\theta,\tau} \mathbf{1}[\tau=t]}_{\text{capture period effect}}$$

508 The intercepts were drawn randomly from $\beta_{p,0} \sim U(0.4, 0.6)$ and $\beta_{\phi,0} \sim U(0.55, 0.8)$. I simulated 21 multi-
 509 collinear covariates (18 continuous, three discretized) drawn from a multivariate Gaussian with marginal
 510 variances of 1 and off-diagonal correlations between 0 to 0.6. Time-as-a-categorical-variable ($\{\beta_t\}_{t=2}^T$) was
 511 also included as a possible influential covariate, for a total of 22 “covariates”. The number of captured
 512 individuals was stratified as $n \in \{50, 100, 200, 400, 800, 1600\}$. There were $T=10$ capture periods.

513 The values of the true β coefficients were drawn randomly according to two different scenarios: A) *sparsity*,
 514 in which case a few β^* values were large but most β values were zero (i.e., many spurious covariates); and B)
 515 *tapering*, in which case the values of the $|\beta^*|$ decreased exponentially from one or two large values, to many
 516 small-but-nonzero values. I ran 120 simulations per scenario A and B. I highlight these scenarios because
 517 sparsity is a fundamental assumption of all model-selection consistent procedures, whereas some authors
 518 suggest that tapering is more in-line with reality (Burnham & Anderson, 2004). Tapering also challenges the
 519 very notion of a “true model”, in which case we can only speak about the best approximating model (but see
 520 Link & Barker, 2006). In an extreme form of tapering, when the magnitudes of the β values actually increase
 521 with sample-size, consistent procedures can have a worst-case estimation error that becomes infinite (Leeb &
 522 Pötscher, 2008), which I highlight to remind practitioners of the price of this type of multimodel inference.

523 For the sparsity scenario (A), three covariates were randomly picked to have a significant effect, i.e. $\beta_\theta^* \neq 0$.
 524 These truly influential covariates, β_θ^* , had norms of 1 on the logit scale, resulting in large marginal effects
 525 ($\text{SD}(\beta_k^T \mathbf{x}_k) \approx 1$) that spanned 0.8–0.9 probability-units. When the β_θ^* were categorical variables, then they
 526 had norms of 3 in order to achieve a similar marginal effect. The coefficients were simulated separately for ϕ
 527 and p .

Table 1: **CJSboost vs AICc for estimating survival and abundance: results of simulation 1**

Model	Abundance \hat{N}_t		Survival $\hat{\phi}_t$	
	efficiency [†]	rank [‡]	efficiency	rank
minimum AICc model	0.55(0.22)	3.86(1.22)	0.42(0.26)	4.27(1.01)
AICc model-averaged	0.57(0.2)	3.24 (0.93)	0.49(0.27)	3.5(1.1)
b_{PLS} -CJSboost-EM	0.58 (0.2)	3.28(1.1)	0.64 (0.24)	2.86 (1.11)
b_{trees} -CJSboost-EM	0.55(0.19)	3.54(1.22)	0.61(0.22)	3.09(1.15)

[†] *observed efficiency*, $\text{MSE}_{\min}/\text{MSE}$, averaged over simulations (S.D. in parentheses).

[‡] *rank* of MSE, averaged over simulations (S.D. in parentheses).

bold values emphasize the best estimator.

528 For the tapering scenario (B), all β_θ values were non-zero. On average 5.6% of β had marginal effects
 529 categorized as “large” ($0.5 < \text{SD}(\beta_k^T \mathbf{x}_k) \leq 1$, or equivalently $0.5 < |\beta_k| \leq 1$), 13.9% were “moderate” ($0.25 <$
 530 $|\beta_k| \leq 0.5$), 37.3% were small ($0.05 < |\beta_k| \leq 0.25$) and 43.1% were negligible ($0 < |\beta_k| \leq 0.05$). The coefficients
 531 were simulated separately for ϕ and p .

532 2.6.3. Data Analysis

533 To analyze each simulated dataset, I use the following base-learners for each p and ϕ sub-model: 22
 534 PLS base-learners ($df = 2$) for each continuous and categorical covariate; a PLS base-learner for the time-
 535 as-a-categorical variable (a.k.a, the $\theta(t)$ model); and a base-learner for the intercept. In stability selection,
 536 base-learners must have equal flexibility/degrees-of-freedom; otherwise, the more complex base-learners will
 537 have a greater probability of being selected (see Section 2.2.3). The regularization parameters ν_p and ν_ϕ were
 538 optimized with ten 70-fold bootstrap-validation exercises, as per Section Appendix C.1.

539 2.6.4. Oracle Estimator

540 Finally, an auxiliary task was to derive an *oracle estimator* (Fan & Li, 2001; Zou, 2006). The goal
 541 is estimate the coefficients as if we knew the “true” model from the beginning, a property of all consistent
 542 procedures (Leeb & Pötscher, 2008). The idea is to threshold the inclusion probabilities at some high threshold
 543 $\pi_{\text{thr}} \in (0.5, 1)$, and use only those covariates where $\tilde{\Pi}_k > \pi_{\text{thr}}$ (called hard-thresholding). A final un-regularized
 544 CJSboost model is used to get “debiased” estimates by running $m \rightarrow \infty$ (Bach, 2008; Murphy, 2012c) ⁴. I
 545 showcase this oracle property on just one simulated dataset from scenario A, in order to demonstrate the
 546 role of the threshold π_{thr} in determining the oracle properties and the number of FDs and FRs.

547 3. Results

548 3.1. Simulation 1: CJSboost vs AIC

549 Table 1 and Figure 4 summarize the estimation performance of boosting-EM and AICc methods across
 550 all simulations. Figure 5 shows the model fits and the true processes for one example simulation ($n = 300$).

⁴After hard-thresholding, the final model may not have a unique MLE, such as as the $\phi(t)p(t)$ model. In such cases, one must impose constraints (such as $\phi_{T-1} = \phi_T$) before attempting to debias the results and run the algorithm until $m \rightarrow \infty$. Regularized CJSboosting does not have this problem because of shrinkage.

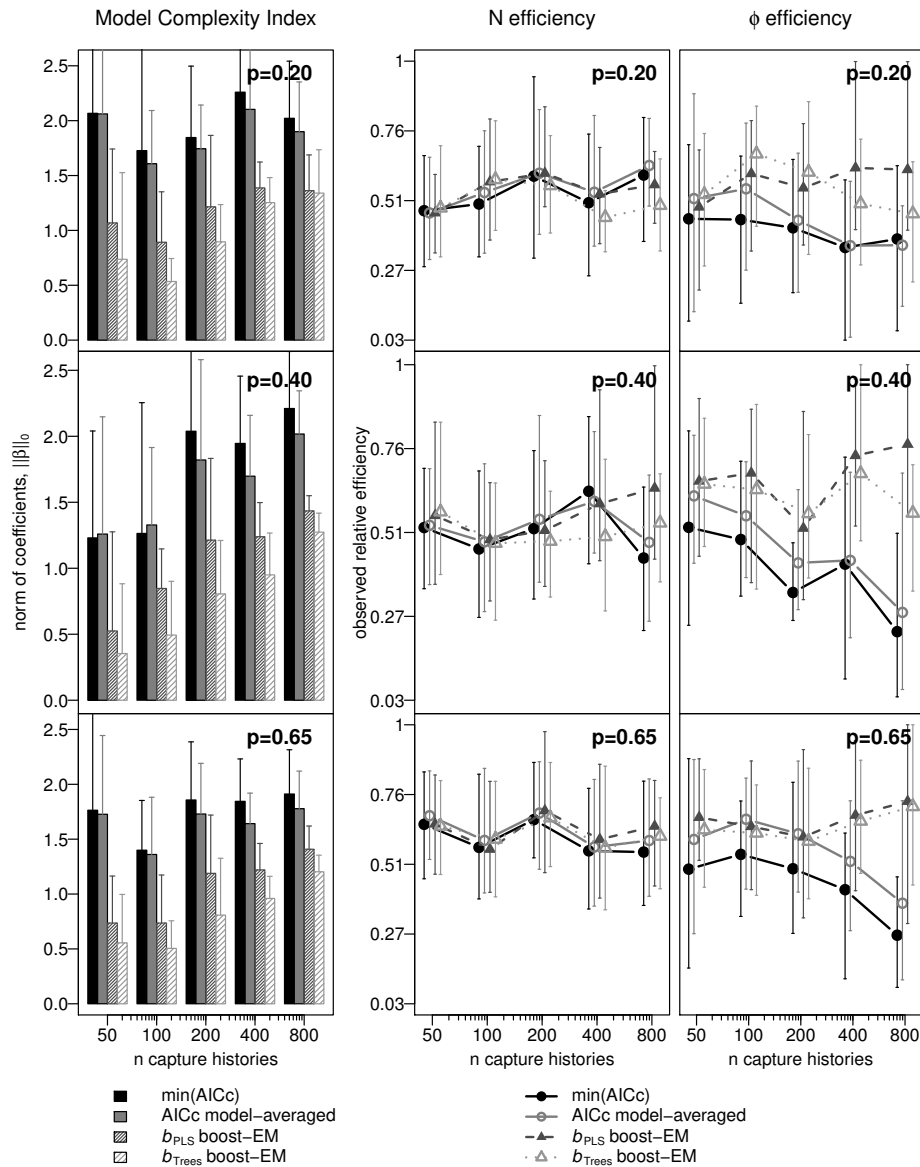


Figure 4: Simulations of Cormack-Jolly-Seber data-sets show how model complexity and estimation performance vary by sample-size (x -axes), true capture-probability ($p = 0.2, 0.4, 0.6$, *panel-rows*), and the multi-model inference paradigm: AICc methods (*thick-lines*) vs CJSboost methods (*dashed-lines*). *Left*: model-complexity increases as the sample-size increases, as measured by the absolute size of the estimated model coefficients (a.k.a the norm of $\hat{\beta}$). *Middle*: relative performance estimating abundance N_t , as measured by the average *observed efficiency* $\text{MSE}_{\min}/\text{MSE} \in (0, 1]$, where MSE_{\min} is the error of the best estimator. Higher efficiency is better. *Right*: The average observed efficiency of survival. Results are averaged over 20 simulations per combination of p (*panel-rows*) and n (x -axes).

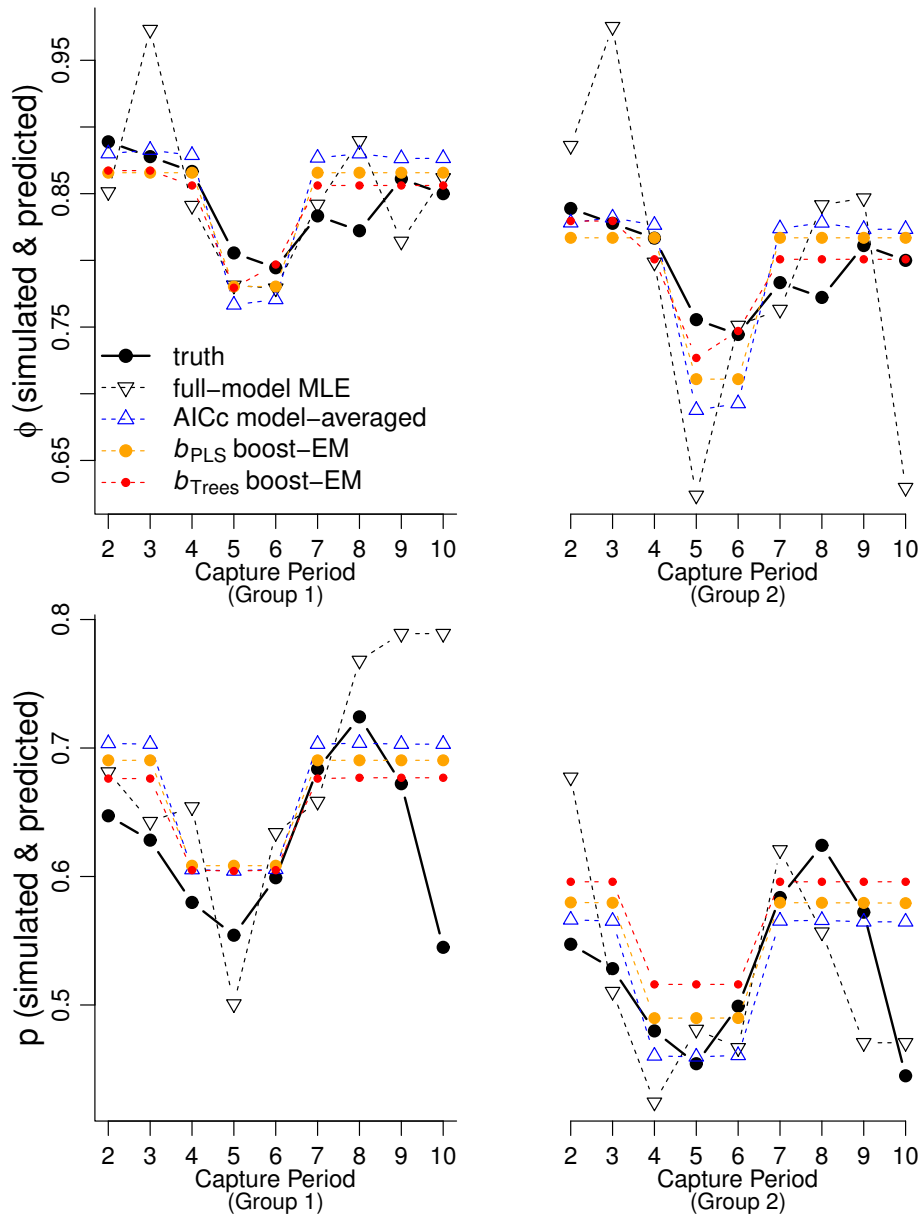


Figure 5: Simulation 1, demonstrating the CJSboost estimates from the Expectation-Maximization technique. A comparison of capture-probability estimates $\hat{p}(t \times x)$ and survival estimates $\hat{\phi}(t \times x)$ from models composed of linear base-learners (OLS and PLS; in orange) and non-linear base-learners (CART-like trees; in red), as well AICc model-averaging (blue) and MLE (dashed black).

551 The general result is that the b_{PLS} -CJSboost model with PLS base-learners did best at minimizing es-
 552 timation errors and obtaining higher relative efficiencies for both abundance (\hat{N}) and survival ($\hat{\phi}$), over all
 553 samples sizes, followed by AICc model-averaging, then b_{trees} -CJSboost with conditional inference trees. The
 554 worse performance was by the minimum AICc model.

555 Regarding abundance estimates, all four estimators had similar performances, with no discernible trend
 556 by sample size (n and p). b_{PLS} -CJSboost had slightly better performance according to the observed efficiency

557 criteria, while AICc model-averaging won narrowly according to the average MSE rank.

558 However, for survival, the CJSboost models clearly outperformed the AICc methods, especially with the
559 PLS base-learners: they obtained the highest overall efficiencies and best mean rank. The results varied by
560 n : when $n \leq 100$, all methods had similar performances; but when $n > 100$, the boosting methods greatly
561 out-performed both AICc methods.

562 To understand why boosting out-performed the AICc methods, it is helpful to look at the growth in the
563 magnitude of the model coefficients ($\|\beta\|$). According to theory on shrinkage, we would expect that $\|\beta\|$
564 would be smaller at low n and low p , for both boosting and AICc methods, to prevent over-fitting. The AIC
565 methods had more extreme coefficient values, especially at low n and low p . Therefore, AIC methods were
566 *underestimating* the correct amount of shrinkage necessary for optimal estimation. The b_{trees} models had
567 slightly lower coefficient norms than the better performing PLS models, which suggests that the tree-models
568 were *overestimating* the correct amount of shrinkage.

569 Interestingly, AICc model-averaging produced better estimates than the best AICc model, with more
570 shrinkage on coefficients. This is unsurprising for estimating abundance. However, there are theoretical
571 problems with model-averaging when it comes to estimating model parameters such as survival, especially
572 under collinearity (Cade, 2015) which is an inherent feature of CMR processes. At low sample sizes ($n = 50$)
573 both AICc methods had very high coefficient values, and a lot of variability. This may suggest that the AICc
574 approximation does not hold well for CMR models at very low sample sizes. Interestingly, the abundance
575 estimates were still competitive with boosting.

576 We can gain more insights into shrinkage by scrutinizing one example simulation (Figure 5). None of
577 the estimators did a convincing job of approximating the true underlying processes. The estimates from
578 boosting-EM and AICc-methods revealed similar patterns for both for ϕ and p , but they differed in the
579 amount of shrinkage: the boosted estimates were *shrunk to the mean* more than model-averaged estimates.
580 More shrinkage resulted in better MSE performance (despite the increase in bias). The tree base-learners had
581 perhaps too much shrinkage and worse MSE. The Figure also shows the MLEs to illustrate the bias-variance
582 trade-off: the MLEs of the full-model $\hat{\phi}(t \times \text{sex})\hat{p}(t \times \text{sex})$ are unbiased but are also high-variance, in the sense
583 that the estimates vary wildly around the true processes.

584 Figure 5 has been repeated in Appendix A using the the Monte-Carlo CJSboost algorithm.

585 3.2. Results: Dipper example

586 This section shows the reanalysis of the European Dipper dataset from Lebreton et al. (1992) by CJSboost-
587 EM. Comparisons were between the linear b_{PLS} -CJSboost model and the nonlinear b_{Trees} -CJSboost model
588 as well as model-averaged estimates by AICc, and the MLEs from the full-model $\phi(t \times \text{sex})p(t \times \text{sex})$. See
589 Figure 6 for the fitted processes. The results can be summarized:

- 590 1. For both survival ϕ and capture-probability p , the three predictive methods (AICc, b_{PLS} -CJSboost
591 or b_{trees} -CJSboost) had similar patterns, unlike the full-model MLE. The predictive models differed
592 according to the amount of shrinkage.

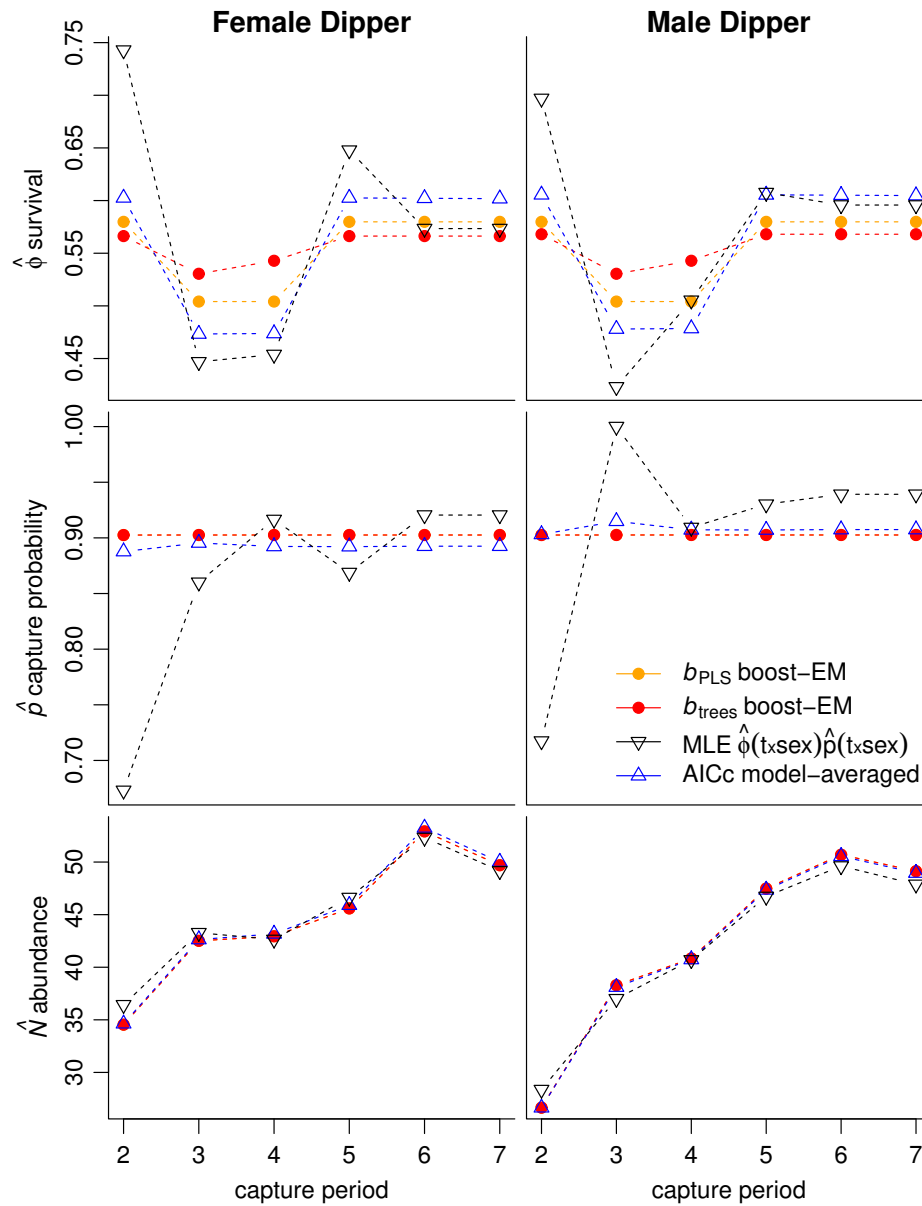


Figure 6: Comparison of Dipper survival ($\hat{\phi}$), capture-probability (\hat{p}), and abundance estimates (\hat{N}) according to three predictive models: i) CJSboost-EM using penalized least-squares base-learners (PLS), ii) CJSboost-EM using non-linear conditional inference trees, and iii) AICc model-averaging in Program MARK. Plus, the MLEs of the full-model $\hat{\phi}(t \times \text{sex}) \hat{p}(t \times \text{sex})$.

- 593 2. The b_{trees} -CJSboost model applied a lot shrinkage towards the time-constant values. Whereas the
594 AICc model-averaged estimates had less shrinkage and were more similar in pattern to the MLEs of the
595 full-model. The b_{PLS} model had shrinkage that was intermediate between the AICc and b_{trees} estimates.
596 3. For survival, all three predictive methods yielded the same estimates: a survival probability of 0.48-0.5
597 during the flood years ($t=3, 4$) and little-to-no sex-effect (< 0.005 difference between male and females).
598 4. For capture-probability, the model-averaged estimates suggested a slight sex effect of about 1.5 prob-
599 ability units, whereas both boosted models shrunk the capture-probability to a constant; in contrast,
600 the MLEs varied much more.
601 5. Abundance estimates showed little variation among methods, due to the high overall capture-
602 probabilities ($p \approx 0.9$).

603 3.3. Simulation 2: sparsity, consistency, and high-dimensional data

604 Figure 7 summarizes the results of 240 high-dimensional simulations and their inclusion probabilities
605 ($\tilde{\Pi}_{\theta,k,n}$) for truly influential and spurious covariates. The figure stratifies the average inclusion probabilities
606 by sample size (n), parameter $\theta \in \{\phi, p\}$, marginal effect sizes ($|\beta_{\theta,k}|$), and by the nature of the true model
607 (*sparsity* vs *tapering*). I remind readers that we desire $\tilde{\Pi}$ values of the truly influential covariates to converge
608 to 1 and be well separated from the $\tilde{\Pi}$ values of the spurious covariates.

609 The results are summarized according to the nature of the true model.

- 610 1. When the true model was *sparse* (i.e. three high-magnitude covariates and many spurious covariates)
611 the results were:
- 612 (a) For survival, there was a good separation of the $\tilde{\Pi}_{\phi}$ values between the truly influential covariates
613 and the spurious covariates, when sample sizes were $n \geq 100$. Ideally, we would prefer that the
614 *minimum* $\tilde{\Pi}$ of influential covariates is high and the *maximum* $\tilde{\Pi}$ of spurious covariates is low.
615 The average minimum $\tilde{\Pi}_{\phi,k,100}$ of the true covariates was 0.77 at $n=100$, and grew to $\gg 0.9$ for
616 $n > 200$. The average maximum $\tilde{\Pi}_{\phi,k,100}$ of the spurious covariates was 0.64 at $n=100$ and grew
617 to ≈ 0.75 at greater sample sizes. For spurious covariates, the overall average $\tilde{\Pi}$ stabilized and
618 plateaued below 0.5, while for the true covariates, the $\tilde{\Pi}_{\phi,k,100}$ values converged to 1 for $n > 200$.
 - 619 (b) For the covariates influencing capture-probabilities, there was less separation of the $\tilde{\Pi}_p$ values
620 between true covariates and spurious covariates, although the true covariates had $\tilde{\Pi}_p$ values which
621 converged to ≈ 1 by $n > 200$, and the spurious covariates remained below 0.5.
 - 622 (c) The time-as-a-categorical variable ($\beta_{\phi,t}$ and $\beta_{\phi,p}$), when spurious, had higher average $\tilde{\Pi}_{\theta}$ values
623 than the other spurious covariates. For ϕ , the average *maximum* $\tilde{\Pi}_{\phi}$ for $\beta_{\phi,t}$ was generally between
624 0.6 – 0.67. For p , the average *maximum* $\tilde{\Pi}_p$ for $\beta_{p,t}$ was generally between 0.8 – 0.85. This may
625 suggest a violation of the assumption of “exchangeability” among spurious covariates (Meinshausen
626 & Bühlmann, 2010).
 - 627 (d) Covariates that were spurious in ϕ but truly influential upon p (and *vice versa*) did not seem to
628 have $\tilde{\Pi}_{\phi}$ values that were different than the other spurious covariates. In other words, the true

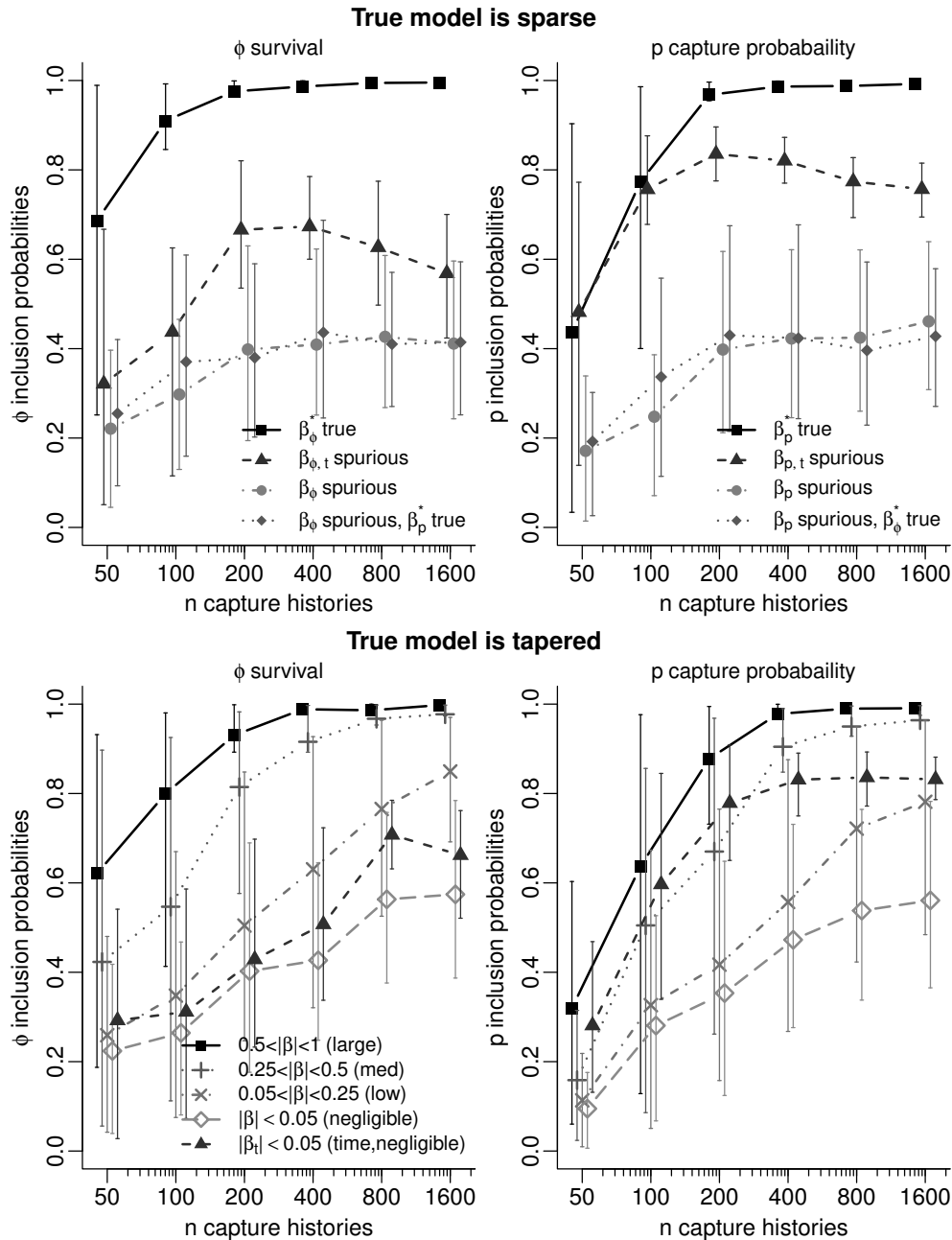


Figure 7: Results of 240 simulations to demonstrate the usefulness of approximate inclusion probabilities (on y -axes) for inference about which covariates are truly influential (i.e. part of the true model) vs. spurious covariates, over different sample sizes (x -axes). Each dot is an average inclusion probability over 20 simulations. Scenario A (top): the true model is sparse: only three covariates out of 22 are truly influential on ϕ or p (black squares); others are spurious (grey circles); some are spurious for ϕ but influential on p (grey diamonds) and vice-versa. Time-as-a-categorical variable, when spurious, is also plotted (dark triangles). Scenario B (bottom): the true model is tapered: all 22 covariates have some contribution to the ϕ/p -process, but they vary in the magnitude of their marginal effects ($|\beta_k|$). Bars are $\approx \pm 1$ S.D.

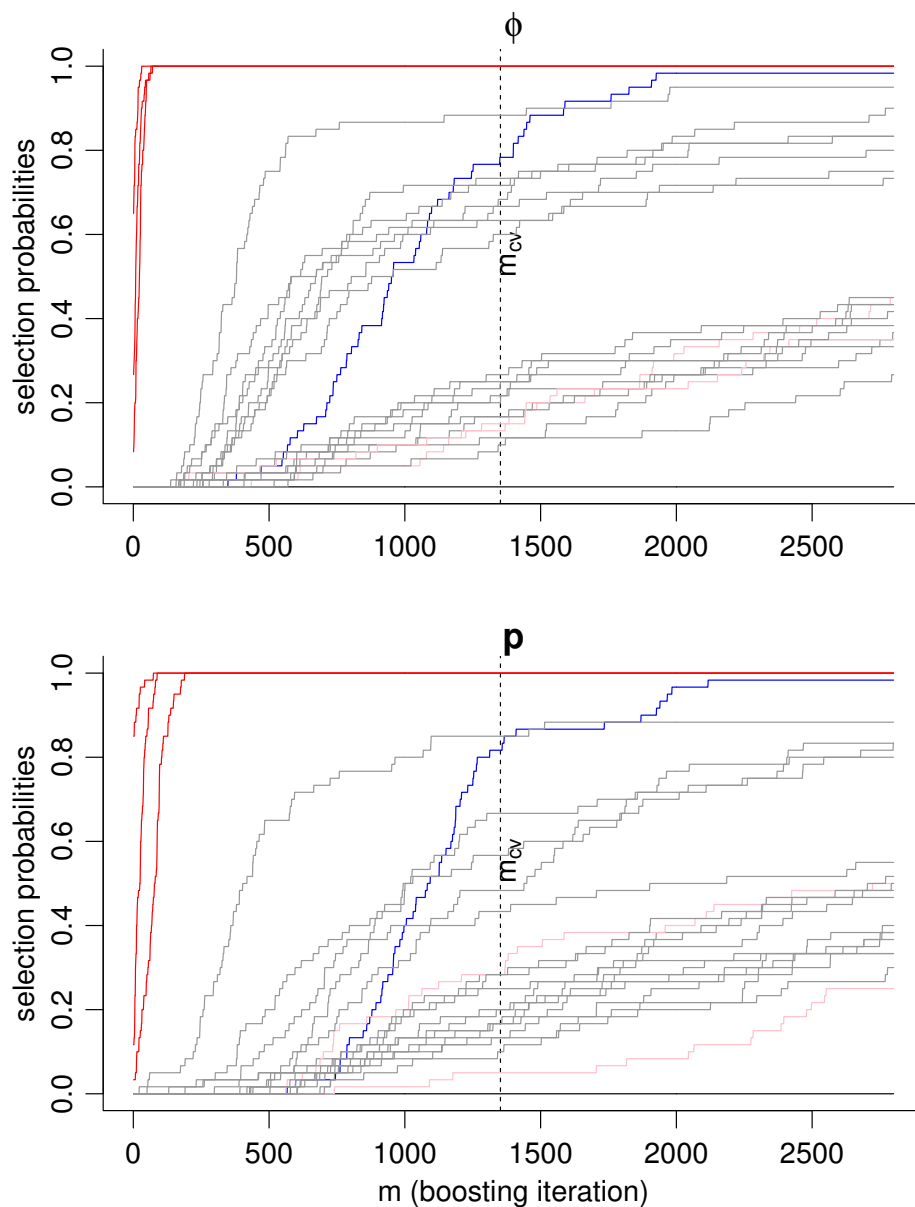


Figure 8: Demonstration of stability selection probabilities for one high-dimensional simulation. As the boosting iteration (m) gets large, regularization gets weaker, and all covariates have a higher selection probability S (estimated from a bootstrap). Lines in **red** are truly influential covariates. Lines in **gray** are non-influential covariates. Lines in **pink** are not-influential for θ , but are influential in the other parameter $-\theta$. Lines in **blue** represent the time-as-a-categorical-variable base-learner, a.k.a $\theta(t)$, which in this simulation was non-influential.

629 model of ϕ did not seem to influence the inclusion probabilities for the covariates in p , and *vice*
630 *versa*. This suggests that the assumption of exchangeability of spurious covariates may hold in
631 multi-parameter boosting.

632 2. When the true model was *tapered* (i.e. all covariates were part of the true model, but with decreasing
633 magnitudes of marginal effects) the results were the following:

634 (a) The overall pattern of $\tilde{\Pi}_\phi$ values behaved as one would expect. The covariates with *large* effects
635 had high $\tilde{\Pi}$ values that converged to 1 as n got large, while the covariates with *medium* and *small*
636 effects had lower average $\tilde{\Pi}$ values that increased as n got large, and the *negligible* effects had the
637 lowest average $\tilde{\Pi}$ values, but which nonetheless increased as n got large (although their average
638 remained below 0.5).

639 (b) Seemingly, all effect sizes had monotonic increases in inclusion probabilities with increasing sample
640 size. This was unlike the sparse scenario, where the $\tilde{\Pi}$ values seemed to plateau at their asymptotic
641 distributions.

642 (c) When time-as-a-categorical variable had negligible marginal effects, it nonetheless got higher $\tilde{\Pi}$
643 values than the other negligible covariates, especially for p . In other words, $\beta_{p,t}$ had a greater
644 propensity to be selected, even when it only had a tiny marginal effect.

645 We can also scrutinize the results of an example simulation (sparse, $n = 300$) and visualize the stability
646 selection pathways that were used to approximate the posterior inclusion probabilities $\tilde{\Pi}$. Figure (8) shows
647 how the truly influential covariates entered the ensemble very early (small m) and achieved stability selection
648 probabilities of $\hat{S}_k = 1$. There was a lot variability in the selection pathways of the spurious covariates, but
649 they generally increased as the amount of regularization got weaker (m got larger). Sometimes their \hat{S}_k
650 values did reach 1. Readers can view an online animated GIF which shows the stability paths for 30 example
651 simulations, at <http://github.com/farawayinspace/HMMboost/> and in the Supplementary Material.

652 The point of these simulations was to show that the inclusion probabilities ($\tilde{\Pi}$) may themselves be a
653 satisfactory end-point for an analysis. Alternatively, we can go one step further and *hard-threshold* the $\tilde{\Pi}$
654 values by π_{thr} and discard the covariates with $\tilde{\Pi}_k < \pi_{\text{thr}}$. See Table 3.3. If π_{thr} is too low, then some
655 spurious covariates will get selected and there are False Discoveries (FDs). If π_{thr} is too high, then some
656 truly influential covariates get Falsely Rejected (FRs). Meinshausen & Bühlmann (2010) suggest that this
657 threshold should be in the vicinity of 0.9 – 0.95, and my simulations support this threshold.

658 Hard-thresholding can also help us derive an oracle estimator and produce estimates that are the same
659 as a model run with 100% foresight about the true model. This type of inference seemingly blends the
660 two domains of MMI: estimation/prediction and consistent model-identification. Our oracle estimates are
661 produced by: i) setting π_{thr} ; ii) discarding spurious covariates $\tilde{\Pi}_{\theta,k} < \pi_{\text{thr}}$; iii) and running a final CJSboost
662 model with $m \rightarrow \infty$ (called “debiasing” by Murphy, 2012c, or “unregularized” by Bach, 2008). If our selection
663 procedure is model-selection consistent, then the new estimates should have oracle properties at large sample
664 sizes. This seems to be the case when the thresholds are high ($0.8 < \pi_{\text{thr}} < 0.99$), and both FDs and FRs are

Table 2: Estimates of coefficients from CJSboost, for one high-dimensional model-selection problem, under different degrees of hard-thresholding

Parameter	Survival Φ										MLE Oracle [‡]	SE Oracle
	Prediction	Inclusion Probability Threshold ^{††}										
	Optimal [†]	0.55	0.65	0.75	0.8	0.85	0.9	0.95	0.99			
$\hat{\beta}_\Phi(\text{time:1})$	-0.002	-0.01	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_\Phi(\text{time:2})$	-0.041	-0.238	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_\Phi(\text{time:3})$	-0.036	-0.271	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_\Phi(\text{time:4})$	-0.026	-0.285	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_\Phi(\text{time:5})$	0.017	0.205	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_\Phi(\text{time:6})$	0.006	-0.005	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_\Phi(\text{time:7})$	0.015	0.124	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_\Phi(\text{time:8})$	0.022	0.196	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_\Phi(\text{time:9})$	0.025	0.264	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_\Phi(\text{time:10})$	-0.001	-0.091	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_\Phi(\text{a})$	-0.083	-0.173	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_\Phi(\text{b})$	0.828	0.982	1.064	1.045	1.067	1.067	1.067	1.067	1.074	1.068	0.143	
$\hat{\beta}_\Phi(\text{c})$	-0.021	0	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_\Phi(\text{d})$	-0.761	-0.93	-0.991	-0.983	-0.965	-0.965	-0.965	-0.965	-0.919	-0.967	0.123	
$\hat{\beta}_\Phi(\text{e})$	0.175	0.262	0.288	0.303	0	0	0	0	0	0	0	0
$\hat{\beta}_\Phi(\text{f})$	0	0	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_\Phi(\text{g})$	0	0	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_\Phi(\text{h})$	0	0	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_\Phi(\text{i})$	-0.051	-0.107	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_\Phi(\text{j})$	0	0	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_\Phi(\text{k})$	-0.717	-0.838	-0.975	-0.968	-0.953	-0.953	-0.953	-0.953	-0.868	-0.955	0.119	
$\hat{\beta}_\Phi(\text{l})$	0	0	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_\Phi(\text{m})$	0	0	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_\Phi(\text{n})$	0	0	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_\Phi(\text{o})$	0	0	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_\Phi(\text{p})$	0	0	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_\Phi(\text{q})$	0	0	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_\Phi(\text{r})$	-0.048	-0.151	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_\Phi(\text{s:1})$	-0.034	-0.109	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_\Phi(\text{s:2})$	0.028	0.093	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_\Phi(\text{t:1})$	0	0	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_\Phi(\text{t:2})$	0	0	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_\Phi(\text{u:1})$	-0.061	-0.165	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_\Phi(\text{u:2})$	0.059	0.166	0	0	0	0	0	0	0	0	0	0
Capture-Probability p												
$\hat{\beta}_p(\text{time:1})$	0	0.002	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_p(\text{time:2})$	0	0.266	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_p(\text{time:3})$	0	-0.23	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_p(\text{time:4})$	0	-0.041	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_p(\text{time:5})$	0	-0.098	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_p(\text{time:6})$	0	0.159	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_p(\text{time:7})$	0	-0.04	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_p(\text{time:8})$	0	0.123	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_p(\text{time:9})$	0	-0.056	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_p(\text{time:10})$	0	-0.062	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_p(\text{a})$	0	0	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_p(\text{b})$	0.942	1.129	1.149	1.184	1.176	1.176	1.176	1.176	0.846	1.178	0.144	
$\hat{\beta}_p(\text{c})$	0	0	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_p(\text{d})$	0	0	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_p(\text{e})$	0	0	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_p(\text{f})$	-0.933	-1.142	-1.181	-1.189	-1.186	-1.186	-1.186	-1.186	-0.856	-1.189	0.135	
$\hat{\beta}_p(\text{g})$	0	0	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_p(\text{h})$	0	0	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_p(\text{i})$	0	0	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_p(\text{j})$	0	0	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_p(\text{k})$	0	0	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_p(\text{l})$	0	0	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_p(\text{m})$	0.042	0	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_p(\text{n})$	0.01	0	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_p(\text{o})$	0.81	0.993	1.033	1.047	1.059	1.059	1.059	1.059	0	1.061	0.124	
$\hat{\beta}_p(\text{p})$	0	0	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_p(\text{q})$	-0.027	0	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_p(\text{r})$	-0.063	0	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_p(\text{s:1})$	-0.15	-0.202	-0.243	0	0	0	0	0	0	0	0	0
$\hat{\beta}_p(\text{s:2})$	0.116	0.161	0.197	0	0	0	0	0	0	0	0	0
$\hat{\beta}_p(\text{t:1})$	0	0	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_p(\text{t:2})$	0	0	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_p(\text{u:1})$	0	0	0	0	0	0	0	0	0	0	0	0
$\hat{\beta}_p(\text{u:2})$	0	0	0	0	0	0	0	0	0	0	0	0
False Discovery Rate:	0.342	0.237	0.053	0.026	0	0	0	0	0	0	0	0
False Rejection Rate:	0	0	0	0	0	0	0	0	0	0.167		

Bold coefficients show oracle-properties: produces estimates that are the same as MLEs when the true model is known in advance).

Covariates $a-r$ are continuous; covariates $s-u$ are categorical; $\beta(\text{time:t})$ is equivalent to a $\theta(t)$ sub-model.

[†] CJSboost-EM model with m_{stop} tuned by bootstrap-validation.

^{††} Debiased CJSboost-EM model (un-regularized; $m \rightarrow \infty$) after discarding covariates with inclusion probabilities below a threshold.

[‡] MLEs when the true model is known in advance.

665 zero. However, readers should heed the warnings of Leeb & Pötscher (2008) who proved that oracle estimates
666 can be very inaccurate at low-to-medium sample sizes, especially if the true model is not sparse. In other
667 words, the maximum expected loss is unbounded. This is intuitive: just because we know the correct model,
668 does not mean we can accurately estimate its true effect.

669 4. Discussion

670 This study presents CJSboost: a type of multi-model inference technique for a class of Hidden Markov
671 Models (HMMs) known as capture-mark-recapture (CMR). I introduce the method using the Cormack-Jolly-
672 Seber model (CJS; Cormack, 1964; Jolly, 1965; Seber, 1965) for inference about the survival and abundance
673 of marked animals under conditions of imperfect detection. The contribution of this paper is to make two
674 modifications to the conventional component-wise boosting algorithm (e.g. Schmid et al., 2010) in order to
675 make boosting appropriate for serially-dependent time-series of CMR data, a.k.a. capture-histories. One
676 CJSboost method interweaves an Expectation-Maximization (EM) step between boosting iterations, and
677 the second method uses stochastic imputation of latent states. Both methods can be used to estimate
678 the *gradient of the loss function* which is the crux of statistical boosting. This paper is meant to prove
679 and motivate these modifications so that boosting can be introduced to a wider-class of CMR models,
680 such as the POPAN or PCRD or spatial capture-recapture. Code is available on the Github site [http:](http://github.com/farawayinspace/HMMboost)
681 [//github.com/farawayinspace/HMMboost](http://github.com/farawayinspace/HMMboost) as well as a tutorial.

682 In this article, I introduce CJSboost by positioning it within the general theory of model-selection and
683 multi-model inference (MMI); specifically, I show that CJSboost can be used for the two domains of multi-
684 model inference: i) efficient estimation and/or prediction, and ii) consistent model-identification a.k.a. finding
685 the hypothesis-cum-model which most support. These are what Aho et al. (2014) refers to as A-type vs.
686 B-type thinking. I show why boosting is very appealing, both theoretically and practically, for CMR practi-
687 tioners who use MMI techniques, such as AIC model-averaging or BIC model-selection.

688 Specifically, CJSboost is a type of *shrinkage estimator*: it negotiates the complexity of a model in order to
689 minimize a prediction error. This error is closely related to the Expected Log-Likelihood which Akaike used
690 to motivate his famous derivation of the AIC (Akaike, 1974, 1998). Akaike explained that model-selection
691 according to the Expected Log-Likelihood is efficient: it performs best at minimizing the square-error between
692 estimates and a true process. Through simulation, I show that boosting is qualitatively similar to AICc-
693 methods at estimating abundance, and it is much better at estimating survival. I also propose that CJSboost
694 can be coupled with a new technique called stability selection (Meinshausen & Bühlmann, 2010) in order
695 to derive a *sparse estimator*, that is, to find covariates that significantly influence survival and are part of
696 the “true model”, much like the BIC. Therefore, CMR practitioners can use the two flavours of CJSboost in
697 order to tackle both domains of MMI: efficient estimation or consistent model-identification.

698 However, CJSboost has many other advantages over AIC/BIC model-selection and their constituent
699 fixed-effect models:

- 700 • it can automatically perform variable-selection and explore higher-order interactions, even in situations
701 of low-sample size (i.e., the $n < p$ problem);
- 702 • it can include non-linear effects such as splines, regression trees, spatial kernels, or any of the base-
703 learners available in the `mboost` family of R packages (Bühlmann & Hothorn, 2007; Hothorn et al.,
704 2006; Mayr et al., 2012; Hofner et al., 2012);
- 705 • it has shrinkage of estimates away from extreme values and inadmissible values (e.g., $\hat{\phi} = 1$) and avoids
706 parameter singularities;
- 707 • its shrinkage properties can handle parameter non-identifiability issues better than the use of arbitrary
708 constraints (e.g., fixing $\phi_T = \phi_{T-1}$);
- 709 • it can better cope with multi-collinearity;

710 There are, however, many disadvantages and challenges to CJSboost. Some challenges are technical
711 and require further research, such as theoretical validation of the consistency of stability selection. Other
712 challenges are conceptual and will require practitioners to embrace new ideas and re-think old habits (such
713 as reliance on p-values). I will briefly comment on some of the conceptual challenges first, then I will suggest
714 new lines of research to address some technical challenges and useful extensions.

715 *4.1. Conceptual challenges*

716 Component-wise boosting is related to many important statistical ideas (Meir & Rätsch, 2003). It is
717 similar to the Lasso solution (Efron et al., 2004; Bühlmann & Hothorn, 2007), which is favoured in machine
718 learning. It is a type of model-averaging (Hand & Vinciotti, 2003) by weighting the outputs of hundreds
719 or thousands of sub-models. It is also a Generalized Additive Model which is itself a type of penalized
720 regression approach (Mayr et al., 2012). Despite these connections with other popular techniques, the eco-
721 logical community has been slow to adopt statistical boosting. I believe this may be due to a few conceptual
722 misunderstandings, such as shrinkage and suspicion of algorithmic learning techniques.

723 *Algorithmic Inference.* Boosting originally arose as a purely algorithmic means of classification (Meir &
724 Rätsch, 2003; Mayr et al., 2014). Some ecologists have embraced such methods (Elith et al., 2008), but
725 I suspect many are sceptical of machine-learning methods in favour of parametric Maximum Likelihood
726 Estimation (MLE), especially given the long-studied optimality properties of the latter. Part of the motivation
727 of this article was to review some theory about model-selection, such as shrinkage and Akaike's AIC, and
728 show why they lend support to component-wise boosting for statistical inference. Namely, we now know that
729 at finite sample sizes, the MLE solution of a multiple-regression problem is inadmissible because of shrinkage
730 (*sensu* Copas, 1983, 1997). Secondly, Akaike (1974) showed us that the Expected Log-likelihood, rather than
731 the Maximum Likelihood, is efficient at deciding the optimal complexity of a model. Therefore, there is solid
732 theory to support the statistical utility of CJSboosting for CMR analysis, given that it is a type of shrinkage
733 estimator and it approximates the Expected (negative) log-Likelihood.

734 *Shrinkage*. Despite a huge body of research about shrinkage (Stein, 1956; James & Stein, 1961; Copas,
735 1983, 1997; Royle & Link, 2002), shrinkage creates a conceptual discomfort for ecologists, and this may be
736 boosting’s greatest hurdle. First, we must do away with familiar tools like p-values and confidence intervals
737 (more below). More importantly, we must grapple with the red-herring of unbiased-ness, to which most
738 practical ecologists seem to consider sacrosanct. Ecologists trained to scrutinize diagnostic residual-plots
739 may look at the bias in Figure 2 and be very alarmed, despite the underlying loss-optimality. In other words,
740 we incur some bias to minimize an expected square-error loss (see Appendix E). This made shrinkage highly
741 controversial 50 years ago at the time of its discovery (Efron & Morris, 1975), and its repercussions have not
742 fully permeated the non-statistician research community.

743 *Bayesian Interpretation*. However, the rising popularity of Bayesianism may be the greatest advocate for
744 component-wise boosting. First, ℓ_1 -regularizers, such as the Lasso and component-wise boosting, have a
745 Bayesian interpretation (Geman et al., 1992; Hooten & Hobbs, 2015), and the outputs are merely a type of
746 a Maximum A Posteriori (MAP) estimate (Murphy, 2012c). Secondly, ecological practitioners seem uncon-
747 cerned with the fact that Bayesians are technically biased due to the role of priors at finite sample sizes.
748 To wit, Bayesians have become popular champions of shrinkage, to the extent that it almost seems like a
749 Bayesian idea, despite its Frequentist origins. For example, Royle & Link (2002) advocated for Hierarchical
750 Bayesian random-effect models for CMR primarily because of the benefits of shrinkage. CJSboost is the
751 Frequentist answer to their work.

752 *4.2. Inference without Confidence Intervals or P-values*

753 In this paper, I have chosen not to show 95%CI nor classical p-values for marginal effects’ null-hypothesis
754 tests. I ignore these in order to focus the reader’s attention on point-wise estimation: the type of inference
755 that shrinkage and AIC-like estimators were specifically developed for and should do optimally. For example,
756 if one desires a time-series of abundance, then boosting or AIC-methods should produce estimates that
757 generally have the lowest mean square-error loss between truth and estimate, i.e. the point-estimates are
758 as close as possible to the truth, over all possible samples from the population. This type of inference
759 does not depend on significant effect sizes or 95%CI; estimation variance is directly incorporated into the
760 procedure through shrinkage (Appendix E). That being said, it is common in the boosting literature to use
761 bootstrapping to approximate CI, and this could be done in CJSboost by bootstrapping capture-histories.

762 However, I would urge practitioners to think carefully about why they wish to have p-values or CI, rather
763 than consider them as default statistics. There is growing concern about the misuse of p-values (Anderson
764 et al., 2000; Gerrodette, 2011) and CI (Hoekstra et al., 2014), and some journals have started banning them
765 altogether (Trafimow & Marks, 2015). I suggest that there are alternative tools which are more aligned with
766 one’s research goals. For example, if a practitioner is interested in using 95%CI or classic p-values to test
767 whether a covariate is “significantly” different from zero, then perhaps the real intention is to discover which
768 covariates are truly influential? For this type of model-identification inference (what Aho et al., 2014, called

769 B-type thinking), I propose the use of stability selection and approximate posterior inclusion probabilities.
770 Similarly, one may wish to cap their False Discoveries (Meinshausen & Bühlmann, 2010; Shah & Samworth,
771 2013). This is a closer marriage of research goals and statistical analysis.

772 Finally, I would also remind readers that the abandonment of CIs or p-values is not a unique deficiency
773 to CJSboost, but is true for all model-selection or shrinkage estimators. The common practice of doing
774 model-selection and then using the CIs or classic p-values from the best model, as if model-selection was
775 never performed, is invalid. Breiman (1992) called this a “Quiet Scandal”. The sampling properties of a
776 post-model-selection estimator can be significantly different from those of a single-model (Leeb & Pötscher,
777 2005). This is the price of multi-model inference vs. declaring a true model *a priori*. Therefore, one’s only
778 recourse in MMI is to use model-averaged CIs (Anderson et al., 2000) or bootstrap-approximated CIs, or
779 multi-model p-values (Lee & Boone, 2011) or, better yet, to calculate statistics which actually address one’s
780 research question.

781 4.3. Extensions and Future Considerations

782 This study is merely the first step in developing and introducing boosting for CMR models. A lot of the
783 theory of loss-efficiency and consistency in univariate boosting for will need further validation in the HMM
784 context.

785 *Estimation.* Regarding estimation performance, the simulations showed that CJSboost is very competitive,
786 and perhaps better, than AICc averaging or model-selection at estimating survival and abundance. However,
787 it is unknown whether CJSboost shares any of the theoretical efficiency properties of its univariate version.
788 For example: does it obtain the minimal worst-case error, i.e. is it minimax optimal (Bühlmann & Yu, 2003)?
789 How sensitive is its performance to the regularization parameters? Of more practical concern, the new basis
790 functions of `mboost` create new ways to address old CMR estimation challenges, such as random-effect base-
791 learners to accommodate individual heterogeneity, or CART for automatic discovery of non-linear processes.
792 These opportunities require further empirical study, such as whether they incur significant estimation trade-
793 offs. For example, Bühlmann & Yu (2003) found worse estimation performance with CART-like learners vs.
794 least-square learners in simple linear regression models.

795 *Consistency.* Regarding variable selection or hypothesis-testing, this type of inference has been much less
796 important in CMR than estimating abundance. However, I expect that it will become more important
797 in certain “Big Data” domains where interest lies in finding significant associations between demographic
798 variation and environmental covariates. For such inferences, the key property that a researcher needs is
799 model-selection consistency: she desires a procedure that can recover the true model with high-probability.
800 This type of MMI is prone to False Discoveries, especially when practitioners use prediction-optimal methods,
801 such as the AIC/c or its derivatives (Shao, 1993; Yang, 2005). This misuse is widespread in ecology, and may
802 contribute to the current crisis of reproducibility (Galipaud et al., 2014). For consistent variable selection,

803 boosting has many potential extensions, such as TwinBoosting (Bühlmann & Hothorn, 2010). I suggest
804 enhancing CJSboost with stability selection to approximate Bayesian inclusion probabilities.

805 *Stability Selection.* This is an exciting and growing field of study, and the stability-selection-enhanced CJS-
806 boost technique may need revision in the near future. In particular, the univariate versions of stability
807 selection have theoretical bounds on the number of False Discoveries (Meinshausen & Bühlmann, 2010; Shah
808 & Samworth, 2013) and selection probabilities of spurious variables (Bach, 2008), but these do not apply to
809 multi-parameter boosting. Secondly, it is unclear whether we must marginalize over all three regularization
810 parameters (m and ν_p and ν_ϕ) or whether we can, as I have suggested, focus only on m . Third, it is un-
811 clear whether there is a violation of the assumption “exchangeability” of spurious covariates, as may be the
812 case with the time-varying covariates vs. individually-varying covariates, as suggested in the simulations.
813 These will require more empirical study. The latter may be partially solved by using the less-restrictive
814 complementary-pairs stability selection of Shah & Samworth (2013). Nonetheless, the simulation results are
815 promising and in-line with other studies: that is, influential covariates are selected with a probability that
816 converges to 1 as sample sizes get large, and there is good discrimination between significant and negligible
817 covariates.

818 *Extensions.* By validating the boosting technique for a simple open-population model, this study paves the
819 way for more popular CMR models, such as POPAN and the PCRD, which have more model parameters in
820 the likelihood function, like temporary-migration processes. With more parameters, the boosting algorithms
821 will require more efficient ways of tuning regularization parameters. See Appendix C.2 for ideas in this
822 regard.

823 *New Base-learners.* One major benefit of the CJSboost framework is its extensibility. It can accommodate
824 phenomena such as individual heterogeneity, spatial capture-recapture and cyclic-splines. These are possible
825 because the CJSboost code is written for compatibility with the `mboost` family of R packages, and leverages
826 their impressive variety of base-learners (Bühlmann & Hothorn, 2007; Hofner et al., 2012). For example, the
827 `brandom` base-learner can accommodate individual random effects for addressing individual heterogeneity in
828 a manner similar to Bayesian Hierarchical models (Rankin et al., 2016). Kernels (`brad`) and spatial splines
829 (`bspatial`) can be used for smooth spatial effects (Kneib et al., 2009; Hothorn et al., 2010; Tyne et al., 2015)
830 offering an entirely new framework for spatial capture-recapture. The largest advantage is that users can add
831 these extensions via the R formula interface, rather than having to modify deep-level code.

832 5. Conclusions

- 833 1. Boosting is a shrinkage estimator and regularization algorithm that can be adapted to capture-mark-
834 recapture through an additional Expectation-Maximization step that imputes latent states.

- 835 2. Boosting negotiates the “bias-variance trade-off” by incurring a slight bias in all coefficients, but yields
836 estimates that are more stable to outliers and over-fitting, across multiple realizations of the data
837 (Appendix E).
- 838 3. CJSboost allows for powerful learners, such as recursive-partitioning trees (e.g., CART) for automatic
839 variable-selection, interaction detection, and non-linearity. This flexibility seems to come at the cost of
840 slightly more conservative estimates (if the underlying true model is linear).
- 841 4. Both AICc model-selection and boosting are motivated by good predictive performance: minimizing
842 an expected loss (a.k.a. risk, or generalization error). When using least-squares or CART-like base-
843 learners, the estimates from CJSboost are qualitatively similar to AICc model-averaging, but with more
844 shrinkage on coefficients.
- 845 5. CJSboost seems to perform very well in high-dimensional model-selection problems, with the ability to
846 recover a small set of influential covariates.
- 847 6. If the goal of a CMR analysis is to not estimate abundance or survival, but to find significant covariates,
848 then CJSboosted models can be enhanced with stability-selection to derive a model-selection consistent
849 estimator. Further research is necessary to validate the consistency property.

850 6. Acknowledgements

851 I would like to thank Professor Sayan Mukherjee for giving this project an initial “thumbs up” during a
852 Duke University course on Probabilistic Machine Learning. I would also like to thank David Anderson, Lars
853 Bejder, Krista Nicholson and Julian Tyne for helpful comments and critiques which greatly strengthened this
854 manuscript.

855 7. Works Cited

- 856 Aho, K., Derryberry, D., & Peterson, T. (2014). Model selection for ecologists: The worldviews of AIC and
857 BIC. *Ecology*, *95*, 631–636. doi:10.1890/13-1452.1.
- 858 Akaike, H. (1974). A new look at the statistical model identification. *Automatic Control, IEEE Transactions*
859 *on*, *19*, 716 – 723. doi:10.1109/TAC.1974.1100705.
- 860 Akaike, H. (1998). Information Theory and an Extension of the Maximum Likelihood Principle. In E. Parzen,
861 K. Tanabe, & G. Kitagawa (Eds.), *Selected Papers of Hirotugu Akaike* Springer Series in Statistics (pp. 199–
862 213). Springer New York. URL: http://link.springer.com/chapter/10.1007/978-1-4612-1694-0_15.
863 doi:10.1007/978-1-4612-1694-0_15.
- 864 Anderson, D. R., Burnham, K. P., & Thompson, W. L. (2000). Null hypothesis testing: Problems, prevalence,
865 and an alternative. *Journal of Wildlife Management*, *64*, 912–923. doi:10.2307/3803199.

- 866 Bach, F. R. (2008). Bolasso: Model consistent Lasso estimation through the bootstrap. In *Proceedings of the*
867 *25th International Conference on Machine Learning ICML '08* (pp. 33–40). New York, NY, USA: ACM.
868 doi:10.1145/1390156.1390161.
- 869 Breiman, L. (1992). The little bootstrap and other methods for dimensionality selection in regression: X-fixed
870 prediction error. *Journal of the American Statistical Association*, *87*, 738–754. doi:10.1080/01621459.
871 1992.10475276.
- 872 Breiman, L. (1998). Arcing classifier (with discussion and a rejoinder by the author). *The Annals of Statistics*,
873 *26*, 801–849. doi:10.1214/aos/1024691079.
- 874 Breiman, L. (1999). Prediction games and Arcing algorithms. *Neural Computation*, *11*, 1493–1517. doi:10.
875 1162/089976699300016106.
- 876 Bühlmann, P., & Hothorn, T. (2007). Boosting algorithms: Regularization, prediction and model fitting.
877 *Statistical Science*, *22*, 477–505. doi:10.1214/07-STS242.
- 878 Bühlmann, P., & Hothorn, T. (2010). Twin Boosting: Improved feature selection and prediction. *Statistics*
879 *and Computing*, *20*, 119–138. doi:10.1007/s11222-009-9148-5.
- 880 Bühlmann, P., & Yu, B. (2003). Boosting with the L2 loss: Regression and classification. *Journal of the*
881 *American Statistical Association*, *98*, 324–339. doi:10.1198/016214503000125.
- 882 Burnham, K. P., & Anderson, D. R. (2004). Multimodel inference: Understanding AIC and BIC in model
883 selection. *Sociological Methods & Research*, *33*, 261–304. doi:10.1177/0049124104268644.
- 884 Burnham, K. P., Anderson, D. R., & Huyvaert, K. P. (2011). AIC model selection and multimodel inference in
885 behavioral ecology: Some background, observations, and comparisons. *Behavioral Ecology and Sociobiology*,
886 *65*, 23–35. doi:10.1007/s00265-010-1029-6.
- 887 Cade, B. S. (2015). Model averaging and muddled multimodel inferences. *Ecology*, *96*, 2370–2382. doi:10.
888 1890/14-1639.1.
- 889 Copas, J. B. (1983). Regression, prediction and shrinkage. *Journal of the Royal Statistical Society. Series B*
890 *(Methodological)*, *45*, 311 – 354.
- 891 Copas, J. B. (1997). Using regression models for prediction: Shrinkage and regression to the mean. *Statistical*
892 *Methods in Medical Research*, *6*, 167–183. doi:10.1177/096228029700600206.
- 893 Cormack, R. (1964). Estimates of survival from the sighting of marked animals. *Biometrika*, *51*, 429–438.
894 doi:10.1093/biomet/51.3-4.429.
- 895 Dietterich, T. G., Ashenfelder, A., & Bulatov, Y. (2004). Training Conditional Random Fields via Gradient
896 Tree Boosting. In *Proceedings of the Twenty-First International Conference on Machine Learning ICML*
897 '04 (p. 28). New York, NY, USA: ACM. doi:10.1145/1015330.1015428.

- 898 Draper, D. (2010). Discussion: Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *72*, 461–462. doi:10.1111/j.1467-9868.2010.00740.x.
- 900 Efron, B., Hastie, T., Johnstone, I., & Tibshirani, R. (2004). Least angle regression. *The Annals of Statistics*,
901 *32*, 407–499. doi:10.1214/009053604000000067.
- 902 Efron, B., & Morris, C. (1975). Data analysis using Stein’s estimator and its generalizations. *Journal of the*
903 *American Statistical Association*, *70*, 311. doi:10.2307/2285814.
- 904 Elith, J., Leathwick, J. R., & Hastie, T. (2008). A working guide to boosted regression trees. *The Journal*
905 *of Animal Ecology*, *77*, 802–813. doi:10.1111/j.1365-2656.2008.01390.x.
- 906 van Erven, T., Grünwald, P., & de Rooij, S. (2012). Catching up faster by switching sooner: A predictive ap-
907 proach to adaptive estimation with an application to the AIC–BIC dilemma. *Journal of the Royal Statistical*
908 *Society: Series B (Statistical Methodology)*, *74*, 361–417. doi:10.1111/j.1467-9868.2011.01025.x.
- 909 Fan, J., & Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties.
910 *Journal of the American Statistical Association*, *96*, 1348–1360. doi:10.1198/016214501753382273.
- 911 Friedman, J., Hastie, T., & Tibshirani, R. (2000). Additive logistic regression: A statistical view of boosting
912 (with discussion). *The Annals of Statistics*, *28*, 337–374. doi:10.1214/aos/1016218223.
- 913 Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*,
914 *29*, 1189–1232.
- 915 Galipaud, M., Gillingham, M. A. F., David, M., & Dechaume-Moncharmont, F.-X. (2014). Ecologists over-
916 estimate the importance of predictor variables in model averaging: A plea for cautious interpretations.
917 *Methods in Ecology and Evolution*, *5*, 983–991. doi:10.1111/2041-210X.12251.
- 918 Geman, S., Bienenstock, E., & Doursat, R. (1992). Neural Networks and the Bias/Variance Dilemma. *Neural*
919 *Computation*, *4*, 1–58. doi:10.1162/neco.1992.4.1.1.
- 920 Gerrodette, T. (2011). Inference without significance: Measuring support for hypotheses rather than rejecting
921 them. *Marine Ecology*, *32*, 404–418. doi:10.1111/j.1439-0485.2011.00466.x.
- 922 Groll, A., & Tutz, G. (2012). Regularization for Generalized Additive Mixed Models by Likelihood-based
923 Boosting. *Methods of Information in Medicine*, *51*, 168–177. doi:10.3414/ME11-02-0021.
- 924 Hand, D. J., & Vinciotti, V. (2003). Local versus global models for classification problems: Fitting models
925 where it matters. *The American Statistician*, *57*, 124–131. doi:10.1198/0003130031423.
- 926 Hoekstra, R., Morey, R. D., Rouder, J. N., & Wagenmakers, E.-J. (2014). Robust misinterpretation of
927 confidence intervals. *Psychonomic Bulletin & Review*, *21*, 1157–1164. doi:10.3758/s13423-013-0572-3.

- 928 Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems.
929 *Technometrics*, *12*, 55–67.
- 930 Hofner, B., Boccuto, L., & Göker, M. (2015). Controlling false discoveries in high-dimensional situations:
931 Boosting with stability selection. *BMC Bioinformatics*, *16*. doi:10.1186/s12859-015-0575-3.
- 932 Hofner, B., Kneib, T., & Hothorn, T. (2014). A unified framework of constrained regression. *Statistics and
933 Computing*, *26*, 1–14. doi:10.1007/s11222-014-9520-y.
- 934 Hofner, B., Mayr, A., Robinzonov, N., & Schmid, M. (2012). *Model-Based Boosting in R: A Hands-on
935 Tutorial Using the R Package Mboost*. Technical Report 120 Department of Statistics, Ludwig-Maximilians-
936 Universität Munich. URL: <http://epub.ub.uni-muenchen.de/12754/>.
- 937 Hooten, M., & Hobbs, N. T. (2015). A guide to Bayesian model selection for ecologists. *Ecological Monographs*,
938 *85*, 3–28. doi:10.1890/14-0661.1.
- 939 Hothorn, T., Hornik, K., & Zeileis, A. (2006). Unbiased recursive partitioning: A conditional inference frame-
940 work. *Journal of Computational and Graphical Statistics*, *15*, 651–674. doi:10.1198/106186006X133933.
- 941 Hothorn, T., Müller, J., Schröder, B., Kneib, T., & Brandl, R. (2010). Decomposing environmental, spatial,
942 and spatiotemporal components of species distributions. *Ecological Monographs*, *81*, 329–347. doi:10.
943 1890/10-0602.1.
- 944 Hutchinson, R., Liu, L., & Dietterich, T. (2011). Incorporating boosted regression trees into ecological latent
945 variable models. In W. Burgard, & D. Roth (Eds.), *Proceedings of the Twenty-Fifth AAAI Conference on
946 Artificial Intelligence* (pp. 1343–1348). Association for the Advancement of Artificial Intelligence. URL:
947 <http://www.aaai.org/ocs/index.php/AAAI/AAAI11/paper/viewFile/3711/4086>.
- 948 James, W., & Stein, C. (1961). Estimation with Quadratic Loss. In *Proceedings of the Fourth Berkeley
949 Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*
950 (pp. 361–379). Berkeley, CA, USA: University of California Press. URL: [http://projecteuclid.org/
951 euclid.bsm/1200512173](http://projecteuclid.org/euclid.bsm/1200512173).
- 952 Jolly, G. M. (1965). Explicit estimates from capture-recapture data with both death and immigration-
953 stochastic model. *Biometrika*, *52*, 225–247. doi:10.1093/biomet/52.1-2.225.
- 954 Kearns, M., & Valiant, L. (1994). Cryptographic limitations on learning Boolean formulae and finite au-
955 tomata. *Journal of the ACM*, *41*, 67–95. doi:10.1145/174644.174647.
- 956 Kneib, T., Hothorn, T., & Tutz, G. (2009). Variable selection and model choice in geoadditive regression
957 models. *Biometrics*, *65*, 626–634. doi:10.1111/j.1541-0420.2008.01112.x.

- 958 Laake, J. L. (2013). *RMark: An R Interface for Analysis of Capture-Recapture Data with MARK*. AFSC
959 Processed Report 2013-01 Alaska Fisheries Science Center, NOAA, National Marine Fisheries Service
960 Seattle, WA, USA. URL: <http://www.afsc.noaa.gov/Publications/ProcRpt/PR2013-01.pdf>.
- 961 Lebreton, J.-D., Burnham, K. P., Clobert, J., & Anderson, D. R. (1992). Modeling survival and testing
962 biological hypotheses using marked animals: A unified approach with case studies. *Ecological Monographs*,
963 *62*, 67–118. doi:10.2307/2937171.
- 964 Lee, A. H., & Boone, E. L. (2011). A frequentist assessment of Bayesian inclusion probabilities for
965 screening predictors. *Journal of Statistical Computation and Simulation*, *81*, 1111–1119. doi:10.1080/
966 00949651003702135.
- 967 Leeb, H., & Pötscher, B. M. (2005). Model selection and inference: Facts and fiction. *Econometric Theory*,
968 *21*, 21–59. doi:10.1017/S0266466605050036.
- 969 Leeb, H., & Pötscher, B. M. (2008). Sparse estimators and the oracle property, or the return of Hodges’
970 estimator. *Journal of Econometrics*, *142*, 201–211. doi:10.1016/j.jeconom.2007.05.017.
- 971 Leeb, H., & Pötscher, B. M. (2009). Model selection. In T. Mikosch, J.-P. Kreiß, R. A. Davis, & T. G.
972 Andersen (Eds.), *Handbook of Financial Time Series* (pp. 889–925). Berlin Heidelberg: Springer. URL:
973 <http://dx.doi.org/10.1007/978-3-540-71297-8>.
- 974 Link, W. A., & Barker, R. J. (2006). Model weights and the foundations of multimodel inference. *Ecology*,
975 *87*, 2626–2635. doi:10.1890/0012-9658(2006)87[2626:MWATF0]2.0.CO;2.
- 976 Mayr, A., Binder, H., Gefeller, O., & Schmid, M. (2014). The evolution of boosting algorithms: From
977 machine learning to statistical modelling. *Methods of Information in Medicine*, *53*, 419–427. doi:10.3414/
978 ME13-01-0122.
- 979 Mayr, A., Fenske, N., Hofner, B., Kneib, T., & Schmid, M. (2012). Generalized additive models for location,
980 scale and shape for high dimensional data—a flexible approach based on boosting. *Journal of the Royal*
981 *Statistical Society: Series C (Applied Statistics)*, *61*, 403–427. doi:10.1111/j.1467-9876.2011.01033.x.
- 982 McDonald, T. L., & Amstrup, S. C. (2001). Estimation of population size using open capture-recapture
983 models. *Journal of Agricultural, Biological, and Environmental Statistics*, *6*, 206–220.
- 984 McQuarrie, A. D. (1999). A small-sample correction for the Schwarz SIC model selection criterion. *Statistics*
985 *& Probability Letters*, *44*, 79–86. doi:10.1016/S0167-7152(98)00294-6.
- 986 Meinshausen, N., & Bühlmann, P. (2010). Stability selection. *Journal of the Royal Statistical Society: Series*
987 *B (Statistical Methodology)*, *72*, 417–473. doi:10.1111/j.1467-9868.2010.00740.x.
- 988 Meir, R., & Rätsch, G. (2003). An introduction to boosting and leveraging. *Lecture Notes in Computer*
989 *Science*, *2600*, 118–183. doi:10.1007/3-540-36434-X_4.

- 990 Mukherjee, S., Rifkin, R., & Poggio, T. (2003). Regression and Classification with Regularization. In
991 D. D. Denison, M. H. Hansen, C. C. Holmes, B. Mallick, & B. Yu (Eds.), *Nonlinear Estimation and*
992 *Classification* Lecture Notes in Statistics (pp. 111–128). New York, NY, USA: Springer. URL: [http://](http://catalog.hathitrust.org/api/volumes/oclc/50510726.html)
993 catalog.hathitrust.org/api/volumes/oclc/50510726.html.
- 994 Murphy, K. P. (2012a). Frequentist Statistics. In *Machine Learning: A Probabilistic Approach* Adaptive
995 Computation and Machine Learning Series (pp. 191–216). Cambridge, MA, USA: MIT Press.
- 996 Murphy, K. P. (2012b). Markov and Hidden Markov Models. In *Machine Learning: A Probabilistic Approach*
997 Adaptive Computation and Machine Learning Series (pp. 589–630). Cambridge, MA, USA: MIT Press.
- 998 Murphy, K. P. (2012c). Sparse Linear Models. In *Machine Learning: A Probabilistic Approach* Adaptive
999 Computation and Machine Learning Series (pp. 421–478). Cambridge, MA, USA: MIT Press.
- 1000 Nikolay Robinzonov (2013). *Advances in Boosting of Temporal and Spatial Models*. Doctoral thesis LMU
1001 München: Fakultät für Mathematik, Informatik und Statistik Munich. URL: [http://nbn-resolving.de/](http://nbn-resolving.de/urn:nbn:de:bvb:19-153382)
1002 [urn:nbn:de:bvb:19-153382](http://nbn-resolving.de/urn:nbn:de:bvb:19-153382).
- 1003 Otis, D. L., Burnham, K. P., White, G. C., & Anderson, D. R. (1978). Statistical inference from capture
1004 data on closed animal populations. *Wildlife Monographs*, (pp. 3–135).
- 1005 Pérez-Jorge, S., Gomes, I., Hayes, K., Corti, G., Louzao, M., Genovart, M., & Oro, D. (2016). Effects
1006 of nature-based tourism and environmental drivers on the demography of a small dolphin population.
1007 *Biological Conservation*, 197, 200–208. doi:10.1016/j.biocon.2016.03.006.
- 1008 R Core Team (2016). R: A language and environment for statistical computing. URL: [http://www.](http://www.r-project.org)
1009 [r-project.org](http://www.r-project.org).
- 1010 Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition.
1011 *Proceedings of the IEEE*, 77, 257–286. doi:10.1109/5.18626.
- 1012 Rankin, R. W., Nicholson, K. E., Allen, S. J., Krützen, M., Bejder, L., & Pollock, K. H. (2016). A full-capture
1013 Hierarchical Bayesian model of Pollock’s Closed Robust Design and application to dolphins. *Frontiers in*
1014 *Marine Science*, 3. doi:10.3389/fmars.2016.00025.
- 1015 Richardson, S. (2010). Discussion: Stability selection. *Journal of the Royal Statistical Society: Series B*
1016 *(Statistical Methodology)*, 72, 448–451. doi:10.1111/j.1467-9868.2010.00740.x.
- 1017 Royle, J. A., & Link, W. A. (2002). Random effects and shrinkage estimation in capture-recapture models.
1018 *Journal of Applied Statistics*, 29, 329–351. doi:10.1080/02664760120108746.
- 1019 Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning*, 5, 197–227. doi:10.1023/A:
1020 1022648800760.

- 1021 Schmid, M., & Hothorn, T. (2008a). Boosting additive models using component-wise P-Splines. *Computa-*
1022 *tional Statistics & Data Analysis*, *53*, 298–311. doi:10.1016/j.cjsda.2008.09.009.
- 1023 Schmid, M., & Hothorn, T. (2008b). Flexible boosting of accelerated failure time models. *BMC Bioinform-*
1024 *matics*, *9*, 269. doi:10.1186/1471-2105-9-269.
- 1025 Schmid, M., Hothorn, T., Krause, F., & Rabe, C. (2012). A PAUC-based estimation technique for disease
1026 classification and biomarker selection. *Statistical Applications in Genetics and Molecular Biology*, *11*.
1027 doi:10.1515/1544-6115.1792.
- 1028 Schmid, M., Potapov, S., Pfahlberg, A., & Hothorn, T. (2010). Estimation and regularization techniques
1029 for regression models with multidimensional prediction functions. *Statistics and Computing*, *20*, 139–150.
1030 doi:10.1007/s11222-009-9162-7.
- 1031 Schmid, M., Wickler, F., Maloney, K. O., Mitchell, R., Fenske, N., & Mayr, A. (2013). Boosted Beta
1032 Regression. *PLoS ONE*, *8*. doi:10.1371/journal.pone.0061623.
- 1033 Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, *6*, 461–464. doi:10.
1034 1214/aos/1176344136.
- 1035 Seber, G. A. F. (1965). A note on the multiple recapture census. *Biometrika*, *52*, 249–259. doi:10.2307/
1036 2333827.
- 1037 Shah, R. D., & Samworth, R. J. (2013). Variable selection with error control: Another look at stability
1038 selection: *Another Look at Stability Selection*. *Journal of the Royal Statistical Society: Series B (Statistical*
1039 *Methodology)*, *75*, 55–80. doi:10.1111/j.1467-9868.2011.01034.x.
- 1040 Shao, J. (1993). Linear Model selection by cross-validation. *Journal of the American Statistical Association*,
1041 *88*, 486–494. doi:10.1080/01621459.1993.10476299.
- 1042 Shao, J. (1997). An asymptotic theory for linear model selection. *Statistica Sinica*, *7*, 221–242.
- 1043 Shibata, R. (1980). Asymptotically efficient selection of the order of the model for estimating parameters of
1044 a linear process. *The Annals of Statistics*, *8*, 147–164. doi:10.1214/aos/1176344897.
- 1045 Shibata, R. (1986a). Consistency of model selection and parameter estimation. *Journal of Applied Probability*,
1046 *23*, 127 – 141. doi:10.2307/3214348.
- 1047 Shibata, R. (1986b). Selection of the number of regression variables; A minimax choice of generalized FPE.
1048 *Annals of the Institute of Statistical Mathematics*, *38*, 459–474. doi:10.1007/BF02482533.
- 1049 Stein, C. (1956). Inadmissibility of the Usual Estimator for the Mean of a Multivariate Normal Distribution.
1050 In *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability, Volume 1:*
1051 *Contributions to the Theory of Statistics* (pp. 197–206). Berkeley, CA, USA: University of California
1052 Press. URL: <http://projecteuclid.org/euclid.bsmsp/1200501656>.

- 1053 Stone, M. (1977). An asymptotic equivalence of choice of model by cross-validation and Akaike's criterion.
1054 *Journal of the Royal Statistical Society. Series B (Methodological)*, 39, 44–47.
- 1055 Sun, W., Wang, J., & Fang, Y. (2013). Consistent selection of tuning parameters via variable selection
1056 stability. *Journal of Machine Learning Research*, 14, 3419–3440.
- 1057 Taylor, A. R., Schacke, J. H., Speakman, T. R., Castleberry, S. B., & Chandler, R. B. (2016). Factors related
1058 to common bottlenose dolphin (*Tursiops truncatus*) seasonal migration along South Carolina and Georgia
1059 coasts, USA. *Animal Migration*, 3. doi:10.1515/ami-2016-0002.
- 1060 Tibshirani, R. (2011). Regression shrinkage and selection via the lasso: A retrospective: Regression Shrinkage
1061 and Selection via the Lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*,
1062 73, 273–282. doi:10.1111/j.1467-9868.2011.00771.x.
- 1063 Trafimow, D., & Marks, M. (2015). Editorial. *Basic and Applied Social Psychology*, 37, 1–2. doi:10.1080/
1064 01973533.2015.1012991.
- 1065 Tyne, J. A., Johnston, D. W., Rankin, R., Loneragan, N. R., & Bejder, L. (2015). The importance of spinner
1066 dolphin (*Stenella longirostris*) resting habitat: Implications for management. *Journal of Applied Ecology*,
1067 52, 621–630. doi:10.1111/1365-2664.12434.
- 1068 Vrieze, S. I. (2012). Model selection and psychological theory: A discussion of the differences between the
1069 Akaike Information criterion (AIC) and the Bayesian Information criterion (BIC). *Psychological Methods*,
1070 17, 228–243. doi:10.1037/a0027127.
- 1071 Ward, G., Hastie, T., Barry, S., Elith, J., & Leathwick, J. R. (2009). Presence-only data and the EM
1072 algorithm. *Biometrics*, 65, 554–563. doi:10.1111/j.1541-0420.2008.01116.x.
- 1073 White, G. C., & Burnham, K. P. (1999). Program MARK: Survival estimation from populations of marked
1074 animals. *Bird Study*, 46, S120–S139. doi:10.1080/00063659909477239.
- 1075 Yang, Y. (2005). Can the strengths of AIC and BIC be shared? A conflict between model identification
1076 and regression estimation. *Biometrika*, 92, 937–950. doi:10.1093/biomet/92.4.937.
- 1077 Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*,
1078 101, 1418–1429. doi:10.1198/016214506000000735.

1079 APPENDICES

1080 Appendix A. The CJSboost algorithm for Monte-Carlo approximation

1081 The second strategy to boost a CJS capture-recapture model is called CJSboost Monte Carlo (MC). The
1082 idea is to garner conditional independence of data-points (y_j, \mathbf{x}_j) by integrating over the distributions of
1083 latent states $\pi(\mathbf{z}_i | \mathbf{y}_i, \mathbf{F}_i)$. The integration is approximated with a large sample from the posterior of \mathbf{z}_i . A

1084 fast and simple “forward-filtering and backward-sampling” algorithm is used to sample latent states (Rabiner,
1085 1989; Murphy, 2012b), detailed in Appendix B.4.

1086 Within each boosting iteration m , we sample S sequences of \mathbf{z}_i . Per s sequence, we estimate a separate
1087 negative-gradient, and fit base-learners to it. After fitting all S samples, we update the prediction vectors
1088 with the best-fitting base-learners from each sequence, $F_\theta^{(m+1)} = F_\theta^{(m)} + \nu_\theta \sum_s \hat{f}_k^{(s)}$. Over $S \times m$ draws, this
1089 is approximately equivalent to the EM algorithm. For comparable results to CJSboost-EM, the learning-rate
1090 parameters ν_{MC} should be set equal to $\frac{1}{S} \nu_{EM}$, i.e., the contribution of any one sequence $\mathbf{z}^{(s)}$ is small.

1091 I now describe the CJSboost-MC algorithm:

- 1092 1. Set regularization parameters S , m_{stop} , ν_ϕ , and ν_p .
- 1093 2. Initialize $m = 1$ and $\hat{\mathbf{F}}^{(0)}$.
- 1094 3. For $s = 1 : S$, do:

- 1095 (a) sample latent state sequence $\mathbf{z}_i^{(s)} \sim \pi(z|\mathbf{y}_i, \hat{\mathbf{F}}_i)$ (see Appendix B.4);
- (b) estimate the negative gradients, conditional on $\mathbf{z}_i^{(s)}$:

$$\hat{u}_{\phi,i,t}^{(m,s)} = -\frac{\partial \ell_{i,t}}{\partial F_\phi^{(m-1)}} = \frac{\mathbf{1}[z_{i,t-1}^{(s)} = 1, z_{i,t}^{(s)} = 1] - \mathbf{1}[z_{i,t-1}^{(s)} = 1, z_{i,t}^{(s)} = 0] \cdot e^{\hat{F}_{\phi,i,t}^{(m-1)}}}{1 + e^{\hat{F}_{\phi,i,t}^{(m-1)}}}$$

$$\hat{u}_{p,i,t}^{(m,s)} = -\frac{\partial \ell_{i,t}}{\partial F_p^{(m-1)}} = \frac{\mathbf{1}[z_{i,t-1}^{(s)} = 1, z_{i,t}^{(s)} = 1] \left((1 + e^{\hat{F}_{p,i,t}^{(m-1)}}) y_{i,t} - e^{\hat{F}_{p,i,t}^{(m-1)}} \right)}{1 + e^{\hat{F}_{p,i,t}^{(m-1)}}}$$

- 1096 (c) for each θ in $\{\phi, p\}$ do:

- 1097 i. for each k base-learner in θ do:
 - 1098 A. fit the base-learner to the gradient: $b_k(\hat{\mathbf{u}}_\theta^{(m,s)}, X_k) \Rightarrow g_k^{(s)}$;
 - 1099 B. make an estimate of the gradient, $\hat{f}_k^{(s)} = g_k^{(s)}(X_k)$;
- 1100 ii. find the base-learner that best-fits the gradient $\tilde{k}_k^{(s)} = \underset{k}{\operatorname{argmin}} (\hat{\mathbf{u}}_\theta^{(m,s)} - \hat{f}_k^{(s)})^2$;
- 1101 iii. append the prediction function of $\tilde{k}_k^{(s)}$ to the ensemble $\mathcal{G}_\theta \leftarrow g_{\tilde{k}_k}^{(s)}$;
- 1102 4. Update the fit vectors for each $\theta \in \{\phi, p\}$, taking the sum over all S : $F_\theta^{(m)} = F_\theta^{(m-1)} + \nu_\theta \sum_s \hat{f}_k^{(s)}$.
- 1103 5. Estimate the empirical risk $L(\mathbf{Y}, \hat{\mathbf{F}}^{(m)})$, or estimate the holdout-risk on an out-of-sample subset of the
1104 data $L(\mathbf{Y}_{\text{OOS}}, \hat{\mathbf{F}}_{\text{OOS}}^{(m)})$ for cross-validation.
- 1105 6. $m = m + 1$
- 1106 7. Repeat steps 3 to 6 until $m = m_{stop}$.

1107 Just as in the CJSboost-EM algorithm, we must tune ν and m_{stop} through cross-validation or bootstrap-
1108 validation (Section 2.2.3).

1109 Notice that the two algorithms have different surrogate loss functions and negative-gradients. However,
1110 the expected loss is still the Expected negative CJS Log-Likelihood, and the empirical risk is the negative
1111 CJS log-likelihood of the observed data.

1112 Figures A.9 and A.10 compare the CJSboost-MC algorithm against the CJSboost-EM algorithm. Figure
1113 A.9 shows model estimates of capture-probability and survival for an example dataset from Simulation 1 of

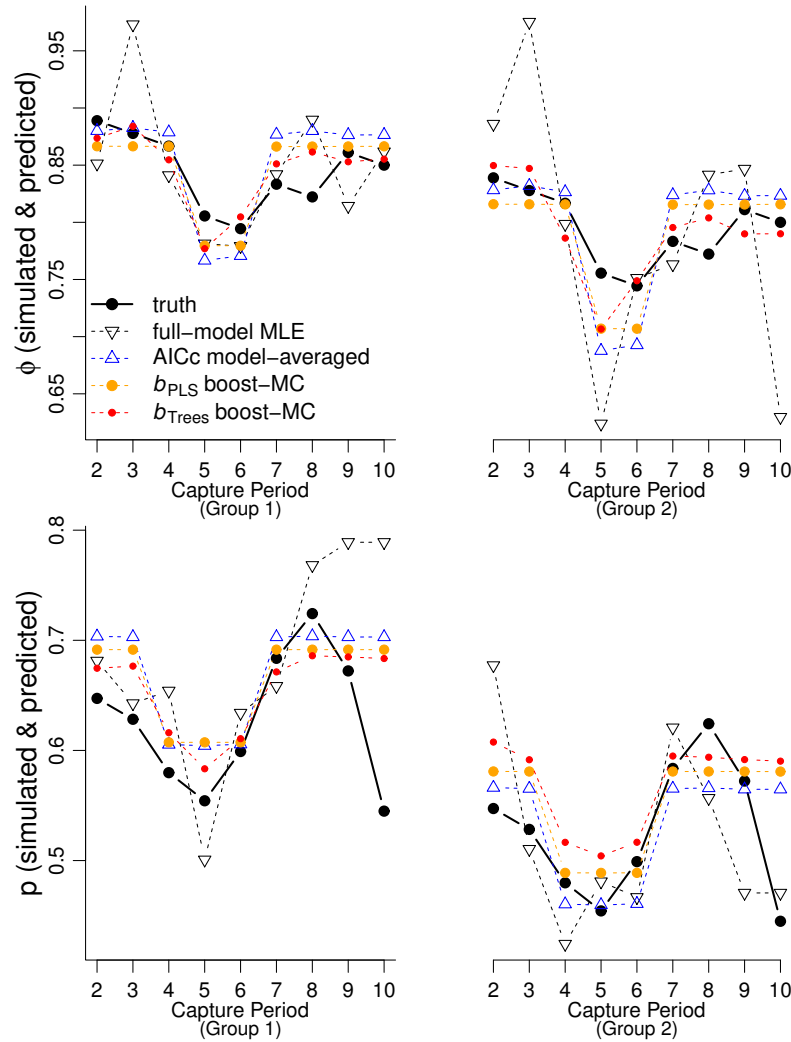


Figure A.9: From Simulation 1 of the main article, a demonstration of CJSboost estimates from the Monte-Carlo approximation technique. A comparison of capture-probability estimates $\hat{p}(t \times x)$ and survival estimates $\hat{\phi}(t \times x)$ from four models: CJSboost-MC with linear base-learners (OLS and PLS; in orange); CJSBoost-MC with non-linear base-learners (CART-like trees; in red); AICc model-averaging (blue); and MLEs of the full-model (dashed black).

1114 the main article; we see that the MC algorithm produces approximately similar estimates, although there is
 1115 some extra variation in the b_{trees} base-learners model. Figure A.10 is from the high-dimensional Simulation
 1116 3 in the main article. The Figure shows a scatter-plot of the estimates from the EM algorithm vs. the MC
 1117 algorithm, using a simulated high-dimensional dataset, where each dot is an individual i at capture-period t .
 1118 The results fall along the 1:1 line, which demonstrates that the algorithms are approximately equivalent.

1119 Appendix B. Algorithms for Filtering and Sampling HMM Latent States

1120 The CJSboost algorithms depend on conditional independence of data pairs $(y_{i,t}, X_{i,t})$ for individuals i
 1121 in capture period t , in order to estimate the negative-gradient in the descent algorithm. This is possible if
 1122 we impute information about the latent state sequences z for pairs of capture periods at t and $t-1$. The

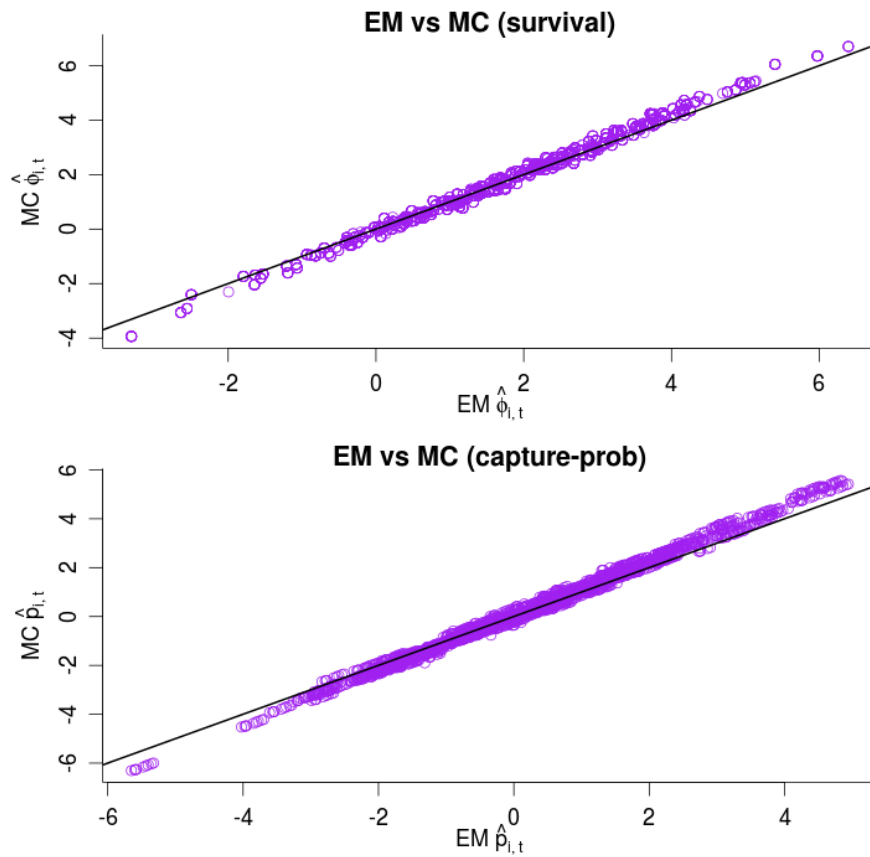


Figure A.10: Simulation 1, demonstrating CJSboost estimates from the Monte-Carlo approximation technique. A comparison of capture-probability estimates $\hat{p}(t \times \mathcal{X})$ and survival estimates $\hat{\phi}(t \times \mathcal{X})$ from models composed of linear base-learners (OLS and PLS; in orange) and non-linear base-learners (CART-like trees; in red), as well as AICc model-averaging (blue) and MLE (dashed black).

1123 two CJSboost algorithms, CJSboost-EM and CJSboost-MC, achieve this same idea with two different, but
 1124 related, techniques. In both cases, we will use a classic “forwards-backwards” messaging algorithm to gain
 1125 information about the probability distribution of the latent state sequences. In CJSboost-EM, we calculate
 1126 the *two-slice marginal probabilities* $p(z_{t-1} = u, z_t = v | \mathbf{y}_{1:T}, \phi, p)$, per boosting iteration; in CJSboost-MC, we
 1127 will *sample* \mathbf{z} from its posterior distribution $\pi(\mathbf{z}_{1:T} | \mathbf{y}_{1:T}, \phi, p)$. See Rabiner (1989) and Murphy (2012b) for
 1128 accessible tutorials.

1129 Both algorithms use a forwards-messaging algorithm and a backwards-messaging algorithm. The forwards
 1130 algorithm passes information about the state of z_t conditional on all previous observations (denoted α_t),
 1131 whereas the backwards algorithm estimates the future conditional likelihood of the capture-data given z_t at
 1132 t (denoted β_t). The α and β values are combined to make inferences about the distribution of latent states
 1133 per time t .

1134 We will drop the indices i , and focus on the capture-history of a single individual. \mathbf{y} is the time-series of
 1135 binary outcomes of length T . \mathbf{z} is a vector of latent states $z \in \{\text{dead}, \text{alive}\}$. We condition on an individual’s
 1136 first capture at time $t = t^0$, and are only concerned with the sequence $\mathbf{z}_{t^0:T}$. Survival from step $t-1$ to t is
 1137 ϕ_t . Conditional on z_t , the capture-probabilities are $p(y_t = 1 | \text{alive}) = p_t$, and $p(y_t = 1 | \text{dead}) = 0$. In HMM
 1138 notation, the CJS processes can be presented as the following column-stochastic matrices:

$$\mathbf{\Phi}_t = \begin{array}{c} \text{dead} \\ \text{alive} \end{array} \begin{array}{cc} \text{dead} & \text{alive} \\ \left(\begin{array}{cc} 1 & 1-\phi_t \\ 0 & \phi_t \end{array} \right) \end{array} \quad \mathbf{\Psi}_t = \begin{array}{c} \text{no capture} \\ \text{capture} \end{array} \begin{array}{cc} \text{dead} & \text{alive} \\ \left(\begin{array}{cc} 1 & 1-p_t \\ 0 & p_t \end{array} \right) \end{array} \quad (\text{B.1})$$

1139 In HMM parlance, $\mathbf{\Phi}$ is the Markovian transition process; we denote the probability $p(z_t = u | z_{t-1} = u)$
 1140 as $\mathbf{\Phi}_t(u, v)$. $\mathbf{\Psi}$ is the emission process representing the conditional capture-probabilities; we denote the
 1141 probability $p(y_t = 1 | z_t = v)$ as $\mathbf{\Psi}_t(v)$.

1142 *Appendix B.1. Forwards-algorithm*

1143 The forward messaging algorithm involves the recursive calculation of $\alpha_t(v)$, per time t and state $z_t = v$.
 1144 α_t is the *filtered belief state* of z_t given all the observed information in \mathbf{y} from first capture t^0 until t . Notice,
 1145 that for clarity, we drop the notation for conditioning on ϕ and p , but these are always implied.

$$\begin{aligned} a_t(v) &:= p(z_t = v | \mathbf{y}_{t^0:t}) \\ &= \frac{1}{Z_t} p(y_t | z_t = v) p(z_t = v | \mathbf{y}_{t^0:t-1}) \\ &= \frac{1}{Z_t} p(y_t | z_t = v) \sum_u p(z_t = v | z_{t-1} = u) p(z_{t-1} = u | \mathbf{y}_{t^0:t-1}) \\ &= \frac{1}{Z_t} \mathbf{\Psi}_t(v) \sum_u \mathbf{\Phi}(u, v) \alpha_{t-1}(u) \\ Z_t &= \sum_v \left(\mathbf{\Psi}_t(v) \sum_u \mathbf{\Phi}(u, v) \alpha_{t-1}(u) \right), \sum_v \alpha_t(v) = 1 \end{aligned} \quad (\text{B.2})$$

1146 The algorithm is initialized at time t^0 (an individual's first capture) with $\alpha_{t^0}(\text{alive}) = 1$ and $\alpha_{t^0}(\text{dead}) = 0$.
 1147 This is true because the animal must be alive for us to capture it. Conditional on the values of $\alpha_t(v)$ for all
 1148 v , one can proceed to calculate the next values of $\alpha_{t+1}(v)$, and so on, until $t=T$.

1149 *Appendix B.2. Backwards-algorithm*

Messages are passed backwards in a recursive algorithm starting at $t=T$ and moving backwards until $t=t^0$, the first-capture period, while updating entries in $\beta_t(v)$. $\beta_{t-1}(u)$ is defined as the likelihood of future observations $\mathbf{y}_{t:T}$ from t to T , conditional on $z_{t-1}=u$ at $t-1$.

$$\begin{aligned} \beta_{t-1}(u) &:= p(\mathbf{y}_{t:T} | z_{t-1} = u) \\ &= \sum_v p(\mathbf{y}_{t+1:T} | z_t = v) p(y_t | z_t = v) p(z_t = v | z_{t-1} = u) \\ &= \sum_v \beta_t(v) \Psi_t(v) \Phi_t(u, v) \end{aligned} \tag{B.3}$$

1150 The algorithm is initialized $\beta_T(\cdot) = 1$ for all states v , and proceeds backwards as above. Notice that the
 1151 elements of $\beta_t(\cdot)$ do not need to sum to 1.

1152 Having calculated the backwards and forwards messages, we can now proceed to characterize the latent
 1153 state distributions and boost ϕ and p .

1154 *Appendix B.3. Two-slice marginal probabilities for Expectation-Maximization*

1155 Expectation-Maximization is an iterative technique for maximizing a difficult objective function by work-
 1156 ing with an easy “complete-data” objective function $\log p(y, z | \theta)$. EM works by cycling through an M-step and
 1157 an E-step. In boosting-EM, the M-step corresponds to the usual update of the fit vectors $\hat{F}_\theta^{(m)} = \hat{F}_\theta^{(m-1)} + \nu_\theta \hat{f}$
 1158 (conditional on z), which are used to estimate $\hat{\theta}^{(m)} = \text{logit}^{-1}(\hat{F}_\theta^{(m)})$. The E-step corresponds to imputing
 1159 the expectations of the latent states z , conditional on the data and current estimates of $\hat{\theta}^{(m)}$.

1160 Technically, we require the expectations for the *pairs* of sequential states (z_{t-1}, z_t) . In CJS, these pairs of
 1161 states are simply $\{\text{alive}, \text{alive}\}, \{\text{alive}, \text{dead}\}, \{\text{dead}, \text{dead}\}$. Using the Complete-Data Likelihood, we substi-
 1162 tute in the two-slice marginal probabilities $w_t := p(z_{t-1}, z_t | \mathbf{y}_{t^0:T}, \phi, p)$ for the pairs (z_{t-1}, z_t) . These probabil-
 1163 ities can be calculated easily for a capture-history \mathbf{y}_i using the outputs (α, β) from the forward-backwards
 1164 algorithm.

$$\begin{aligned} w_t(u, v) &:= p(z_{t-1} = u, z_t = v | \mathbf{y}_{t^0:T}) \\ &= \frac{1}{\xi_t} p(z_{t-1} | \mathbf{y}_{t^0:t-1}) p(z_t | z_{t-1}, \mathbf{y}_{t:T}) \\ &= \frac{1}{\xi_t} p(z_{t-1} | \mathbf{y}_{t^0:t-1}) p(y_t | z_t) p(\mathbf{y}_{t+1:T} | z_t) p(z_t | z_{t-1}) \\ &= \frac{1}{\xi_t} \alpha_{t-1}(u) \Psi_t(v) \beta_t(v) \Phi_t(u, v) \end{aligned} \tag{B.4}$$

$$\xi_t = \sum_u \sum_v \alpha_{t-1}(u) \Psi_t(v) \beta_t(v) \Phi_t(u, v), \quad \sum_u \sum_v w_t(u, v) = 1$$

1165 The E-step is completed after evaluating the set $\{w_{i,t}(\text{alive, alive}), w_{i,t}(\text{alive, dead}), w_{i,t}(\text{dead, dead})\}$, for
 1166 each capture period $t > t_i^0$ and for each individual $\{\mathbf{y}_i\}_{i=1}^n$. This is an expensive operation; computational
 1167 time can be saved by re-evaluating the expectations every second or third boosting iteration m , which, for
 1168 large $m_{\text{stop}} > 100$ and small ν , will have a negligible approximation error.

1169 *Appendix B.4. Sampling state-sequences from their posterior*

1170 For the CJSboost Monte-Carlo algorithm, we sample a latent state sequence \mathbf{z}_i from the posterior
 1171 $\pi(\mathbf{z}_{1:T}|\mathbf{y}_{1:T}, \phi, p)$, for each individual i per boosting step m . Conditional on the latent states, the negative-
 1172 gradients are easily evaluated and we can proceed to boost the estimates and descend the risk gradient.
 1173 However, because the algorithm is stochastic, we must avoid getting trapped in a local minima by sampling
 1174 many sequences (e.g., $S \approx 10 - 20$), thereby approximating the full posterior distribution of \mathbf{z} . Over all S
 1175 samples, the average gradient will *probably* be in the direction of the global minima. For large m and small
 1176 ν , the approximation error is small.

The algorithm performs backwards-sampling of the posterior using the chain rule:

$$p(\mathbf{z}_{t^0:T}|\mathbf{y}_{t^0:T}) = p(z_T|\mathbf{y}_{t^0:T}) \prod_{t=T-1}^{t^0} p(z_t|z_{t+1}, \mathbf{y}_{t^0:T}) \quad (\text{B.5})$$

1177 We start with a draw at time $t = T$, $z_T^{(s)} \sim p(z_T = v|\mathbf{y}_{t^0:T}) = \alpha_T(v)$, and condition earlier states on
 1178 knowing the next-step-ahead state, proceeding backwards until $t = t^0$.

$$\begin{aligned} z_t^{(s)} &\sim p(z_t = u|z_{t+1} = v, \mathbf{y}_{t^0:t}) \\ &= \frac{p(z_t, z_{t+1}|\mathbf{y}_{t^0:t+1})}{p(z_{t+1}|\mathbf{y}_{t^0:t+1})} \\ &\propto \frac{p(y_{t+1}|z_{t+1})p(z_t, z_{t+1}|\mathbf{y}_{t^0:t})}{p(z_{t+1}|\mathbf{y}_{t^0:t+1})} \\ &= \frac{p(y_{t+1}|z_{t+1})p(z_{t+1}|z_t)p(z_t|\mathbf{y}_{t^0:t})}{p(z_{t+1}|\mathbf{y}_{t^0:t+1})} \\ &= \frac{\Psi_{t+1}(v)\Phi_{t+1}(u, v)\alpha_t(u)}{\alpha_{t+1}(v)} \end{aligned} \quad (\text{B.6})$$

1179 Thus, knowing α , β , Φ and Ψ , we can easily generate random samples of \mathbf{z} from its posterior distribution.
 1180 The backwards sampling step is repeated for each $t > t_i^0$ capture period, for each s sequence, for each individual
 1181 i , and for each m boosting iteration.

1182 **Appendix C. Algorithms for Tuning the Regularization Parameters**

1183 This section will present a simple work-flow for finding approximately optimal values of m_{stop} , ν_ϕ and
 1184 ν_p that minimize our expected loss \mathcal{L} , a.k.a. the generalization error. We approximate \mathcal{L} through B -fold
 1185 bootstrap-validation. For each b bootstrap, we create a CJSboost model, $G^{(b)}(X; m, \nu_\phi, \nu_p)$ which is trained
 1186 on the bootstrapped data and is a function of the regularization parameters ν_ϕ , ν_p and m . We calculate
 1187 the holdout-out risk using the out-of-bootstrap b^c capture-histories and covariate data, $(\mathbf{Y}^{(b^c)}, \mathbf{X}^{(b^c)})$. The
 1188 objective to minimize is the average hold-out risk, L_{cv} , estimated over B bootstraps.

$$\mathcal{L} \approx L_{cv} = \operatorname{argmin}_{m, \nu_\phi, \nu_p} \frac{1}{B} \sum_{b=1}^B L\left(\mathbf{Y}^{(b^c)}, G^{(b)}(\mathbf{X}^{(b^c)}; m, \nu_\phi, \nu_p)\right)$$

1189 In univariate boosting, it is easy and routine to find the optimal m_{stop} through bootstrap-validation,
 1190 conditional on a fixed value of ν . It is easy because we can simultaneously fit a model *and* monitor the
 1191 holdout-risk per m step. Therefore, we need only perform one round of bootstrapping to find the m_{cv} that
 1192 minimizes the average holdout-risk.

1193 However, the focus of this section will be to estimate the optimal values of ν_ϕ and ν_p . This is a seemingly
 1194 difficult task because they are continuous: we cannot realistically run a different bootstrap exercise per
 1195 combination of $\mathbb{R}^+ \times \mathbb{R}^+$. The challenge of optimizing ν_p and ν_ϕ is not unique to CJSboost, but is inherent to
 1196 all multi-parameter boosting techniques, such as boosted-GAMLSS. Readers who are already familiar with
 1197 the boosted-GAMLSS literature may notice that my approach differs slightly from other authors (e.g. Schmid
 1198 et al., 2013; Mayr et al., 2012). These authors used a single fixed value of ν for all parameters, and then
 1199 optimized separate values of m_θ per parameter θ . Alternatively, I propose to optimize a global m_{stop} for
 1200 both parameters, after optimizing the *ratio* of ν_{θ_1} to ν_{θ_2} . The two methods are equivalent in their outcome. I
 1201 wish to emphasize that although the boosting literature has claimed that there is little benefit in optimizing
 1202 m and/or ν separately for each parameter (Schmid et al., 2013), this is untrue for CJSboost. The optimal
 1203 estimate of $\dot{\nu}_\phi$ may be several orders of magnitude different than the optimal $\dot{\nu}_p$.

1204 The most easy-to-understand method to optimize ν_ϕ and ν_p is to discretize the set of plausible com-
 1205 binations, such as $(10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}) \otimes (10^{-4}, 10^{-3}, 10^{-2}, 10^{-1})$. This is not a terrible idea because
 1206 Bühlmann & Yu (2003) showed that the generalization error has a very shallow minima around the optimal
 1207 values of m . This means that our regularization parameters need only get within the vicinity of their optimal
 1208 values, rather than strict numerical convergence. However, searching for optimal values on a small grid of
 1209 combinations would be very expensive and imprecise. Therefore, we seek an adaptive algorithm that can get
 1210 closer to the optimal values of ν_ϕ and ν_p with only 7-10 bootstrap-validation exercises.

1211 *Appendix C.1. Algorithm 1 for Optimizing Learning-Rates*

1212 For just two parameters (ϕ, p) , we can find the minimum L_{cv} by optimizing the ratio $\lambda = \frac{\nu_p}{\nu_\phi}$, for a fixed
 1213 mean $\nu_\mu = \frac{1}{2}(\nu_\phi + \nu_p)$. We can safely fix ν_μ because it has a straight-forward inverse relationship to m_{stop} ;
 1214 so if we fix one, we merely solve for the other. The point is that we have reduced the problem to a univariate
 1215 search to find the $\dot{\lambda}$ which minimizes $L_{cv}(\lambda)$. Recall also that we can always find the optimal m_{stop} for a
 1216 given λ and ν_μ , so we can drop m from our objective function, which is now a univariate objective:

$$L_{cv} = \operatorname{argmin}_{\lambda} \frac{1}{B} \sum_{b=1}^B L\left(\mathbf{Y}^{(b^c)}, G^{(b)}(\mathbf{X}^{(b^c)}; \lambda | \nu_\mu)\right)$$

1217 This is less daunting than it may seem, because the range of λ is practically bounded. For example, for
 1218 large m_{stop} and $\dot{\lambda} = 100$, then $\nu_p \gg \nu_\phi$, and ϕ is effectively shrunk to its intercept starting value. Higher
 1219 values of λ will have little effect on the generalization error. Also, $L_{cv}(\lambda)$ is typically a convex function of

1220 λ (assuming that as we reuse the same bootstrap-weights for all new estimates of $L_{cv}(\lambda)$). In other words,
 1221 we are searching a U-shaped Real-line for its minimum. This means we can employ any convex optimization
 1222 algorithm for a univariate non-differentiable function to iteratively search for the optimal $\hat{\lambda}$.

1223 The thrust of any such algorithm is a multiplicative “stepping-out” procedure to quickly find the correct
 1224 order of magnitude for $\hat{\lambda}$. For example, starting at $\lambda^{(0)} = 1$, we need only 7 doubling steps to grow λ to
 1225 $128 \times \lambda^{(0)}$; further refinements will have little practical impact on the final model estimates. I suggest the
 1226 following convex optimization algorithm:

- 1227 1. set $\nu_\mu = 0.01$ and $\lambda^{(0)} = 1$; generate the B bootstrap samples and their out-of-sample compliments;
- 1228 2. initialize the sorted list $\Lambda = \{\lambda^{(0)}, \frac{1}{2}\lambda^{(0)}\}$;
- 1229 3. for each λ in Λ , estimate $L_{cv}(\lambda)$ and store the values in the list $\mathbf{L} = \{L_{cv}(\lambda^{(0)}), L_{cv}(\frac{1}{2}\lambda^{(0)})\}$;
- 1230 4. for j in $1:J$, do:
 - 1231 (a) get the current best value for the ratio $\lambda_{\min(L)} = \underset{\lambda \in \Lambda}{\operatorname{argmin}} L_{cv}(\lambda)$
 - 1232 (b) propose a new candidate λ^* :
 - 1233 if $\lambda_{\min(L)} = \min(\Lambda)$, then $\lambda^* = \frac{1}{2}\min(\Lambda)$;
 - 1234 else if $\lambda_{\min(L)} = \max(\Lambda)$, then $\lambda^* = 2 \cdot \max(\Lambda)$;
 - 1235 else $\lambda^* = \lambda_{\min} + k \cdot \alpha$, where k is the step direction and α is the step size.
 - 1236 (c) re-calculate the learning rates from λ^* : $\nu_\phi^{(j)} = \frac{2 \cdot \nu_\mu}{\lambda^* + 1}$; $\nu_p^{(j)} = \lambda^* \cdot \nu_\phi^{(j)}$;
 - 1237 (d) perform bootstrap-validation to estimate $L_{cv}^{(j)}(\lambda^*)$;
 - 1238 (e) append $\Lambda \leftarrow \lambda^*$ and append $\mathbf{L} \leftarrow L_{cv}^{(j)}$;

1239 The algorithm continues until a pre-defined convergence criteria is met, or, practically, a maximum number
 1240 of J iterations is reached. The final values of ν_ϕ , ν_p , and m_{cv} are those which correspond to the minimum
 1241 $L_{cv} \in \mathbf{L}$.

1242 There are many convex optimization algorithms which differ in how they calculate k and α . In CJSboost,
 1243 most of the optimization benefits occur during the “stepping-out” procedure, and so exact values of k and
 1244 α are less important, so long as they guarantee convergence. I suggest the following sub-algorithm (nested
 1245 within step 4b above). This is entirely arbitrary but succeeds in quickly ruling-out large sections of sub-
 1246 optimal values of λ .

- 1247 1. Define the triplet set Γ composed of the current best estimate of $\lambda_{\min(L)}$ as well as the sorted values
 1248 just to the left and right, such that $\lambda_{\min(L)}^{(-1)} < \lambda_{\min(L)} < \lambda_{\min(L)}^{(+1)}$;
- 1249 2. Sort the entries of Γ according to the order $L_{cv}(\gamma^{(1)}) < L_{cv}(\gamma^{(2)}) < L_{cv}(\gamma^{(3)})$;
- 1250 3. Estimate the step size and direction:
 - 1251 if $\|\gamma^{(1)} - \gamma^{(2)}\| \geq \|\gamma^{(1)} - \gamma^{(3)}\|$:
 - 1252 then $\alpha = \frac{1}{2}\|\gamma^{(1)} - \gamma^{(2)}\|$ and $k = \operatorname{sign}(\gamma^{(1)} - \gamma^{(2)})$;
 - 1253 else $\alpha = \frac{1}{2}\|\gamma^{(1)} - \gamma^{(3)}\|$ and $k = \operatorname{sign}(\gamma^{(1)} - \gamma^{(3)})$;

1254 4. $\lambda^* = \lambda_{\min}(L) + k \cdot \alpha$

1255 Typically, seven or ten iterations are necessary in order to find suitable values of $\hat{\lambda}$, ν_ϕ and ν_p . Unfortunately,
 1256 this strategy is only useful for a two-parameter likelihood with a single ratio to optimize. For other capture-
 1257 recapture models with more parameters (e.g., POPAN, PCRD), a different tuning strategy may be necessary,
 1258 such as a bivariate convex optimization algorithm.

1259 *Appendix C.2. Algorithm 2 For Tuning the Learning-Rates ν*

1260 With more parameters in the capture-recapture likelihood, the number of necessary steps in algorithm
 1261 1 will increase exponentially. I suggest a second iterative algorithm whose number of iterations may only
 1262 increase linearly with the number of parameters.

1263 The principle of this second algorithm is based on the observation that when the ratio $\frac{\nu_p}{\nu_\phi}$ is poorly
 1264 optimized, then additional boosting steps along the gradient $\frac{\partial \ell}{\partial F_\theta}$ will *over-fit* and *increase* in the holdout-
 1265 risk. This happens asymmetrically for F_ϕ vs F_p . Therefore, we can monitor the extent of the asymmetry
 1266 and adjust the ratio $\frac{\nu_p}{\nu_\phi}$ until the number of boosting steps which successfully decrease the hold-out risk is
 1267 roughly the same for F_ϕ vs F_p (averaged over all bootstrap hold-out samples).

1268 Call $\Delta_\theta^{(m)}$ a boosting step along the partial derivative of $\frac{\partial \ell}{\partial F_\theta}$ which successfully reduces the holdout-risk.
 1269 I suggest using the ratio of Δ_p vs. Δ_ϕ as an estimate of $\hat{\lambda} = \frac{\nu_p}{\nu_\phi}$.

$$\hat{\lambda}^{(j)} = \hat{\lambda}^{(j-1)} Q \left(\frac{\sum_{m=1}^{m_k} \Delta_p^{(m)}}{\sum_{m=1}^{m_k} \Delta_\phi^{(m)}} \right) \quad (\text{C.1})$$

1270 where Q is a robust measure of central tendency (e.g., trimmed mean) over all B bootstraps, and m_k is some
 1271 boosting step $m_k \gg m_{cv}$.

1272 The first estimate $\hat{\lambda}^{(1)}$ will typically be an underestimate, so the algorithm is iterated, each time using
 1273 the previous values of $\hat{\lambda}^{(j-1)}$ for setting $\nu_p^{(j)}$ and $\nu_\phi^{(j)}$ used to run CJSboost. The bootstrap-validation exercise
 1274 is repeated to estimate the next $\hat{\lambda}^{(j)}$ value according to Eqn. (C.1). $\hat{\lambda}^{(j)}$ will typically converge to a single
 1275 value within approximately 10 iterations. $\hat{\lambda}^{(j)}$ is *not* the optimal $\hat{\lambda}$ as estimated by algorithm 1, but it is
 1276 within the vicinity of the optimal value (Figure C.11).

1277 For just two ν parameters and one ratio (as in CJSboost), this second algorithm is not competitive
 1278 with algorithm 1. But, when there are more than two parameters in the likelihood, this algorithm can
 1279 simultaneously estimate all pertinent ratios.

1280 Further refinements will be necessary. However, these preliminary simulations suggest that the risk
 1281 gradient trajectories have information which can help optimize the regularization parameters.

1282 **Appendix D. Specifying Base-learners**

1283 In component-wise boosting, there are some base-learner parameters that must be specified *a priori*. For
 1284 example, PLS and P-spline base-learners have *effective degrees-of-freedom* parameters which constrain their
 1285 flexibility to fit a process. Schmid & Hothorn (2008a) suggest that such parameters can be fixed to default

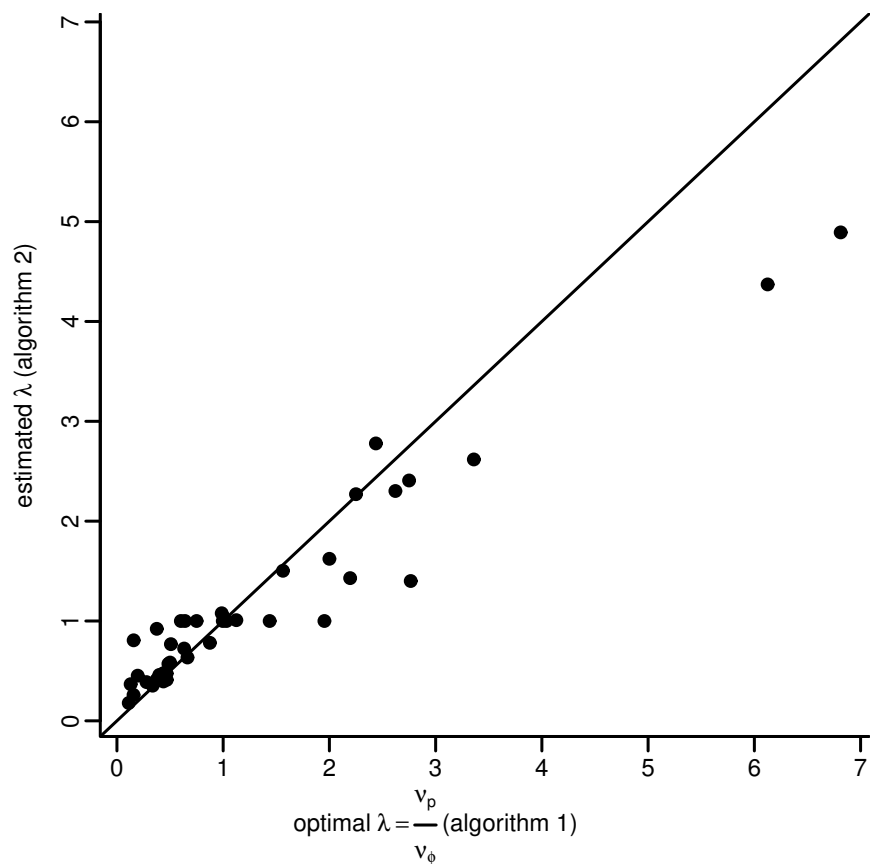


Figure C.11: Two algorithms for tuning the learning-rate regularization parameters ν_ϕ and ν_p , and their ratio λ , in order to minimize the expected loss (estimated via bootstrap-validation). Forty simulations compare the two algorithms, where algorithm 1 is considered optimal.

1286 values, and that practitioners should instead focus primarily on optimizing m_{stop} . Furthermore, Bühlmann &
1287 Yu (2003) suggest that base-learners should be relatively weak, *a priori*, and that the overall model complexity
1288 should be tuned by controlling the shrinkage parameters m_{stop} .

1289 A more important consideration is the relative flexibility of competing base-learners. For example, multi-
1290 covariate learners and unpenalized learners have more flexibility to fit a process and minimize estimation
1291 error. Therefore, they may be preferentially selected in the component-wise boosting algorithm: recall
1292 that in step 7(b) of the CJSboost algorithm, it selects the best base-learner by a goodness-of-fit criterion.
1293 Therefore, practitioners should enforce a similar effective degrees-of-freedom among all base-learners, as well
1294 as decompose higher-order interactions and non-linear curves into their constituent components.

1295 For example, if one desires model-selection among covariates x_1 and x_2 and their interaction $x_1 \times x_2$, then
1296 one should specify four PLS base-learners of equal effective-*df*: one PLS base-learner for the x_1 main-effect; a
1297 second PLS base-learner for the x_2 main-effect; a third PLS base-learner for the main-effects of both x_1 and
1298 x_2 together (no interaction); and a final PLS base-learner for the interaction. This would be analogous to a
1299 shrinkage version of the R GLM model `glm(~x1*x2, ...)`. In the `mboost` R formula interface, the boosted
1300 model would be set-up with the following syntax:

```
1301 ~ bols(x1,df=2)+bols(x2,df=2)+bols(x1,x2,df=2)+bols(x1,by=x2,df=2)
```

1302 For non-linear splines on x_1 , we may wish to separate the linear and non-linear components, called
1303 “centring” in Kneib et al. (2009) and Hofner et al. (2012). In this case, the `mboost` formula interface would
1304 be `~bols(x1)+bbs(x1,center=TRUE,df=1)`.

1305 The above techniques are especially important if practitioners wish to gain some mechanistic understand-
1306 ing of the ϕ and p processes, such as concluding which covariates have a significant contribution to survival.
1307 This is crucial for using the stability-selection-enhanced CJSboost to find ecologically important covariates.

1308 However, when the research goal is not to uncover significant effects, but merely to accurately estimate
1309 abundance, then it is less important to enforce equal *effective-df* among base-learners. An extreme form
1310 of this is when estimation becomes a “black-box” exercise, for example, as with CART-like base-learners:
1311 `~btree(x1,x2,tree_controls=ctree_control(maxdepth=2))`. Here, variable selection and non-linear ef-
1312 fects and interactions are automatically incorporated, at the expense of interpretability.

1313 Appendix E. Primer On The Bias-Variance Trade-off

1314 This appendix uses simulations to illustrate the “bias-variance trade-off” and shows how CJSboost and
1315 the AICc each negotiate the trade-off in order to minimize the expected error of estimating survival ϕ over
1316 T capture periods. The trade-off is fundamental to understanding the optimality of Frequentist shrinkage
1317 estimators and AIC model-selection. The illustrations are inspired by Murphy (2012a, figure 6.5), but adapted
1318 to capture-mark-recapture and the Cormack-Jolly-Seber model.

1319 The trade-off is an old idea without a citable origin (although Geman et al., 1992, is often considered to be
1320 a definitive reference, but the phenomenon is clearly discussed as early as 1970 by Hoerl & Kennard). Despite

1321 being an old and fundamental concept of statistical estimation, I have noticed that it poorly understood among
1322 academics and government scientists. In particular, it is my experience that ecologists are unduly wedded to
1323 the idea of being unbiased (in estimation), such that when they are presented with visual and quantitative
1324 evidence about the optimality of biased shrinkage estimators, they recoil at the sight of systematic bias, and
1325 ignore the crucial role of variance. Of course, bias is not desirable in and of itself, but so long as the bias
1326 goes to zero at a rate proportional to that of the variance, we may be able to improve our overall estimation
1327 performance by incurring a little bias.

1328 In the following simulations, the goal is to minimize the Expected Error of estimating survival, as quan-
1329 tified by the Mean Square Error (MSE). It is a population-level abstract quantity that can only be measured
1330 in simulations when we know to the “true” process. It is Frequentist in the sense that we hope to minimize
1331 the error over all possible data-sets that one might sample from the true population \mathbb{Y} . These multiple
1332 realizations are shown as grey lines in Figures E.12 and E.13. Of course, an analyst only has one dataset,
1333 and his goal is to get his estimates as close as possible to the truth.

1334 The bias-variance trade-off arises from a classic decomposition of the expected error: $\text{MSE} = \mathbb{E}_{\mathbb{Y}}[\hat{\phi} -$
1335 $\phi^{(\text{true})}]^2 + \text{Var}(\hat{\phi}) + c$. Figure E.12 also shows this decomposition. The first term is the expected difference
1336 between an estimate and the true value, i.e. the bias. This difference is visualized as the red polygon in
1337 Figure E.12. In the same figure, the bias manifests as shrinkage from the true red line towards a flat global
1338 mean. Quantifying the bias requires knowledge of the truth $\phi^{(\text{true})}$, and is therefore inaccessible in real-life
1339 situations. The second term is the variance and it does not depend on knowledge of the truth. Rather, it
1340 arises due to the vagaries of random sampling as well as the complexity of the estimation procedure: overly
1341 complex models which “over-fit” one dataset will vary wildly when fitted to a new dataset sampled from the
1342 same population. The variance can be visualized as the spread of the grey lines, or the green polygon in
1343 Figure E.12.

1344 The MSE decomposition has a naive meaning: in order to optimize our estimation performance, we should
1345 reduce the bias and/or the variance. Clearly, most ecologists see the value of tackling either of these two
1346 terms. But the nature of a *trade-off* has a more elusive importance: we cannot, in general, minimize both
1347 terms for a given sample-size, and we may deliberately increase one term in order to decrease the other.
1348 Shrinkage estimators incur a little bias and have lower variance (i.e., the red polygon is bigger but the green
1349 polygon is smaller). This strategy results in much smaller MSE values than complex unbiased estimators. In
1350 contrast, the MLEs of the complex full-model are unbiased but they typically have very high variance. This
1351 strategy is often worse at minimizing the MSE, for small-to-moderate samples sizes.

1352 The following simulations show how different statistical methods have different strategies in negotiating
1353 the bias-variance trade-off. Imagine an analyst who is confronted with four different methods to estimate
1354 survival. The first is estimation by Maximum Likelihood using the full-model $p(t)\phi(t)$. The second method
1355 is AICc model-selection, and the third is AICc model-averaging; both use the following fixed-effects models:
1356 $p(\cdot)\phi(\cdot)$, $p(t)\phi(\cdot)$, $p(\cdot)\phi(t)$ and $p(t)\phi(t)$ with constraints on $p_T = p_{T-1}$ and $\phi_T = \phi_{T-1}$ terms. The fourth

1357 method is CJSboost with base-learners equivalent to the aforementioned fixed-effect models (but without
1358 the previous constraints). The AICc-methods should theoretically do best because they are fundamentally
1359 motivated by trying to minimize an objective function that is very closely related to MSE called the KL-
1360 loss (Akaike, 1974, 1998). Likewise, CJSboost is trying to minimize a related generalization-error called the
1361 negative Expected log-Likelihood, which is approximated through bootstrap-validation.

1362 The fake data-sets were generated according to the following. $\phi_t^{(\text{true})} = \cos\left(\frac{t-2.3}{1.2}\right) / 11 + 0.75$. $p_t^{(\text{true})}$ were
1363 drawn from a beta distribution with shape parameters $A = 12$ and $B = 12$, resulting in an average capture-
1364 probability of 0.5. The $p_t^{(\text{true})}$ values were the same for all simulations. The first-captures were distributed
1365 randomly through-out the capture periods $t \in \{1, \dots, 10\}$, with highest weight on $t = 1$. MLE and AICc
1366 analyses were run in Program MARK (White & Burnham, 1999) and RMark (Laake, 2013). For CJSboost,
1367 a ten-times 70-fold bootstrap-validation exercise was run per dataset to tune the CJSboost regularization
1368 parameters. The simulations and analyses were repeated 40 times for three scenarios pertaining to the number
1369 of capture-histories $n \in \{50, 200, 800\}$.

1370 The results clearly show the trade-off (Figure E.13). At high sample sizes ($n = 800$), the shrinkage
1371 estimator CJSboost has the lowest MSE and therefore wins at estimating survival. However, it has the
1372 highest bias. How can it be considered a better estimator than the other methods when it is biased? The
1373 answer is obvious when looking at the grey lines in Figure E.13, where each line is an estimate of $\{\phi_t\}_{t=2}^T$
1374 from an independent realization of data: compared to the other methods, each grey line from CJSboost is
1375 much more likely to be closer to the truth, despite systematic bias. In contrast, using the MLEs, one can
1376 only claim to be unbiased *over all possible realizations of the data* as shown by the closeness of the dashed
1377 black line to the true red line. But, for any one realization (a single grey line) the MLEs can be very far
1378 away from the truth due to much higher variance.

1379 At smaller sample sizes, we see that the bias becomes much more extreme for both AICc methods and
1380 CJSboost. In the case of the AICc methods, the model with most support is often $\phi(\cdot)$, in which case the
1381 estimates are a single flat line. This is also the case in CJSboost, where shrinkage is so extreme as to force a
1382 flat line. Therefore, at low sample sizes, we are much better off, in terms of MSE, to use the flat-lined $\hat{\phi}(\cdot)$
1383 estimates rather than use the full-model MLEs, which vary so wildly as to be useless.

1384 This primer is meant to illustrate the role of bias and variance in estimation errors. Simulations show
1385 how shrinkage estimators (CJSboost) and model-selection (by AICc) each negotiate the trade-off between
1386 bias and variance to try and minimize the Expected Error. CJSboost does particularly better by incurring
1387 a little bias.

1388 Appendix F. Extra Notes on Stability Selection

1389 In the main article, I introduce stability selection for capture-mark-recapture (CMR) and use it to enhance
1390 the consistency properties of CJSboost, called SS-CJSboost. Stability selection is a new and rapidly growing
1391 group of methods, and SS-CJSboost borrows elements from different but related techniques by Bach (2008)

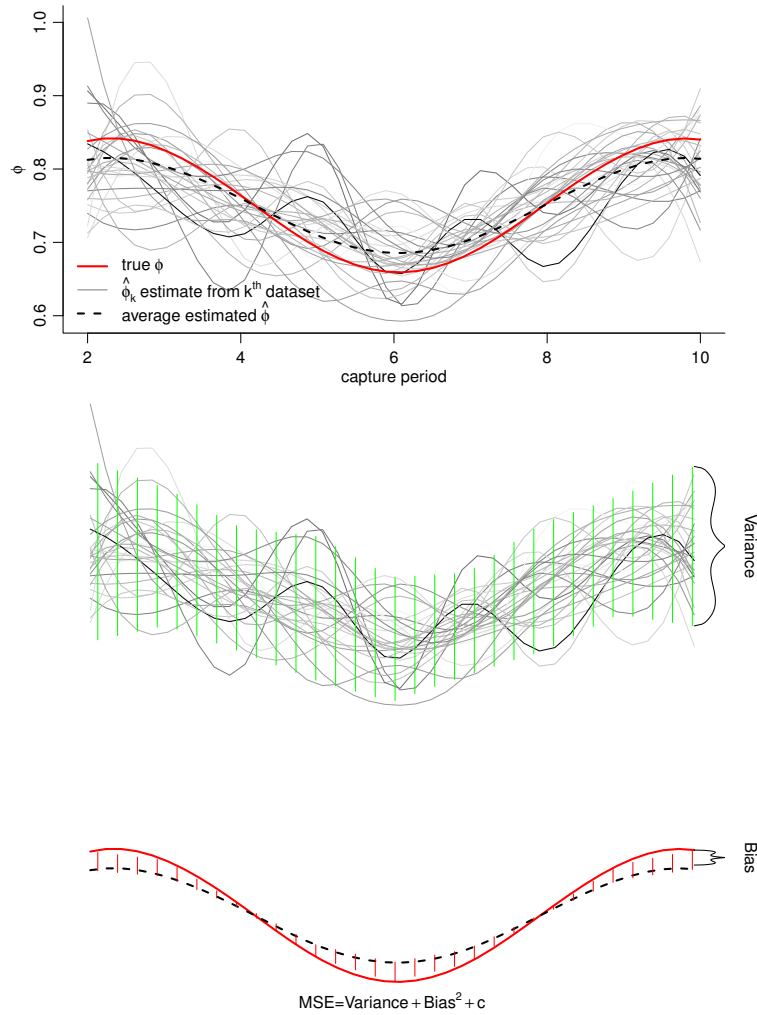


Figure E.12: Decomposing the error of estimation (MSE) into its bias and variance components. An estimation procedure will negotiate the bias and variance so to minimize the MSE. *Top*, a simulation of a true survival process (red line). Each grey line represents one dataset sampled from the population and an analyst's attempt to estimate survival using multi-model inference procedures, such as boosting. The dashed black line is the mean estimate over all 30 independent grey-lines. *Middle*, a visualization of the variance component, showing the variability of point-wise estimates due to randomness in the sampled data and a procedure's sensitivity to such differences. *Bottom*, a visualization of the bias: the expected difference between the truth and the procedure's estimates, over all realizations of the data.

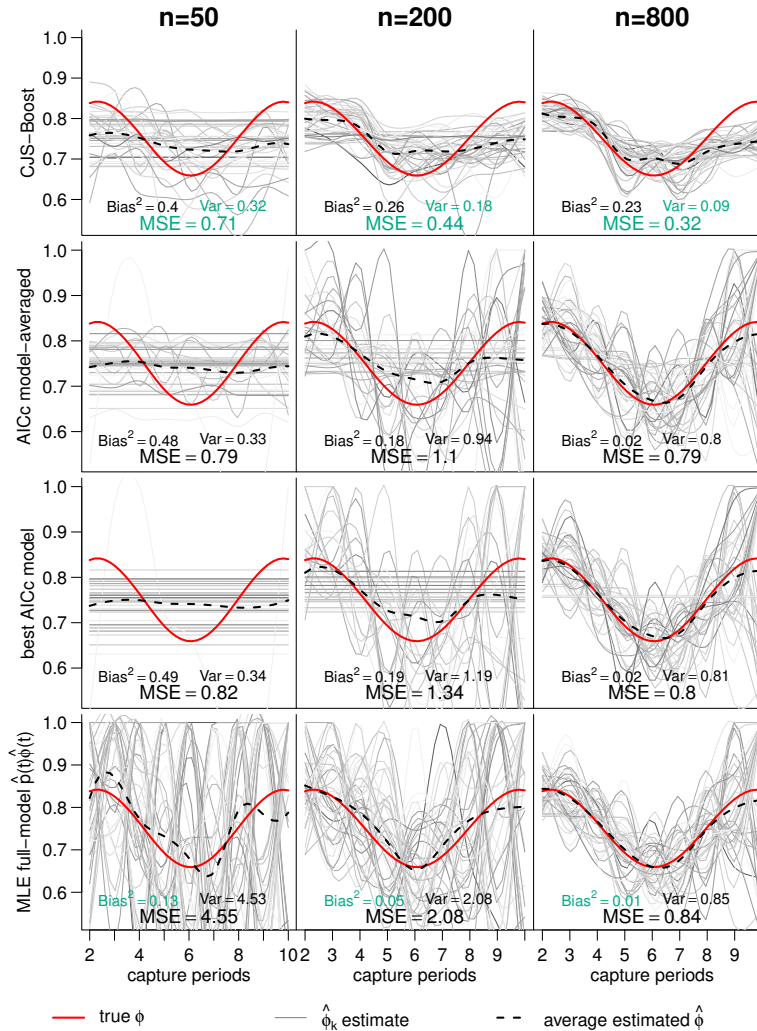


Figure E.13: Visualizing the bias-variance trade-off and the error of estimating survival in a Cormack-Jolly-Seber analysis, using four procedures (*panel rows*): *i*) the shrinkage estimator CJSboost; *ii*) AICc model-averaging based on four fixed-effect models of time-varying vs. time-constant survival and capture-probabilities; *iii*) the best AICc model; and *iv*) the Maximum Likelihood Estimate using the full-model $p(t)\phi(t)$. *Panel columns* are different sample sizes (number of capture-histories) over $T = 10$ primary periods. The red-lines show the true survival. Each grey line is an independently sampled dataset and an analyst's attempt to estimate survival. The dashed-lines represent each procedure's average estimate over 40 simulated datasets and analyses. The best estimation procedure has the lowest MSE (turquoise for emphasis). Each procedure may have a high/low bias or low/high variance, but generally cannot succeed at minimizing both. The bias is the difference between the red and dashed line. The variance is represented by the dispersion among grey lines. At small sample sizes, the AICc methods and boosting are very biased but have better MSE.

1392 and Meinshausen & Bühlmann (2010, hereafter referred to as *MeBü*) and Shah & Samworth (2013, *ShSa*).
1393 In this appendix, I will highlight how SS-CJSboost relates to these methods and where further validation
1394 may be necessary.

1395 To review, the proximate aim of SS-CJSboost is to calculate $\tilde{\Pi}_{\theta,k}$, an approximation of the posterior
1396 inclusion probability, $\pi(\beta_{\theta,k} \neq 0 | \mathbf{Y}, \mathbf{X})$: the probability that a k^{th} covariate is part of the correct model of
1397 θ . Inclusion probabilities are routine in Bayesian analyses to address questions such as: does covariate k
1398 have some structural influence on survival? The analysis proceeds by bootstrapping the capture-histories
1399 B times, and for each b bootstrap running a CJSboost model on the b^{th} resampled data. We must score
1400 whether a covariate has been selected by CJSboost and has entered the ensemble $\mathcal{G}_{\theta}^{(m,\nu)}$, for each value of the
1401 regularization parameters (m, ν) and for each k covariate and for each b bootstrap and for each $\theta \in \{\phi, p\}$. We
1402 denote this selection indicator $I_{\theta,k}^{(b,m,\nu)} = \mathbf{1}[k \in \mathcal{G}_{\theta}^{(b,m,\nu)}]$. A short-cut is to pre-optimize the values of ν_{ϕ} and
1403 ν_p , exactly as one would do in regular CJSboost analysis, and then condition all SS-CJSboost bootstrapped
1404 models on these values, called $\dot{\nu}$. The stability selection probabilities are calculated over B bootstraps per m
1405 and k and θ : $\hat{S}_{\theta,k}^{(m|\dot{\nu})} = \frac{1}{B} \sum_b I_{\theta,k}^{(b,m|\dot{\nu})}$. Finally, our Frequentist inclusion probability is the *mean* of the stability
1406 selection probabilities summed over all values of the regularization parameter m : $\tilde{\Pi}_{\theta,k} = \frac{1}{m_{\text{stop}}} \sum_{m=1}^{m_{\text{stop}}} \hat{S}_{\theta,k}^{(m|\dot{\nu})}$.

1407 Crudely, SS-CJSboost is most similar to the Bolasso (Bach, 2008), but with an emphasis on inclusion
1408 probabilities, as discussed in MeBü and ShSa. In the following paragraphs, I explain where and why certain
1409 techniques were incorporated into SS-CJSboost, and possible problems with the assumptions.

1410 *Selection Procedure.* Bach, MeBü, and ShSa all demonstrate their methods on the Lasso. For Bach, the
1411 consistency results only hold for a region of the Lasso-regularization parameter in relation to sample size.
1412 MeBü allow for any selection procedure, so long as two assumptions hold: i) all the spurious covariates
1413 have the same random distribution of being selected, called “exchangeability”; and ii) the true-covariates are
1414 selected with higher probability. While CJSboost can satisfy the second assumption, the multi-parameter
1415 likelihood may violate the exchangeability assumption; for example, when a covariate significantly influences
1416 capture-probability but not survival, such structural correlations may make certain covariates more select-
1417 able than others. Later on, ShSa weakens these requirements through a special variant of stability selection
1418 called complementary-pairs SS.

1419 *Univariate vs. Multiple-Parameter Regularization.* The theoretical properties derived by Bach, MeBü, and
1420 ShSa were all based on univariate least-squares regularization. Stability selection has since been used for
1421 univariate GLMs and GAMs (see Hofner et al., 2015, and citations therein). At the time of writing this
1422 article, no stability selection work has been published in a multiple parameter context, for example, using
1423 a boosted-GAMLSS model. It is unknown whether any of the theoretical properties of univariate stability
1424 selection hold for multiple-parameter regularization, or for a HMM like CJSboost. Two obvious issues arise.
1425 First, what is the effect of having different generative models for each parameter in the likelihood, and does
1426 this violate the exchangeability assumption? For example, does a k^{th} covariate with a significant effect in

1427 one parameter θ_1 result in a biased-high estimate of $\tilde{\Pi}_{\theta_2, k}$ for another parameter θ_2 ? My simulations suggest
1428 that this is not an issue and such covariates have the same null-distribution of $\tilde{\Pi}$ values as covariates which
1429 are spurious for both θ_1 and θ_2 . Secondly, stability selection demands that we compute $\hat{S}_{\theta, k}^{(\cdot)}$ for all *reasonable*
1430 values of the regularization parameters. This is simple in univariate boosting with only one regularization
1431 parameter, but it becomes computationally unfeasible when the regularization parameter space is bivariate
1432 or trivariate (m and ν). I have proposed a short-cut to set $\hat{\nu}$ to their prediction optimized values, and then
1433 calculate $\hat{S}_{\theta, k}^{(m)}$ over m conditional on $\hat{\nu}$. In simulations, this seems to lead to reasonable $\tilde{\Pi}$ values.

1434 *Subsampling and Resampling.* Bach used the bootstrap, whereas MeBü used subsampling at a rate of 50%,
1435 and ShSa used complementary-pairs sampling by repeatedly dividing the data into equal-halves, but ac-
1436 knowledged the similarity to bootstrapping. For MeBü and ShSa, the exact rate is important for deriving an
1437 upper bound on the expected number of False Discoveries (FD) in least-squares regularization. Their bounds
1438 do not apply naively to multi-parameter regularization, and so there is no reason in CJSboost to maintain
1439 their 50% subsampling rate, which otherwise has some disadvantages. For example, Schmid et al. (2012) had
1440 to subsample at a rate of 80%, and, in lieu of ShSa’s theoretical control on the FDs, they focused instead
1441 on rejecting unimportant covariates with $\tilde{\Pi}$ values below an arbitrary threshold $\pi_{\text{thr}} \in (0.6, 0.9)$. To justify
1442 this alternative use of stability selection, Schmid et al. relied on statements by MeBü that exact values of
1443 $\pi_{\text{thr}} \gg 0.5$ have little impact on the FD error rate. Bach took a different approach, and first found a theoretic-
1444 al region of the Lasso’s regularization parameter λ and sample-size, where truly influential covariates would
1445 be selected with probability ≈ 1 , and spurious covariates would be selected randomly, due to the vagaries of
1446 the sampled data. Therefore, if one had multiple independent realizations of the data, then one could run
1447 the Lasso on all datasets, intersect the selection probabilities, and discard covariates $< 0.9-1$. Of course, one
1448 never has multiple independent datasets, and so Bach suggests the bootstrap to kull covariates that seem
1449 to be selected at random. In CJSboost, it is not clear whether the theoretical properties of the Bolasso
1450 hold, but I rely on research that shows how the Lasso and statistical boosting are near-equivalent estimators
1451 (Bühlmann & Yu, 2003; Efron et al., 2004). Nonetheless, the intuition behind the Bolasso bootstrap is the
1452 same: spurious covariates will have some random selection probability $\ll 1$. This makes SS-CJSboost crudely
1453 similar to the Bolasso, or the *ad hoc* application of stability selection as in Schmid et al. (2012): we calculate
1454 inclusion probabilities and pick a high threshold to reject non-influential or insignificant covariates, in hopes
1455 of obtaining consistent model-selection.

1456 *Role of the Regularization Parameter.* Stability selection probabilities $S_{\theta, k}^{(m)}$ are calculated per value of a
1457 regularization parameter m , while inclusion probabilities $\tilde{\Pi}_{\theta, k}$ are some marginalization over m . MeBü used
1458 a *max* operator. ShSa suggested a *mean* operator, which results in biased $\tilde{\Pi}$ values but with much lower
1459 variance. Richardson (2010) questions whether some other integration over m is desirable. In simulations
1460 with CJSboost, I tried both *max* and *mean* operators, and there was considerably better separation between
1461 true and spurious covariates with the *mean* operator. Using the *max* operator, the overall results were very

1462 similar to Figure 7, except that the spurious covariates obtained higher $\tilde{\Pi}$ values, and there was a lot more
1463 variability among $\tilde{\Pi}$ values. Also, the spurious time-as-a-categorical variables converged to ≈ 1 , for both ϕ
1464 and p .

1465 *Inclusion Probabilities.* The idea that stability selection can be used to approximate Bayesian posterior
1466 inclusion probabilities was mentioned in the Discussion and Rejoinder of MeBü by Richardson (2010) and
1467 Draper (2010). Therefore, I suggest that $\tilde{\Pi}$ values represent interpretable end-points for a CMR analysis
1468 and can lead to correct inferences about the significance of covariates, as in Bayesian multi-model studies.
1469 Further study will be necessary to elucidate the implied prior and whether there is any meaning in the $\tilde{\Pi}$
1470 values beyond their original role as thresholding statistics. The original developers of stability selection did
1471 not espouse such a view: MeBü and ShSa use $\tilde{\Pi}$ as a threshold to control FDs; Schmid et al. (2012) wished
1472 to pre-screen a high-dimensional dataset of its spurious covariates; and Bach explicitly desired a method to
1473 discard covariates and derive a consistent estimator. In other words, stability selection and $\tilde{\Pi}$ are tools to
1474 threshold one's candidate set of covariates, and then perform estimation (but see Leeb & Pötscher, 2008).
1475 Direct interpretation of the $\tilde{\Pi}$ values will require further study, but the CJSboost simulations suggest that
1476 $\tilde{\Pi}$ offer a fruitful means of inference about the true model.