

Distance Sampling with Random Scale Detection Function

Jeff Laake and Hans Skaug

November 30, 2012

1 Introduction

Distance sampling was developed to estimate abundance from observational surveys with visibility bias (Buckland et al. 2001). A probability density function is fitted to the sample of observed distances to infer the detection probability for the sample. For line transect sampling, distance is the perpendicular distance from the line to the object and for points, the radial distance is from the point to the object. The detection function $g(x)$ is the probability of seeing an object at distance x given the object is at that distance and $h(x)$ is the probability that the object is at distance x . The probability density function $f(x)$ for the sample of observed distances is:

$$f(x) = \frac{g(x)h(x)}{\int g(u)h(u)du} \quad ,$$

which is the conditional probability the object was seen at x given it was seen somewhere. From the assumption of random line or point placement, we assume uniform density locally with respect to x over a sample of lines or points. For lines, this means that $h(x) = 1/w$ where w is the strip half-width w and for points $h(x) = 2\pi x/(\pi w^2)$ where w is the radius of the circle. From here on, we will refer to line transect sampling although the methods we describe are the same for points with the adjustment for a different $h(x)$. With $h(x) = 1/w$, $f(x)$ simplifies to

$$f(x) = \frac{g(x)}{\int_0^w g(u)du} \quad . \tag{1}$$

With the additional assumption that detection at $x=0$ is perfect ($g(0) = 1$), then

$$f(0) = \frac{1}{\int_0^w g(u)du} \quad . \quad (2)$$

and for n observations from strips of length L and width $2w$, object density is:

$$D = \frac{n}{2wLp} = \frac{n}{2wL \int_0^w g(u)\frac{1}{w}du} = \frac{nf(0)}{2L} \quad , \quad (3)$$

where $p = \int_0^w g(u)\frac{1}{w}du$ is the average detection probability. The quantity $\mu = \int_0^w g(u)du = wp$ is called the effective strip width (ESW), but is actually a half-strip width for each side of the line.

Currently there are 3 primary ways to fit detection functions for distance sampling data. The most common is the key function and adjustment series described in Buckland et al (2001). The general formula is

$$g(x) = \frac{k(x)(1 + \sum_{j=1}^m a_j p_j(x))}{k(0)(1 + \sum_{j=1}^m a_j p_j(0))}$$

where $k(x)$ is a key function and $p_j(x)$ is a series of adjustment functions with coefficients a_j . The demoninator scales the function such that $g(0)=1$ although it is not necessary for fitting because the denominator cancels in eq 1. An example is a half-normal key function and a cosine adjustment series

$$g(x) = \frac{\exp(-(x/\gamma)^2/2)(1 + \sum_{j=1}^m a_j \cos(j\pi x/w))}{(1 + \sum_{j=1}^m a_j)}$$

The key-adjustment approach is quite flexible but requires a selected truncation width (w), requires non-linear constraints to maintain monotonicity and ensure $1 > g(x) > 0$, and fitting of detection functions with long tails is problematic.

A second approach is to include explanatory covariates (z) in the detection function (Marques and Buckland 2003). An example using a half-normal detection function is:

$$g(\mathbf{x}|\mathbf{z}) = \exp(-[\mathbf{x}/\exp(\mathbf{X}_z\boldsymbol{\beta})]^2/2)$$

where \mathbf{x} is the vector of distance, \mathbf{z} is a data vector or matrix, \mathbf{X}_z is a design matrix for the covariates and $\boldsymbol{\beta}$ is a parameter vector. The analysis is conditional on \mathbf{z} , so it is essential that the unspecified distribution for \mathbf{z} is independent of \mathbf{x} . An obvious example that fails would be to use animal behavior that might differ with \mathbf{x} (*e.g.* movement). This approach provides monotone detection functions without constraints, does not require truncation and can fit long tails. It has the added

advantage of providing better small-area estimates of density when the covariates vary spatially. The covariate approach does depend on being able to identify and measure covariates that affect detection probability. If there is any remaining lack of fit, the first and second approaches can be combined using covariates in the key function and a series adjustment; however, it is then fraught with the same problems and the constraints can become even more problematic because they depend on the explanatory covariate values. Even if the problem is constrained correctly for all observed values of \mathbf{z} , predictions for unobserved values of \mathbf{z} may yield invalid probabilities with the addition of adjustment functions.

The third approach is rather recent and involves fitting a mixture of detection functions (Miller and Thomas in prep) along the lines of Pledger (2000) for capture-recapture models. The detection function can be represented as:

$$g(x) = \sum_{j=1}^m \pi_j g'_j(x)$$

where $\sum_{j=1}^m \pi_j = 1$ and $g'_j(x)$ is a properly specified detection function. As long as each component detection function is monotone, $g(x)$ will be monotone.

2 Random Scale Detection Function

An additional approach we present here is to use random effects in the detection function scale to allow for unmodelled heterogeneity in detection. For example, consider a half-normal detection function with an intercept and a normal error with mean 0:

$$g(x|\epsilon) = \exp(-x^2/(2\gamma(\epsilon)^2))$$

and

$$\gamma(\epsilon) = \exp(\beta + \epsilon) \quad .$$

We will assume a normal distribution for ϵ and use the shorthand $N(\epsilon, 0, \sigma_\epsilon) = \exp\left(-0.5\left(\frac{\epsilon}{\sigma_\epsilon}\right)^2\right) (\sqrt{2\pi}\sigma_\epsilon)^{-1}$ for the normal density function evaluated at ϵ with mean 0 and standard deviation σ_ϵ . If you consider that long-tails may result from some objects or some conditions under which the object is detectable even at long distances, then it is clear that the random scale will be able to cope with long-tailed detection functions.

If you think about the random effect as an unknown covariate value, then the likelihood can be derived directly from equations 2.39 and 2.40 in Borchers and Burnham et al. (2004) by including an integral over the unknown random effect:

$$\prod_{i=1}^n \frac{\int_{-\infty}^{\infty} g(x_i|\epsilon) N(\epsilon, 0, \sigma_{\epsilon}) d\epsilon}{\int_{-\infty}^{\infty} \int_0^w g(u|\epsilon) du N(\epsilon, 0, \sigma_{\epsilon}) d\epsilon} \quad (4)$$

The numerator is the marginal probability that the object was seen at x :

$$\int_{-\infty}^{\infty} g(x|\epsilon) N(\epsilon, 0, \sigma_{\epsilon}) \frac{1}{w} d\epsilon \quad , \quad (5)$$

and the denominator is the probability that the object was seen:

$$\int_{-\infty}^{\infty} \int_0^w g(u|\epsilon) du N(\epsilon, 0, \sigma_{\epsilon}) \frac{1}{w} d\epsilon \quad , \quad (6)$$

An alternative likelihood is:

$$\prod_{i=1}^n \frac{\int_{-\infty}^{\infty} f(x_i|\epsilon) N(\epsilon, 0, \sigma_{\epsilon}) d\epsilon}{\int_{-\infty}^{\infty} \int_0^w f(u|\epsilon) du N(\epsilon, 0, \sigma_{\epsilon}) d\epsilon} \quad (7)$$

where

$$f(x|\epsilon) = \left(\sqrt{2\pi\gamma(\epsilon)} \right)^{-1} \exp \left(-x^2 / (2\gamma(\epsilon)^2) \right)$$

is scaled to be a probability density function. These two likelihoods are equivalent except that $\beta_f = \beta_g + \sigma_{\epsilon}^2$ where β_f and β_g are derived from eq. 7 and eq. 4 respectively. One might be tempted to use the following likelihood:

$$\prod_{i=1}^n \int_{-\infty}^{\infty} \frac{g(x_i|\epsilon)}{\int_0^w g(u|\epsilon) du} N(\epsilon, 0, \sigma_{\epsilon}) d\epsilon \quad (8)$$

but it is not correct and will produce biased estimates. If $w = \infty$, then the relationship $\beta_f = \beta_g + \sigma_{\epsilon}^2$ holds where β_f is derived from eq 8 but it does not hold if the data are truncated. A mixed-effects model in which explanatory covariates (\mathbf{z}) are included in the detection function can be accomplished by using:

$$\gamma(\epsilon, \mathbf{z}) = \exp(\mathbf{X}_{\mathbf{z}}\boldsymbol{\beta} + \epsilon) \quad . \quad (9)$$

where $\mathbf{X}_{\mathbf{z}}$ is a design matrix for the known covariate values. In this case, the likelihood is conditional on the observed covariate values.

The estimate of density or likewise abundance can be obtained using eq. 6 in place of p in eq 3. When explanatory covariates are included the Horvitz-Thompson-like estimator (eq 2.44 in Borchers and Burnham 2004) can be used:

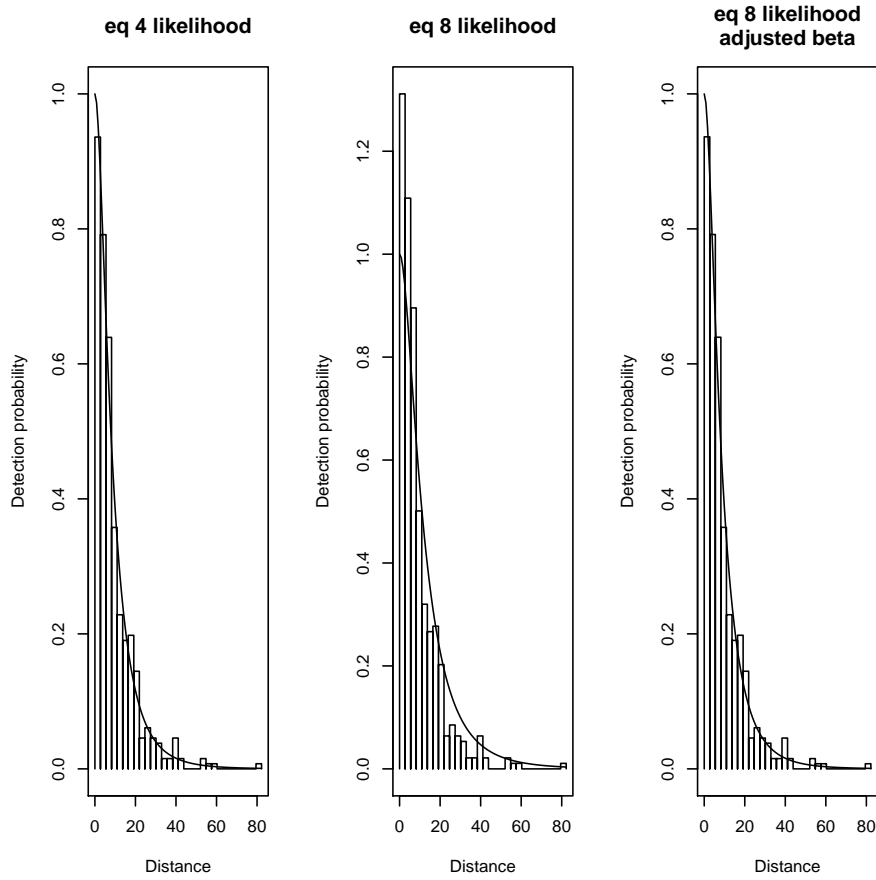
$$D = \sum_{i=1}^n \frac{1}{2wLp_i} = \sum_{i=1}^n \frac{1}{2wL \int_{-\infty}^{\infty} \int_0^w g(u|\epsilon, z_i) du N(\epsilon, 0, \sigma_\epsilon) \frac{1}{w} d\epsilon} \quad , \quad (10)$$

3 Model fitting code

R code for fitting models using maximum likelihood with likelihood eq. 4 and 8, to plot the fitted model and to compute abundance using eq 10 multiplied by $2wL$ is in the package RandomScale (<https://github.com/jlaake/RandomScale>). In addition, the RandomScale package contains TPL files that were developed to fit the models with ADMB (Fournier et al. 2012) using each of the likelihoods (eq. 4, 7, and 8). Additional C++ code was also written to enable use of multinomial weights with Gauss-Hermite integration for the random effects and is contained in the package. From some simulation results using the various likelihood equations, we have found that in ADMB eq. 4 is less stable than eq. 7 when the data were substantially truncated such that the data contains less information about the random effect.

The following generates some simulated data, fits models with the R code using likelihood equations 4 and 8, and shows plots for each including one with the adjustment of $\beta_f = \beta_g + \sigma_\epsilon^2$:

```
> x=simdata(n=500,w=Inf,beta_eps=-.5)
> par(mfrow=c(1,3))
> results_random=fitdata(x,w=Inf,beta_eps=-.5)
> plotfit(x,w=max(x),results_random$model$par,nclass=30,
+         main="eq 4 likelihood")
> results_random_wrong=fitdata(x,w=Inf,beta_eps=-.5,wrong=TRUE)
> param=results_random_wrong$model$par
> plotfit(x,w=max(x),param,nclass=30,main="eq 8 likelihood")
> plotfit(x,w=max(x),c(param[1]-exp(2*param[2]),param[2]),nclass=30,
+         main="eq 8 likelihood\nadjusted beta")
>
```

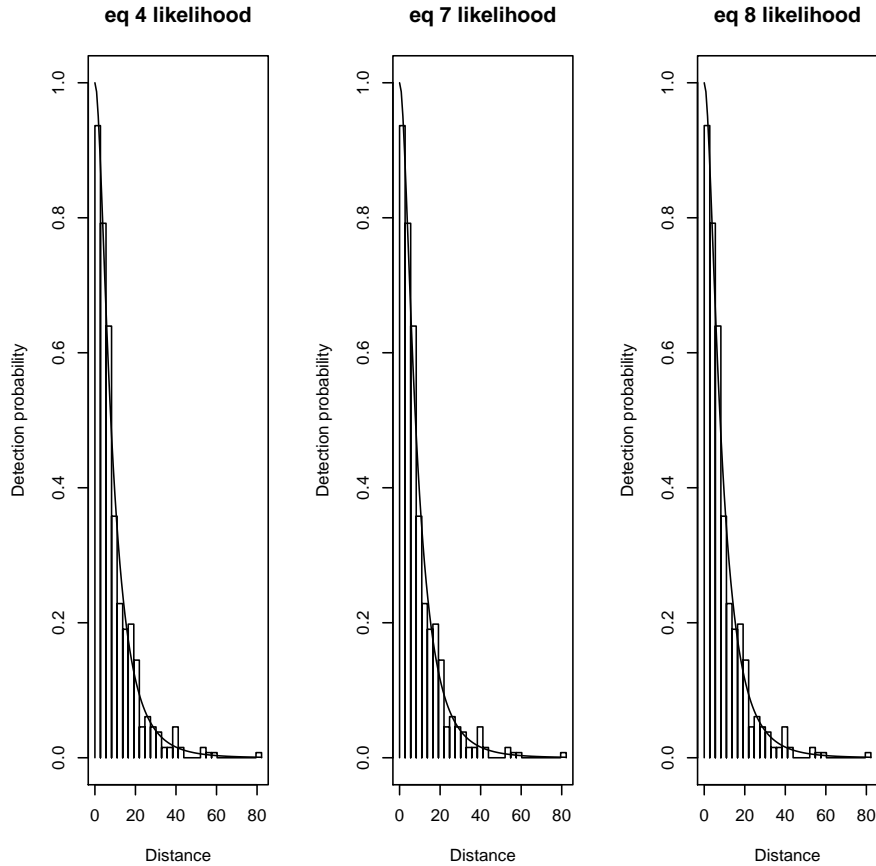


Using the same data, we use ADMB to fit models using likelihood equations 4, 7, and 8 and plot the fitted models with the appropriate adjustments to β_f . The results from the R code and ADMB are nearly identical (Table 1).

```
> par(mfrow=c(1,3))
> gllike=fitadmb(x,w=Inf,likelihood="g")
> plotfit(x,w=Inf,c(gllike$coeflist[[1]],gllike$coeflist[[2]]),nclass=30,
+         main="eq 4 likelihood")
> f2like=fitadmb(x,w=Inf,likelihood="f2")
> param=c(f2like$coeflist[[1]],f2like$coeflist[[2]])
> plotfit(x,w=Inf,c(param[1]-exp(2*param[2]),param[2]),nclass=30,
+         main="eq 7 likelihood")
> f1like=fitadmb(x,w=Inf,likelihood="f1")
> param=c(f1like$coeflist[[1]],f1like$coeflist[[2]])
> plotfit(x,w=Inf,c(param[1]-exp(2*param[2]),param[2]),nclass=30,
```

+

`main="eq 8 likelihood")`



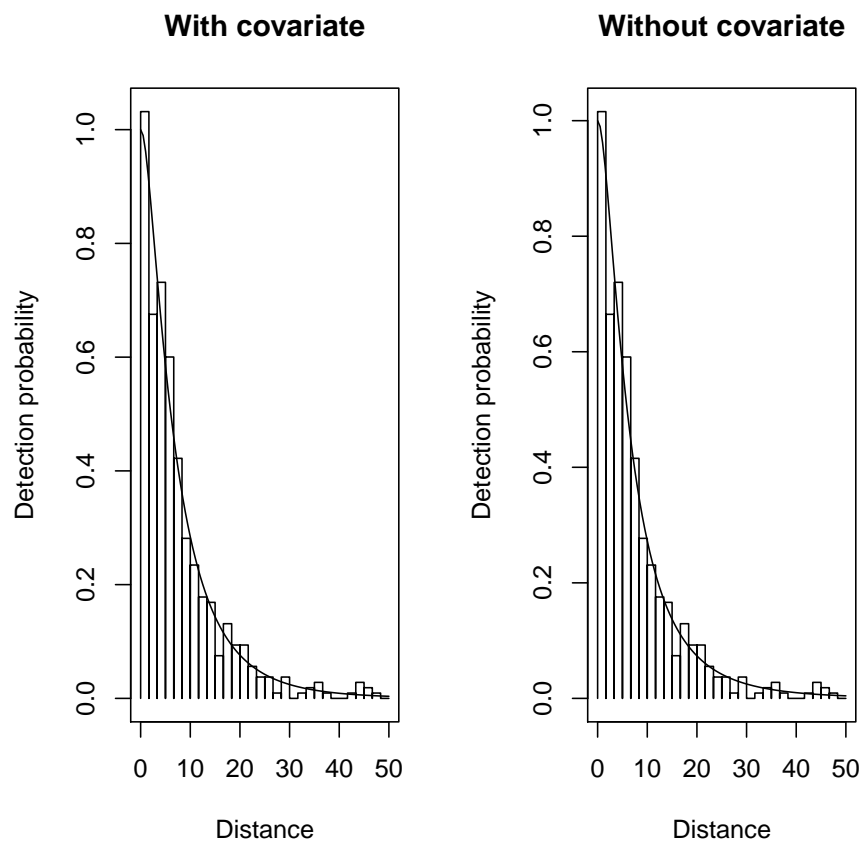
The following is an example of a mixed-effects model that can only be fitted at present with the ADMB code and likelihood eq. 7. We simulate data from a population with 2000 objects with a larger scale and 1000 objects with a smaller scale. All objects have the same random effect distribution. We fit a model with and without the covariate to the data to the model. The arguments for plotfit are a little clumsy at present and a more flexible structure will be created.

```
> par(mfrow=c(1,2))
> x1=simdata(n=2000,w=50,beta_eps=-.5,beta=2,fixed=FALSE,reject=TRUE)
> x2=simdata(n=1000,w=50,beta_eps=-.5,beta=1,fixed=FALSE,reject=TRUE)
> df=data.frame(covariate=c(rep(0,length(x1)),rep(1,length(x2))),
+   distance=c(x1,x2))
> fwlike=fitadmb(df,w=50,formula=~covariate,likelihood="f2")
```

```

> param=c(fwlike$coeflist[[1]],fwlike$coeflist[[2]])
> Nhatwcov=plotfit(df$distance,w=50,
+                 par=c(param[1]-exp(2*param[3]),param[2],param[3]),
+                 nclass=30,dm=model.matrix(~covariate,df),main="With covariate")
> flike=fitadmb(df,w=50,formula=~1,likelihood="f2")
> param=c(flike$coeflist[[1]],flike$coeflist[[2]])
> Nhatwocov=plotfit(df$distance,w=50,
+                 par=c(param[1]-exp(2*param[2]),param[2]),nclass=30,
+                 main="Without covariate")

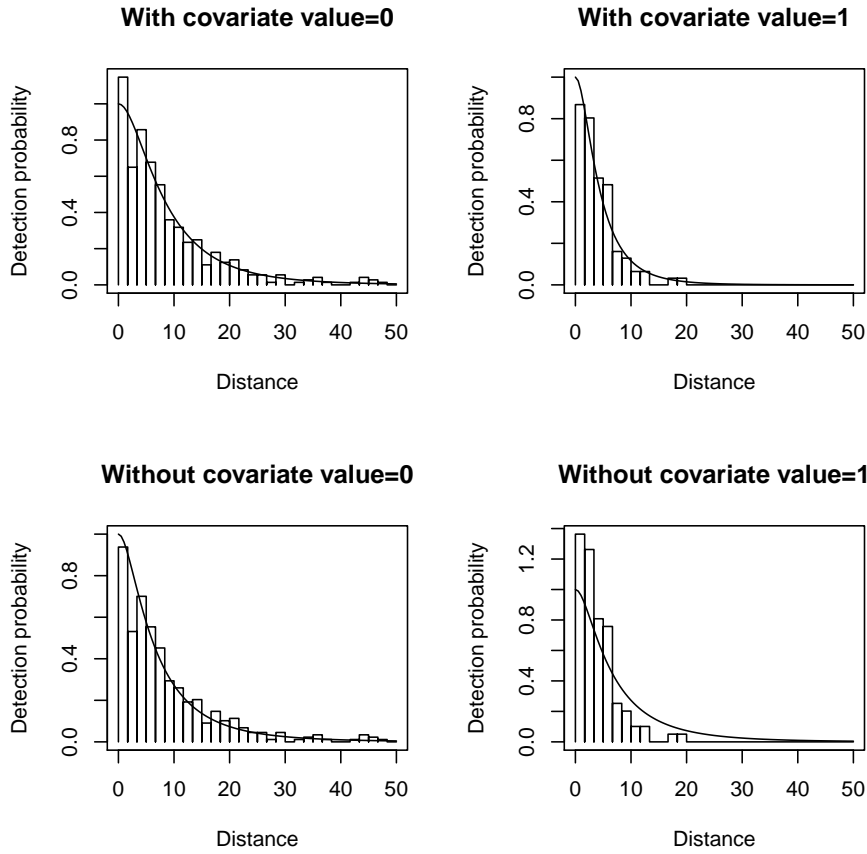
```



The fit of each model looks similar in the plots but the model with the covariate was clearly better with a delta AIC of 37.62. The estimates of abundance (with covariate: 3199 without covariate: 3249) are similar because when the covariate is excluded from the model the variation from the covariates is absorbed into the variance of the random effect which captures unmodeled heterogeneity.

Table 1: Comparison of random scale model fitting results from R and ADMB using simulated data.

Code	Likelihood eq	Log-likelihood	β_f	β_g	$\log(\sigma_\epsilon)$
R	eq. 4	-1614.82	NA	1.950	-0.538
R	eq. 8	-1614.82	2.291	1.950	-0.538
ADMB	eq. 4	-1614.79	NA	1.948	-0.536
ADMB	eq. 7	-1614.79	2.290	1.948	-0.536
ADMB	eq. 8	-1614.80	2.290	1.948	-0.537



However, when abundance is estimated for each type of object (with covariate: 2170 and 933.4; without covariate: 2655 and 594.1) and the plots are shown for each type of object, the importance of including the covariate is obvious.