

Kangaroo distance sampling survey along roads

Marques and Rexstad

2019-06-18

Some set of introductory comments about the vignette

Make package functions available

```
library(densgrad)
```

Acquire data

Data come from three sources:

- distance sampling survey (transect lengths, perpendicular distances and group size)
- random locations within truncation distance of the transects (road)
 - actually, random data don't appear to be part of the actual analysis
- GPS collar data from telemetered kangaroos within truncation distance of transects

```
real <- read.csv(file="../data/GSPoints.txt", header=TRUE, sep="\t")
```

Make some adjustments to field names

```
names(real) <- c("infid", "nearfid", "dist", "reserve", "veg", "animal")
```

Actual distance sampling survey data

```
survey <- read.csv(file="../data/nonstratified_farrar.txt", header=TRUE, sep="\t")
```

Data munging

Transect 23 had no kangaroo detections, hence in the data the perpendicular distance is recorded as NA. That record provides no information about the detection function and needs to be removed for purposes of parameter estimation (note the effort associated with transects without detections does need to be retained). Similarly, the truncation distance is set here and the survey data edited accordingly.

```
survey.truncation <- 150
gradient.max.distance <- 100
survey.data <- survey$Distance[!is.na(survey$Distance) & survey$Distance<survey.truncation]
```

Density gradient estimation

Apparently this optimisation is not really necessary, but perhaps used for plotting?

```
gradient <- optim(par=c(35,1.2),
                 fn=likdensity,
                 distances=real$dist,
                 truncation=gradient.max.distance,
                 control=list(fnscale=-1))#,method="L-BFGS-B",lower=c(20,-100),upper=c(200,100))
if (gradient$convergence == 0) {
  sigma.est <- gradient$par[1]
  beta.est <- gradient$par[2]
  likelihood <- gradient$value
} else {
```

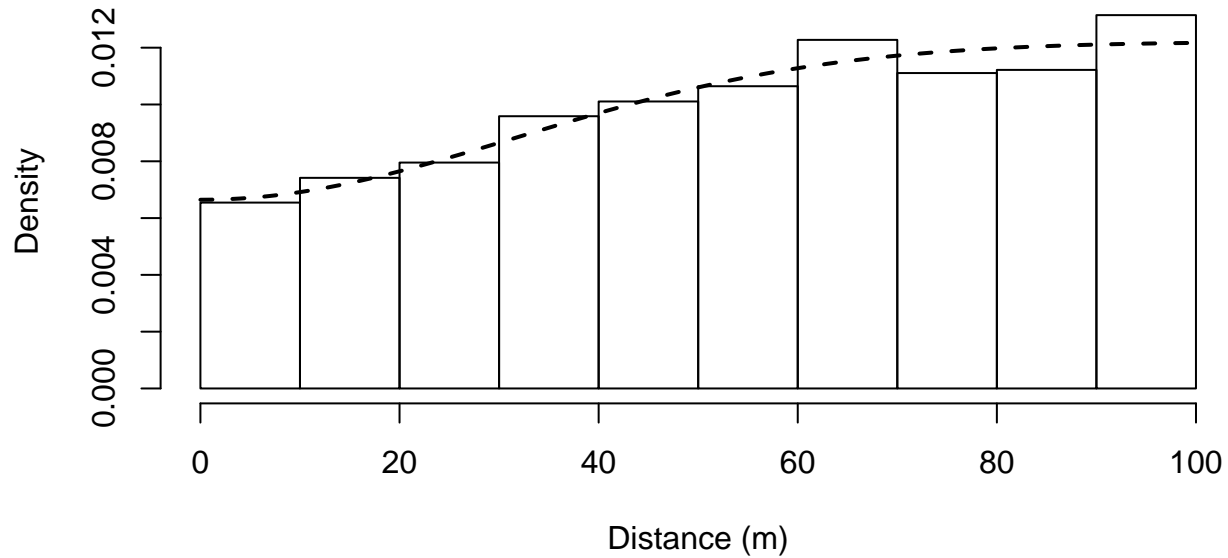


Figure 1: Fitted animal density gradient

```
print("Convergence not achieved, no parameter estimates produced")
}
```

Let's see a plot of the fitted animal distribution with respect to transects

```
xs <- seq(from=0, to=gradient.max.distance)
par(mfrow=c(1,1),mar=c(4,4,0.5,0.5))
hist(real$dist,prob=T,main="",xlab="Distance (m)")
lines(xs,pi.x(xs,sigma.est,beta.est,gradient.max.distance),type="l",lty=2,lwd=2)
```

Joint likelihood estimation

First the half normal detection function

```
hn.det <- optim(par=c(50,35,35),
               fn=full.lik.fx.HN,
               detectdists=survey.data,
               gpsdists=real$dist,
               truncdet=survey.truncation,
               truncden=gradient.max.distance,
               control=list(fnscale=-1))#,method="L-BFGS-B",lower=c(20,-100),upper=c(200,100))
if (hn.det$convergence == 0) {
  sigma.detect.est <- hn.det$par[1]
  sigma.gradient.est <- hn.det$par[2]
  beta.est <- hn.det$par[3]
  likelihood.hn <- hn.det$value
  aic.hn <- -2 * likelihood.hn + 2 * length(hn.det$par)
} else {
  print("Convergence not achieved, no parameter estimates produced")
}
```

Next the hazard rate detection function

```

hz.det <- optim(par=c(50,1.2,35,35),
               fn=full.lik.fx.HR,
               detectdists=survey.data,
               gpsdists=real$dist,
               truncdet=survey.truncation,
               truncden=gradient.max.distance,
               control=list(fnscale=-1))#,method="L-BFGS-B",lower=c(20,-100),upper=c(200,100))
if (hz.det$convergence == 0) {
  sigma.detect.est <- hz.det$par[1]
  beta.detect.est <- hz.det$par[2]
  sigma.gradient.est <- hz.det$par[3]
  beta.gradient.est <- hz.det$par[4]
  likelihood.hz <- hz.det$value
  aic.hz <- -2 * likelihood.hz + 2 * length(hz.det$par)
} else {
  print("Convergence not achieved, no parameter estimates produced")
}

```

The AIC for the half normal detection model is 52918.51 while the corresponding value for the hazard rate detection model is 52919.34.

Estimated parameters from both models:

```

hn.estimates <- t(c(aic.hn, hn.det$par[1], NA, hn.det$par[2:3]))
hz.estimates <- t(c(aic.hz, hz.det$par))
param.table <- as.data.frame(rbind(hn.estimates, hz.estimates))
rownames(param.table) <- c("Half normal", "Hazard rate")
colnames(param.table) <- c("AIC", "$\\widehat{\\sigma}_D$", "$\\widehat{\\beta}_D$",
                           "$\\widehat{\\tau}_G$", "$\\widehat{\\beta}_G$")
knitr::kable(param.table, digits=2, caption="Parameter estimates for two joint models.")

```

Table 1: Parameter estimates for two joint models.

	AIC	$\widehat{\sigma}_D$	$\widehat{\beta}_D$	$\widehat{\tau}_G$	$\widehat{\beta}_G$
Half normal	52918.51	73.20	NA	31.42	35.99
Hazard rate	52919.34	82.02	2.99	32.21	36.44

Plot of selected detection function

```

hist(survey$Distance[survey$Distance<survey.truncation],
     main="", yaxt="n", ylab="g(y)", xlab="Distance (m)")
xs <- seq(from=0, to=survey.truncation)
lines(xs, 60*detfn(z=xs, pars=hn.det$par[1], key="HN", adjn=0, w=survey.truncation),
      ylab="g(y)", lwd=2, type="l", lty=3)
axis(2, at=seq(0, 60, length=5), labels=seq(0, 1, length=5))

```

Probability of detection in covered region

Integration of the selected detection function over all perpendicular distances gives an estimate of detection probability (\hat{P}) used to adjust number of detections to account for imperfect detectability.

```
Phat <- estimate.Phat("HN", hn.det$par, survey.truncation)
```

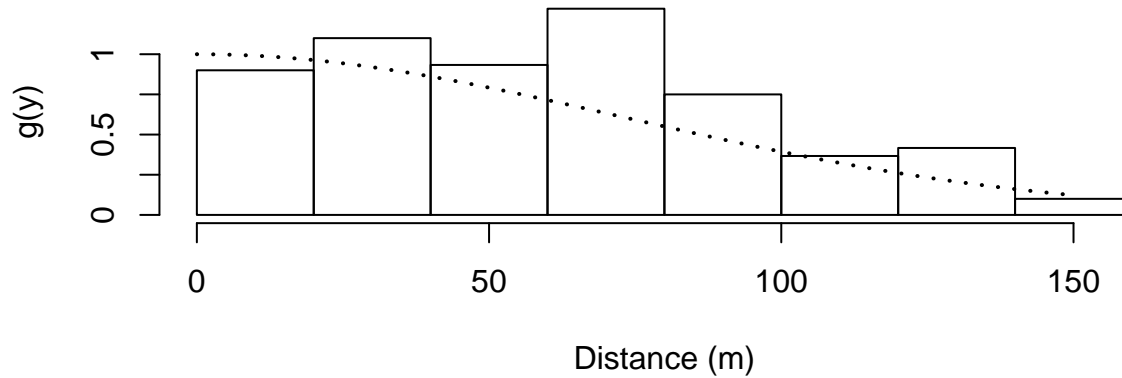


Figure 2: Fitted half normal detection function accounting for non-uniform distribution.

Density estimate of individuals

To estimate density in the covered region, the classic formula is applied,

$$\hat{D} = \frac{n \cdot E[s]}{2wL\hat{P}}$$

using probability of detection derived from the best function that incorporates a density gradient.

In addition, three statistics from the survey are required:

- number of detections (n)
- total survey effort (L)
- average group size ($E[s]$) (size bias adjustment could be incorporated)

with truncation distance (w) earlier specified by the user. Note the argument to the function below is the survey data frame, including transects on which no detections were made. This is necessary such that effort (L) is calculated correctly.

```
survey.design <- survey.design.summary(survey, survey.truncation)
n <- survey.design$n
L <- survey.design$L
Es <- survey.design$Es
```

Perpendicular distances as well as survey effort were recorded in meters. However, the density estimate in the original paper was reported as numbers ha^{-2} . The conversion factor of 10000 converts density from m^{-2} to ha^{-2} . Estimated density is then

```
Dhat <- (n * Es) / (2 * survey.truncation * L * Phat) * 10000
```

Size bias adjustment to group size

```
Esadj <- size.bias("HN", survey.data, survey, real$dist, survey.truncation, gradient.max.distance)
Dadj <- (n * Esadj) / (2 * survey.truncation * L * Phat) * 10000
```

Point estimate

```
out.table <- data.frame(n=n, Es=Es, Esadj=Esadj, w=survey.truncation, L=L,
                        P=Phat, D=Dhat, Dadj=Dadj)
colnames(out.table) <- c("n", "E(s)", "$E(s)_{sb}$", "w", "L", "$\\hat{P}$", "$\\hat{D}$", "$\\hat{D}_{sb}$")
knitr::kable(out.table, caption="Summary statistics and estimate of density from selected model",
              digits = 3)
```

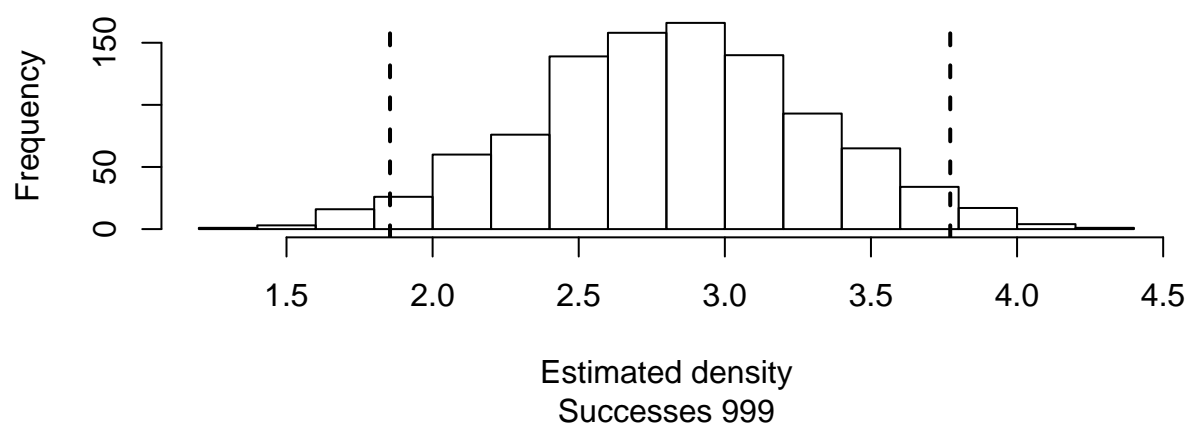
Table 2: Summary statistics and estimate of density from selected model

n	E(s)	$E(s)_{sb}$	w	L	\hat{P}	\hat{D}	\hat{D}_{sb}
351	2.125	1.755	150	13400.69	0.542	3.426	2.829

Variance estimates

```
bounds <- varbootstrap("HN", survey, real, survey.truncation, gradient.max.distance,
                       Nboot=999, plotit=TRUE, alpha=0.05)
#> MIfuncs 5.1
```

Bootstrap density estimates



```
knitr::kable(data.frame(LCI=signif(unname(bounds$CIbounds[1]),4), UCI=signif(unname(bounds$CIbounds[2]),4))
```

LCI	UCI
1.854	3.771

Because details of each simulation replicate is saved and returned as the first object in the named list, analysts can examine the sampling distribution of all the parameters estimated by the joint likelihood: $\hat{\sigma}_d, \hat{b}, \hat{\sigma}_x, \hat{\beta}$ as well as the derived parameters \hat{P} and \hat{D} .

```
alpha <- 0.05
alphaover2 <- alpha/2
CIlimits <- c(alphaover2, 1-alphaover2)
```

```

par(mfrow=c(2,2))
hist(bounds$par.estimates$sigma1, main=expression(widehat(sigma[d])), xlab="Estimated value")
abline(v=quantile(bounds$par.estimates$sigma1, probs = CIlimits), lwd=2, lty=3)
hist(bounds$par.estimates$sigma2, main=expression(widehat(sigma[g])), xlab="Estimated value")
abline(v=quantile(bounds$par.estimates$sigma2, probs = CIlimits), lwd=2, lty=3)
hist(bounds$par.estimates$beta, main=expression(widehat(beta)), xlab="Estimated value")
abline(v=quantile(bounds$par.estimates$beta, probs = CIlimits), lwd=2, lty=3)
hist(bounds$par.estimates$P, main=expression(widehat(P)), xlab="Estimated value")
abline(v=quantile(bounds$par.estimates$P, probs = CIlimits), lwd=2, lty=3)

```

