`dsm` version 2 – what's new?

David L. Miller (University of Rhode Island)

RUWPA seminar, University of St Andrews

18 December 2012

outline (disclaimer)

- nothing realy “new”

outline (disclaimer)

- nothing realy “new”
- *not* my work! I just found things and added bugs

outline (disclaimer)

- nothing realy "new"
- *not* my work! I just found things and added bugs
- (in fact mostly people just told me about cool stuff)

outline (disclaimer)

- nothing realy "new"
- *not* my work! I just found things and added bugs
- (in fact mostly people just told me about cool stuff)
- `dsm` verison 2

outline (disclaimer)

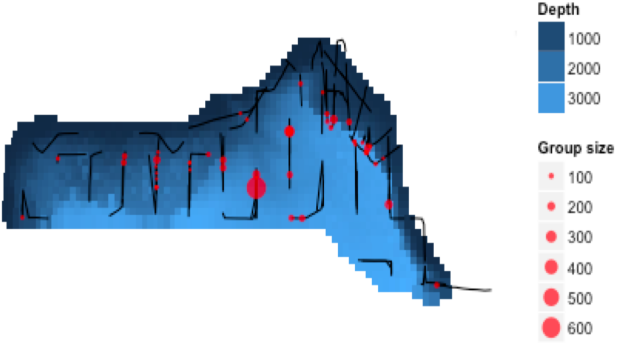
- nothing realy “new”
- *not* my work! I just found things and added bugs
- (in fact mostly people just told me about cool stuff)
- `dsm` verison 2
- variance estimation

outline (disclaimer)

- nothing realy “new”
- *not* my work! I just found things and added bugs
- (in fact mostly people just told me about cool stuff)
- `dsm` verison 2
- variance estimation
- other tricks

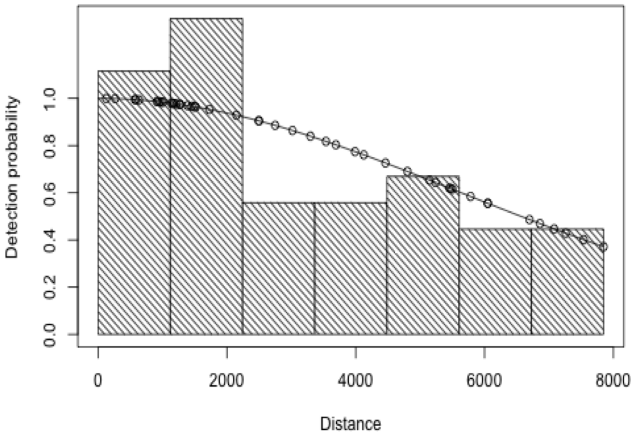
a very brief overview of density surface modelling

Collect spatially referenced data



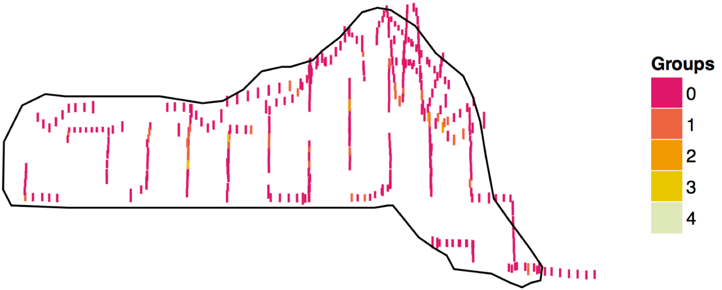
. . .

Fit detection function



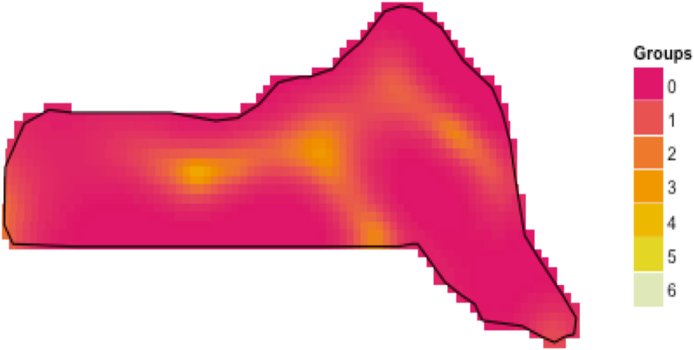
...

Aggregate to segments



. . .

Fit spatial model



dsm version 2

models look like:

```
dsm(N ~ s(x,y), ddf.obj,
    segment.data, observation.data,
    family=quasipoisson(), ...)
```

the `...` is where the magic happens

`mgcv`-like syntax

`dsm` features

- once you've fitted the model you can treat it like a `gam`

dsm features

- once you've fitted the model you can treat it like a `gam`
- `gam.check`, `predict`, `summary` all do sensible things

`dsm` features

- once you've fitted the model you can treat it like a `gam`
- `gam.check`, `predict`, `summary` all do sensible things
- also: documentation *and* examples (on CRAN)

uncertainty estimation

- want uncertainty in abundance (integral of the smooth)

uncertainty estimation

- want uncertainty in abundance (integral of the smooth)
- two sources of uncertainty:

uncertainty estimation

- want uncertainty in abundance (integral of the smooth)
- two sources of uncertainty:
  - GAM

uncertainty estimation

- want uncertainty in abundance (integral of the smooth)
- two sources of uncertainty:
  - GAM
  - detection function

uncertainty estimation

- want uncertainty in abundance (integral of the smooth)
- two sources of uncertainty:
  - GAM
  - detection function
- standard approach:

uncertainty estimation

- want uncertainty in abundance (integral of the smooth)
- two sources of uncertainty:
  - GAM
  - detection function
- standard approach:
  - moving block bootstrap for GAM

uncertainty estimation

- want uncertainty in abundance (integral of the smooth)
- two sources of uncertainty:
  - GAM
  - detection function
- standard approach:
  - moving block bootstrap for GAM
  - likelihood-based intervals for detection function

uncertainty estimation

- want uncertainty in abundance (integral of the smooth)
- two sources of uncertainty:
  - GAM
  - detection function
- standard approach:
  - moving block bootstrap for GAM
  - likelihood-based intervals for detection function
  - combine with $\text{CV}^2(\hat{N}) = \text{CV}^2(\hat{p}) + \text{CV}^2(\hat{N}_{\text{GAM}})$

uncertainty estimation issues

- bootstrapping spatial data

uncertainty estimation issues

- bootstrapping spatial data
  - autocorrelation

uncertainty estimation issues

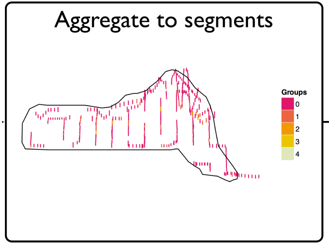
- bootstrapping spatial data
  - autocorrelation
  - is this a good idea?

uncertainty estimation issues

- bootstrapping spatial data
  - autocorrelation
  - is this a good idea?
- propagation of uncertainty from detection function

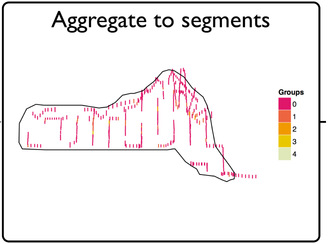
GAM confidence intervals

- let's stop using moving block bootstraps!



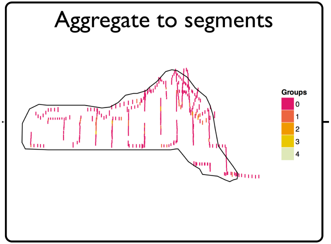
GAM confidence intervals

- let's stop using moving block bootstraps!
- using a bit of GAM theory we can find analytical results



GAM confidence intervals

- let's stop using moving block bootstraps!
- using a bit of GAM theory we can find analytical results
- not *more* assumptions but *different* ones



lpmatrix

- finding the variance is easy w. identity link.

`lpmatrix`

- finding the variance is easy w. identity link.
- use the "`lpmatrix`", $\mathbf{X}_p$ s.t. $\hat{\eta}_p = \mathbf{X}_p \hat{\beta}$

lpmatrix

- finding the variance is easy w. identity link.
- use the "lpmatrix", $\mathbf{X}_p$ s.t. $\hat{\eta}_p = \mathbf{X}_p \hat{\beta}$
- maps the model parameters to the linear predictor

`lpmatrix`

- finding the variance is easy w. identity link.
- use the "`lpmatrix`", $\mathbf{X}_p$ s.t. $\hat{\eta}_p = \mathbf{X}_p \hat{\beta}$
- maps the model parameters to the linear predictor
- so: $\mathbf{V}_{\hat{\eta}_p} = X_p \mathbf{V}_{\hat{\beta}} X_p^\mathsf{T}$

`lpmatrix`

- finding the variance is easy w. identity link.
- use the "`lpmatrix`", $\mathbf{X}_p$ s.t. $\hat{\eta}_p = \mathbf{X}_p \hat{\beta}$
- maps the model parameters to the linear predictor
- so: $\mathbf{V}_{\hat{\eta}_p} = X_p \mathbf{V}_{\hat{\beta}} X_p^\mathsf{T}$
- non-linear links more complicated...

solution by simulation

- $\beta \sim N(\hat{\beta}, \mathbf{V}_{\hat{\beta}})$

solution by simulation

- $\beta \sim N(\hat{\beta}, \mathbf{V}_{\hat{\beta}})$
- could use the following algorithm:

solution by simulation

- $\beta \sim N(\hat{\beta}, \mathbf{V}_{\hat{\beta}})$
- could use the following algorithm:
  1. For $b = 1, \ldots, N_b$ do the following:

solution by simulation

- $\beta \sim N(\hat{\beta}, \mathbf{V}_{\hat{\beta}})$
- could use the following algorithm:
  1. For $b = 1, \ldots, N_b$ do the following:
     a. Simulate from $\beta \sim N(\hat{\beta}, \mathbf{V}_{\hat{\beta}})$, to obtain β_b.

solution by simulation

- $\beta \sim N(\hat{\beta}, \mathbf{V}_{\hat{\beta}})$
- could use the following algorithm:
    1. For $b = 1, \ldots, N_b$ do the following:
        a. Simulate from $\beta \sim N(\hat{\beta}, \mathbf{V}_{\hat{\beta}})$, to obtain β_b.
        b. Calculate $\hat{\eta}_b = \exp(\mathbf{X}_p \beta_b)$ (e.g. if we are using the log-link)

solution by simulation

- $\beta \sim N(\hat{\beta}, \mathbf{V}_{\hat{\beta}})$
- could use the following algorithm:

  1. For $b = 1, \ldots, N_b$ do the following:

     a. Simulate from $\beta \sim N(\hat{\beta}, \mathbf{V}_{\hat{\beta}})$, to obtain β_b.
     b. Calculate $\hat{\eta}_b = \exp(\mathbf{X}_p \beta_b)$ (e.g. if we are using the log-link)
     c. Sum over the survey area

solution by simulation

- $\beta \sim N(\hat{\beta}, \mathbf{V}_{\hat{\beta}})$
- could use the following algorithm:

1. For $b = 1, \ldots, N_b$ do the following:
   a. Simulate from $\beta \sim N(\hat{\beta}, \mathbf{V}_{\hat{\beta}})$, to obtain β_b.
   b. Calculate $\hat{\eta}_b = \exp(\mathbf{X}_p \beta_b)$ (e.g. if we are using the log-link)
   c. Sum over the survey area
2. Calculate the appropriate summary statistics, e.g. median, 95% quantiles etc over b.

solution by simulation

- $\beta \sim N(\hat{\beta}, \mathbf{V}_{\hat{\beta}})$
- could use the following algorithm:
  1. For $b = 1, \ldots, N_b$ do the following:
     a. Simulate from $\beta \sim N(\hat{\beta}, \mathbf{V}_{\hat{\beta}})$, to obtain β_b.
     b. Calculate $\hat{\eta}_b = \exp(\mathbf{X}_p \beta_b)$ (e.g. if we are using the log-link)
     c. Sum over the survey area
  2. Calculate the appropriate summary statistics, e.g. median, 95% quantiles etc over b.
- MVB, suggests that this is unneccessary: *"once you've got a CV so big that the delta-method doesn't work, then your estimate is officially Crap and there is not much point in expending extra effort to work out exactly how Crap!"*

analytic approximate solution

beef up the variance based (roughly) on the uncertainty in the linear predictor:

$$\left(\frac{\partial^2 \log_e \eta}{\partial \eta^2} \bigg|_{\eta=\hat{\eta}} \circledast \mathbf{X}_p \right) \mathbf{V}_\beta \left(\frac{\partial^2 \log_e \eta}{\partial \eta^2} \bigg|_{\eta=\hat{\eta}} \circledast \mathbf{X}_p \right)^\mathsf{T}$$

where $\frac{\partial^2 \log_e \eta}{\partial \eta^2} \big|_{\eta=\hat{\eta}}$ is the vector of second derivatives evaluated at the values of the linear predictor

and $\circledast$ denotes R-style matrix-vector multiplication

step back

- this is nothing new

step back

- this is nothing new
- just an easier way of getting uncertainty estimates for GAMs

step back

- this is nothing new
- just an easier way of getting uncertainty estimates for GAMs
- all in Simon Wood's 2006 book

step back

- this is nothing new
- just an easier way of getting uncertainty estimates for GAMs
- all in Simon Wood's 2006 book
- we can now combine uncertainty between GAM and d.f. as usual: $\text{CV}^2(\hat{N}) = \text{CV}^2(\hat{p}) + \text{CV}^2(\hat{N}_{\text{GAM}})$

step back

- this is nothing new
- just an easier way of getting uncertainty estimates for GAMs
- all in Simon Wood's 2006 book
- we can now combine uncertainty between GAM and d.f. as usual: $\text{CV}^2(\hat{N}) = \text{CV}^2(\hat{p}) + \text{CV}^2(\hat{N}_{\text{GAM}})$
- in `dsm` as `dsm.var.gam`

variance propagation

Trick from MVB to get detection function uncertainty into the GAM

Our model is of the form:

$$\log(\mathbb{E}[N_j]) = \log(2 l_j w \hat{p}(\theta)) + \sum_k f_k(z_k)$$

add in a derivative of $\hat{p}(\theta)$:

$$\log(\mathbb{E}[N_j]) = \log(2 l_j w \hat{p}(\theta)) + \frac{d \log(\hat{p}(\theta))}{d\theta}\Big|_{\theta=\hat{\theta}} . \gamma + \sum_k f_k(z_k)$$

should have no effect on $\mathbb{E}[N_j]$

variance propagation (2)

- Detection function uncertainty now included

variance propagation (2)

- Detection function uncertainty now included
- in `dsm` as `dsm.var.prop`

variance propagation (2)

- Detection function uncertainty now included
- in `dsm` as `dsm.var.prop`
- only works when there are no covariates in d.f.

summary – uncertainty estimation

- Wood (2006) and Augustin et al. (2009) for more on uncertainty in GAMs

summary – uncertainty estimation

- Wood (2006) and Augustin et al. (2009) for more on uncertainty in GAMs
- faster than MBB, in (limited) simulations marginally “better”

summary – uncertainty estimation

- Wood (2006) and Augustin et al. (2009) for more on uncertainty in GAMs
- faster than MBB, in (limited) simulations marginally "better"
- See Williams et al. (2010) for the detection function uncertainty trick (there is an error in the paper!)

select

option `select=TRUE`

extra shrinkage allows smooths to be removed from the model

effective degrees of freedom can be shrunk to zero

```
Approximate significance of smooth terms:
                     edf Ref.df      F  p-value    
s(roughness)   1.401e+00      6  0.661   0.0531 .  
s(gchl_winter) 5.999e-07      6  0.000   0.2947    
s(depthm)      8.322e-01      6  3.651 4.83e-08 ***
s(x)           5.399e+00      6 15.571  < 2e-16 ***
s(y)           3.909e+00      6 13.928  < 2e-16 ***
```

here `gchl_winter` has been “removed” from the model

more info in `?gam`

Tweedie

- sum of M gamma variables where M is Poisson distributed

Tweedie

- sum of M gamma variables where M is Poisson distributed
- one parameter $1 < p \leq 2$

Tweedie

- sum of M gamma variables where M is Poisson distributed
- one parameter $1 < p \leq 2$
- only sensitive to changes $>$0.1, so line search feasible in most cases

Tweedie

- sum of M gamma variables where M is Poisson distributed
- one parameter $1 < p \leq 2$
- only sensitive to changes >0.1, so line search feasible in most cases
- use GCV/REML score to determine "correct" p

Tweedie

- sum of M gamma variables where M is Poisson distributed
- one parameter $1 < p \leq 2$
- only sensitive to changes >0.1, so line search feasible in most cases
- use GCV/REML score to determine "correct" p
- `?Tweedie` or Candy (2004)

finding theta for negbin()

- the theta parameter can be fixed e.g. family=negbin(theta=0.1)

finding `theta` for `negbin()`

- the `theta` parameter can be fixed e.g. `family=negbin(theta=0.1)`
- or a range can be supplied e.g. `family=negbin(theta=c(0.01,1))`

finding `theta` for `negbin()`

- the `theta` parameter can be fixed e.g. `family=negbin(theta=0.1)`
- or a range can be supplied e.g. `family=negbin(theta=c(0.01,1))`
- optimisation is by outer iteration (default) or performance iteration (`optimizer="perf"`)

finding `theta` for `negbin()`

- the `theta` parameter can be fixed e.g. `family=negbin(theta=0.1)`
- or a range can be supplied e.g. `family=negbin(theta=c(0.01,1))`
- optimisation is by outer iteration (default) or performance iteration (`optimizer="perf"`)
- [[TALK ABOUT TECHNICAL STUFF]]

finding `theta` for `negbin()`

- the `theta` parameter can be fixed e.g. `family=negbin(theta=0.1)`
- or a range can be supplied e.g. `family=negbin(theta=c(0.01,1))`
- optimisation is by outer iteration (default) or performance iteration (`optimizer="perf"`)
- [[TALK ABOUT TECHNICAL STUFF]]
- default `outer` optimisation is slower but doesn't have convergence issues

finding `theta` for `negbin()`

- the `theta` parameter can be fixed e.g. `family=negbin(theta=0.1)`
- or a range can be supplied e.g. `family=negbin(theta=c(0.01,1))`
- optimisation is by outer iteration (default) or performance iteration (`optimizer="perf"`)
- [[TALK ABOUT TECHNICAL STUFF]]
- default `outer` optimisation is slower but doesn't have convergence issues
- you can find which `theta` was used with `b$family$getTheta()`

finding `theta` for `negbin()`

- the `theta` parameter can be fixed e.g. `family=negbin(theta=0.1)`
- or a range can be supplied e.g. `family=negbin(theta=c(0.01,1))`
- optimisation is by outer iteration (default) or performance iteration (`optimizer="perf"`)
- [[TALK ABOUT TECHNICAL STUFF]]
- default `outer` optimisation is slower but doesn't have convergence issues
- you can find which `theta` was used with `b$family$getTheta()`
- `?negbin` has more information

the future!

- MMPP

the future!

- MMPP
- we only think about one way variation detection function $\Rightarrow$ GAM

the future!

- MMPP
- we only think about one way variation detection function $\Rightarrow$ GAM
- temporal inference

the future!

- MMPP
- we only think about one way variation detection function $\Rightarrow$ GAM
- temporal inference
- GAMMs? (technically possible but untested (by me at least))

the future!

- MMPP
- we only think about one way variation detection function $\Rightarrow$ GAM
- temporal inference
- GAMMs? (technically possible but untested (by me at least))
- talk available on my website http://ninepointeightone.net/about

the future!

- MMPP
- we only think about one way variation detection function $\Rightarrow$ GAM
- temporal inference
- GAMMs? (technically possible but untested (by me at least))
- talk available on my website http://ninepointeightone.net/about
- example analysis at http://github.com/dill/dsm/wiki

References

Augustin, N., Musio, M., Wilpert, von, K., Kublin, E., Wood, S.N. & Schumacher, M. (2009) Modeling spatiotemporal forest health monitoring data. Journal of the American Statistical Association, 104, 899–911.

Candy, S. (2004) Modelling catch and effort data using generalised linear models, the Tweedie distribution, random vessel effects and random stratum-by-year effects. CCAMLR Science, 11, 59–80.

Williams, R., Hedley, S.L., Branch, T.A., Bravington, M.V., Zerbini, A.N. & Findlay, K.P. (2011) Chilean blue whales as a case study to illustrate methods to estimate abundance and evaluate conservation status of rare species. Conservation Biology, 25, 526–535.

Wood, S.N. (2006) Generalized Additive Models: an Introduction with R. Chapman & Hall/CRC.