Model selection

David L Miller

What do we mean by ”model selection”?

- Term “selection”
  - Path dependence
  - Shrinkage
- Selection between models
  - Term formulation
  - Selection between structurally different models

Term selection

Term selection via p-values

- Old paradigm – select terms using p-values
- p-values are **approximate**

1. treat smoothing parameters as *known*
2. rely on asymptotic behaviour

(p-values in `summary.gam()` have changed a lot over time — all options except current default are deprecated as of `v1.18-13` (*i.e.*, ignore what's in the book!).)

Technical stuff

Test of **zero-effect** of a smooth term

Default p-values rely on theory of Nychka (1988) and Marra & Wood (2012) for confidence interval coverage.

If the Bayesian CI have good across-the-function properties, Wood (2013a) showed that the p-values have:

- almost the correct null distribution
- reasonable power

Test statistic is a form of χ^2 statistic, but with complicated degrees of freedom.

(RE)ML rant again...

Best behaviour when smoothness selection is by **ML** (**REML** also good)

Neither of these are the default, so remember to use `method = "ML"` or `method = "REML"` as appropriate

Term selection by shrinkage

Shrinkage & additional penalties

- Usually can't remove a whole function when smoothing
- Penalties used act only on *range space*
- *null space* of the basis is unpenalised.

Parts of f that don't have 2[nd] derivatives aren't penalised

$$\int_{\mathbb{R}} \left(\frac{\partial^2 f(x)}{\partial x^2} \right)^2 dx$$

(Note that penalty form depends on the basis!)

Double-penalty shrinkage

$\mathbf{S}_j$ penalty matrix j, eigendecompose:

$$\mathbf{S}_j = \mathbf{U}_j \mathbf{\Lambda}_j \mathbf{U}_j^T$$

where $\mathbf{U}_j$ is a matrix of eigenvectors and $\mathbf{\Lambda}_j$ a diagonal matrix of eigenvalues (i.e., this is an eigen decomposition of $\mathbf{S}_j$).

$\mathbf{\Lambda}_j$ contains some **0**s due to the spline basis null space — no matter how large the penalty λ_j might get no guarantee a smooth term will be suppressed completely.

Shrinkage & additional penalties

`mgcv` has two ways to penalize the null space $\Rightarrow$ term selection

- *double penalty approach* via `select = TRUE`
- *shrinkage approach* via special bases `"ts"` and `"cs"`

Marra & Wood (2011) review other options.

Double-penalty shrinkage

Create a second penalty matrix from $\mathbf{U}_j$, considering only the matrix of eigenvectors associated with the zero eigenvalues

$$\mathbf{S}_j^* = \mathbf{U}_j^* \mathbf{U}_j^{*T}$$

Now we can fit a GAM with two penalties of the form

$$\lambda_j \beta^T \mathbf{S}_j \beta + \lambda_j^* \beta^T \mathbf{S}_j^* \beta$$

In practice, add `select = TRUE` to your *gam()* call

Shrinkage

- Double penalty $\Rightarrow$ twice as many smoothing parameters
- Alternative is shrinkage, add small value to zero eigenvalues
- Null space terms to be shrunk at the same time

Use `s(..., bs = "ts")` or `s(..., bs = "cs")` in **mgcv**

Selecting between models

Model selection by REML/ML

- If you **don't** use shrinkage/double penalty you can't use REML scores to select models
- Need model to be fully penalized for REML score selection
- You can use ML though
- *but* there are other options…

AIC

AIC

- Use full likelihood of $\boldsymbol{\beta}$ *conditional* upon λ_j is used, with the EDF replacing k, the number of model parameters
- This *conditional* AIC tends to select complex models, especially those with random effects, as the EDF ignores that λ_j are estimated
- Wood et al (2015) suggests a correction that accounts for uncertainty in λ_j (`AIC`)

Okay, fine, but…

GAMs are Bayesian models

Bayesian models

- *duh*
  - we can build Bayesian GLMs
  - see also INLA and BayesX
- `mgcv` fits Bayesian models
- penalties are prior precision matrices
- (improper) Gaussian prior on $\boldsymbol{\beta}$

Empirical Bayes...?

- Improper prior derives from $\mathbf{S}_j$ not being of full rank
  - zeroes in $\mathbf{\Lambda}_j$.
- Double penalty and shrinkage smooths make prior proper
  - **Double penalty**: no assumption as to how much to shrink the null space
  - **Shrinkage smooths**: assume null space should be shrunk less than the wiggles

Practical Bayes

Marra & Wood (2011) show that the double penalty and the shrinkage smooth approaches:

- performed significantly better than alternatives in terms of *predictive ability*, and
- performed as well as alternatives in terms of variable selection

Examples

Back to the dolphins…

Fitting some models

```
dolphins_depth_xy <- gam(count ~ s(x, y) + s(depth) +
offset(off.set),
                  data = mexdolphins,
                  family=nb(), method="REML")
dolphins_depth <- gam(count ~ s(depth) + offset(off.set),
                  data = mexdolphins,
                  family=nb(), method="REML")
dolphins_xy <- gam(count ~ s(x, y) + offset(off.set),
                  data = mexdolphins,
                  family=nb(), method="REML")
```

Comparing terms by p-value

```r
summary(dolphins_depth_xy)
```

```

Family: Negative Binomial(0.027)
Link function: log

Formula:
count ~ s(x, y) + s(depth) + offset(off.set)

Parametric coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -19.7573     0.6243  -31.65   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
           edf Ref.df Chi.sq  p-value
s(x,y)   2.001  2.002  2.259    0.323
s(depth) 5.312  6.416 37.328 2.68e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  -0.0072   Deviance explained = 39.5%
-REML = 369.37  Scale est. = 1         n = 387
```

Comparing models by AIC

```
AIC(dolphins_xy, dolphins_depth, dolphins_depth_xy)
```

```
                         df      AIC
dolphins_xy        5.044883 775.7682
dolphins_depth     8.248728 744.4248
dolphins_depth_xy 10.417717 746.0289
```

Shrinkage (basis)

```
dolphins_depth_xy_s <- gam(count ~ s(x, y, bs="ts") + s(depth,
bs="ts") + offset(off.set),
                    data = mexdolphins,
                    family=nb(), method="REML")
```

Shrinkage (basis)

```
summary(dolphins_depth_xy_s)
```

```

Family: Negative Binomial(0.027)
Link function: log

Formula:
count ~ s(x, y, bs = "ts") + s(depth, bs = "ts") + offset(off.set)

Parametric coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -19.2704     0.4933  -39.06   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
             edf Ref.df Chi.sq  p-value
s(x,y)   0.02385     29  0.017    0.472
s(depth) 4.64835      9 42.845 1.91e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.0476   Deviance explained = 36.2%
-REML = 376.26  Scale est. = 1         n = 387
```

Shrinkage (extra penalty)

```
dolphins_depth_xy_e <- gam(count ~ s(x, y) + s(depth) +
offset(off.set),
                     data = mexdolphins, select=TRUE,
                     family=nb(), method="REML")
```

Shrinkage (basis)

```
summary(dolphins_depth_xy_e)
```

```

Family: Negative Binomial(0.027)
Link function: log

Formula:
count ~ s(x, y) + s(depth) + offset(off.set)

Parametric coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -19.3173     0.4539  -42.56   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
            edf Ref.df Chi.sq p-value
s(x,y)   0.1779     29  0.145   0.421
s(depth) 4.7667      9 46.659 1.5e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.0457   Deviance explained = 37.5%
-REML = 374.09  Scale est. = 1         n = 387
```

These last two models were empirical Bayes models

That’s all from the dolphins for now…