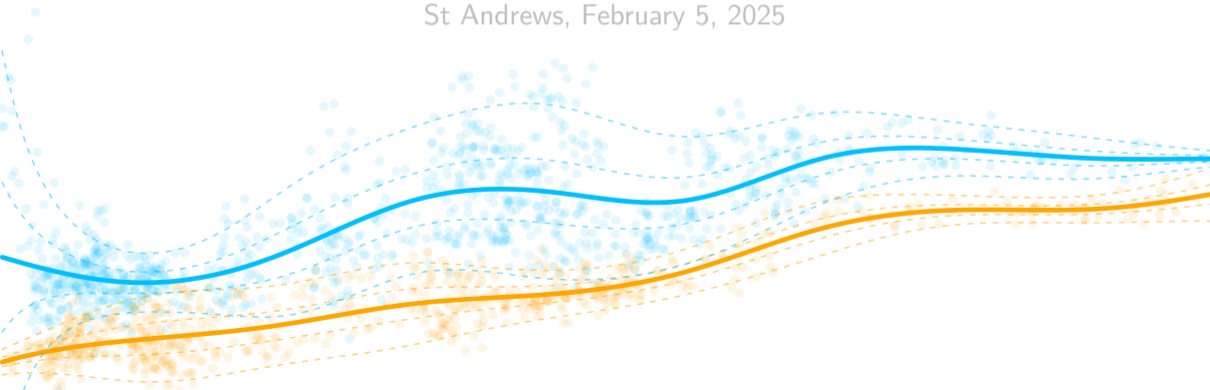


Statistics colloquium

# Efficient smoothness selection for nonparametric Markov-switching models via quasi restricted maximum likelihood

Jan-Ole Koslik

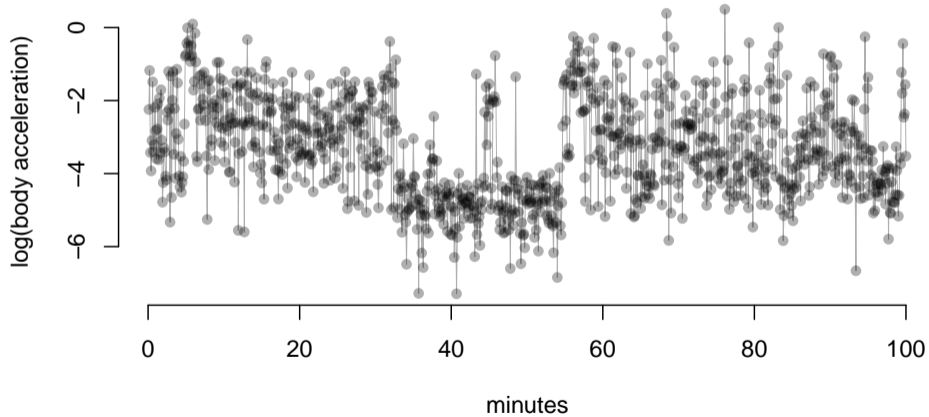
St Andrews, February 5, 2025



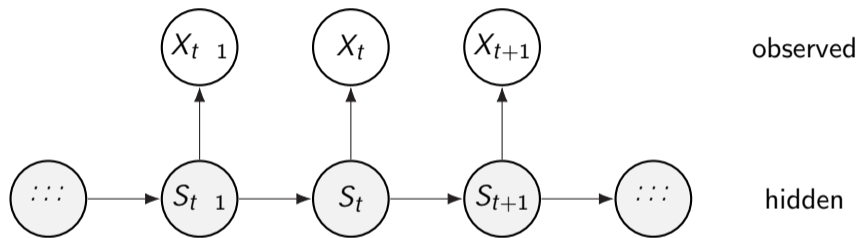


- PhD student in the **Statistics and Data Analysis Group** at Bielefeld University
- mostly working on **hidden Markov models** and their relatives

 [@olemole.bsky.social](https://bsky.app/profile/olemole.bsky.social)

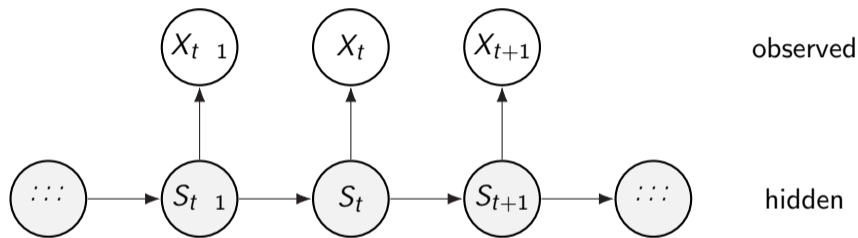


## HMM — model formulation



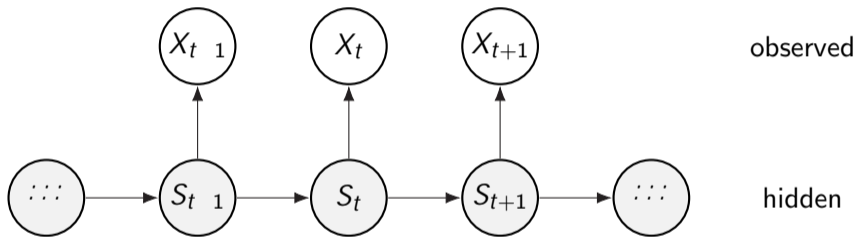
- every observation  $x_t$  is generated by one of  $N$  possible distributions  $f_1; \dots; f_N$

## HMM — model formulation

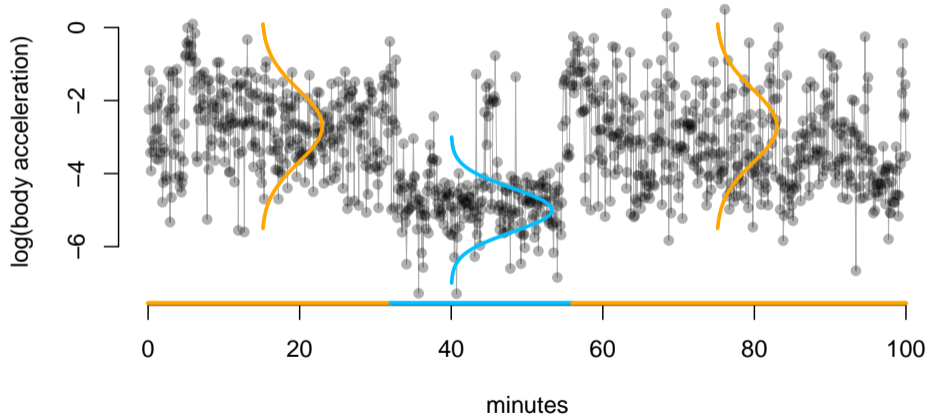


- every observation  $x_t$  is generated by one of  $N$  possible distributions  $f_1; \dots; f_N$
- hidden state process selects which distribution is active at time  $t$

## HMM — model formulation



- every observation  $x_t$  is generated by one of  $N$  possible distributions  $f_1; \dots; f_N$
- hidden state process selects which distribution is active at time  $t$
- state process is a **Markov chain**



## Reminder: Markov chains

**Markovian state process** is fully characterised by the initial distribution

$$^{(1)} = \Pr(S_1 = 1); \dots; \Pr(S_1 = N)$$

and the **transition probabilities**

$$^{(t)}_{ij} = \Pr(S_{t+1} = j \mid S_t = i);$$

which we summarise in the **transition probability matrix** (t.p.m.)

$$^{(t)} = \left( ^{(t)}_{ij} \right)_{i,j=1,\dots,N};$$



## Estimating HMMs

We can efficiently calculate the HMM likelihood using the **forward algorithm**

$$L(\lambda) = \mathbf{1}^T P(x_1) A^{(1)} P(x_2) A^{(2)} \cdots A^{(T-1)} P(x_T) \mathbf{1};$$

where  $P(x_t) = \text{diag} \{f_1(x_t), \dots, f_N(x_t)\}$ .

## Estimating HMMs

We can efficiently calculate the HMM likelihood using the **forward algorithm**

$$L(\theta) = \mathbf{1}^T P(x_1) \mathbf{1} P(x_2) \mathbf{1} P(x_3) \mathbf{1} \dots \mathbf{1}^T P(x_T) \mathbf{1};$$

where  $P(x_t) = \text{diag} [f_1(x_t); \dots; f_N(x_t)]$ .

With some adjustments, we can also calculate the **log-likelihood**  $\ell(\theta)$  to avoid numerical underflow / optimise in R using standard optimisers like `nlm()` or `optim()`.

## Estimating HMMs

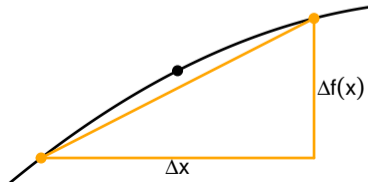
We can efficiently calculate the HMM likelihood using the **forward algorithm**

$$L(\theta) = \mathbf{1}^T P(x_1) \mathbf{1} P(x_2) \mathbf{1} P(x_3) \mathbf{1} \dots \mathbf{1} P(x_T) \mathbf{1};$$

where  $P(x_t) = \text{diag} [f_1(x_t); \dots; f_N(x_t)]$ .

With some adjustments, we can also calculate the **log-likelihood**  $\ell(\theta)$  to avoid numerical underflow ! optimise in R using standard optimisers like `nlm()` or `optim()`.

These approximate the gradient via **finite differencing**.



## Why nonparametrics?

- component distributions typically selected from **parametric** family  
/ difficult as we can't do state-specific EDA

## Why nonparametrics?

- component distributions typically selected from **parametric** family  
/ difficult as we can't do state-specific EDA
- potential covariate effects typically modelled using **linear** predictors  
/ may result in us missing interesting relationships

## Why nonparametrics?



= complicated

! often **substantial** lack of fit

## Why nonparametrics?

Misspecification will be compensated by more states but this complicates interpretation.

# **Selecting the Number of States in Hidden Markov Models: Pragmatic Solutions Illustrated Using Animal Movement**

Jennifer POHLE, Roland LANGROCK, Floris M. van BEEST, and  
Niels Martin SCHMIDT

## Why nonparametrics?

Misspecification will be compensated by more states but this complicates interpretation.

# Selecting the Number of States in Hidden Markov Models: Pragmatic Solutions Illustrated Using Animal Movement

Jennifer POHLE, Roland LANGROCK, Floris M. van BEEST, and  
Niels Martin SCHMIDT

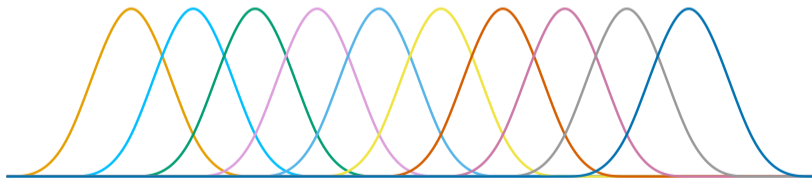
Obvious alternative: **nonparametric** approach using **penalised splines**



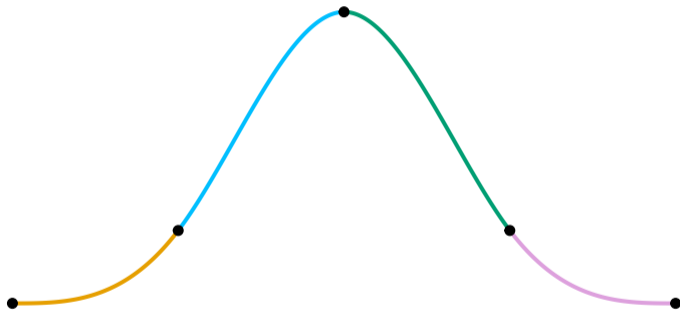
## An informal introduction to penalised splines

Idea: Perform **basis expansion** in  $x$  and represent smooth function  $s(x)$  as a linear combination of fixed basis functions  $B_k(x)$

$$s(x) = b_1 B_1(x) + b_2 B_2(x) + \dots + b_k B_k(x) = b^T B(x)$$

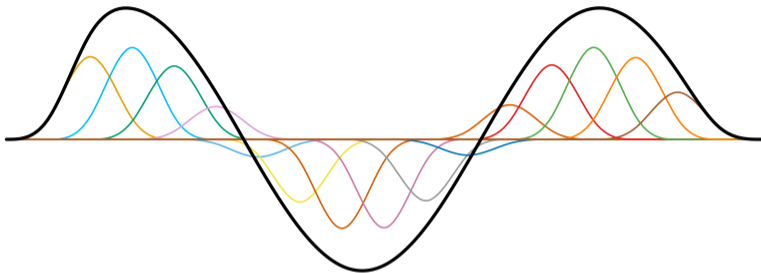


For example, when using **B-Splines**  $B_k(x)$  is a **piecewise** polynomial

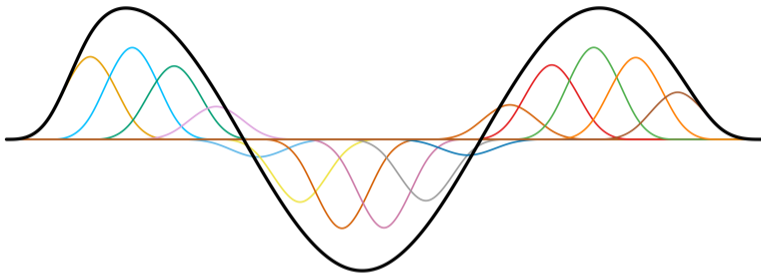


and zero outside the outer knots.

Approximate the true function with a sufficient number of basis functions:



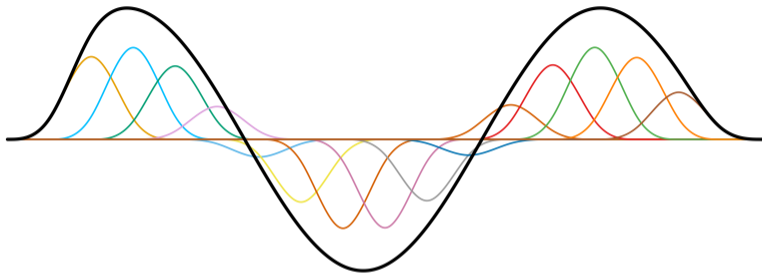
Approximate the true function with a sufficient number of basis functions:



As  $s$  should not be too **wiggly**, we add the penalty

$$\int s''(x)^2 dx$$

Approximate the true function with a sufficient number of basis functions:

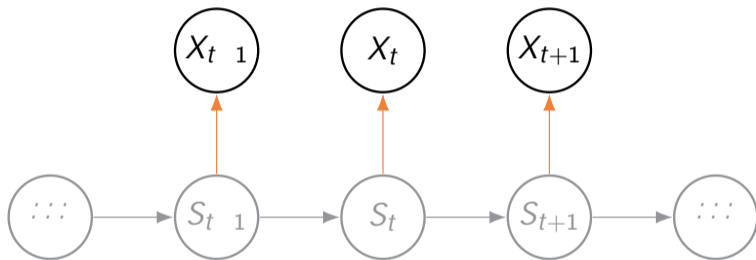


As  $s$  should not be too **wiggly**, we add the penalty

$$\int s''(x)^2 dx = b^T S b;$$

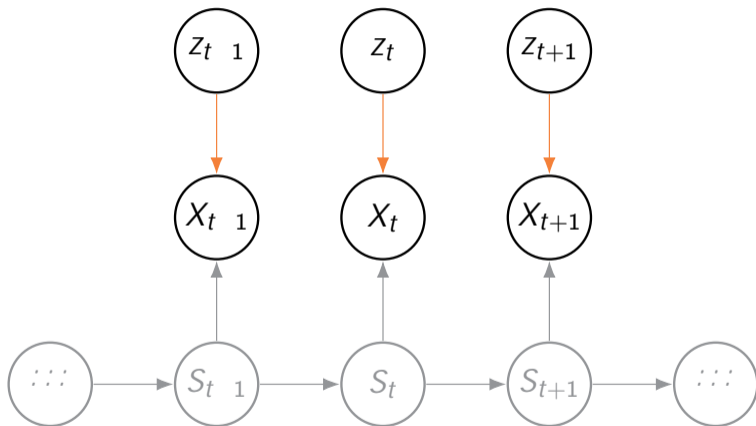
where  $S$  is fixed penalty matrix with entries  $S_{ij} = \int B_i''(x) B_j''(x) dx$ .

## Nonparametric emission distributions



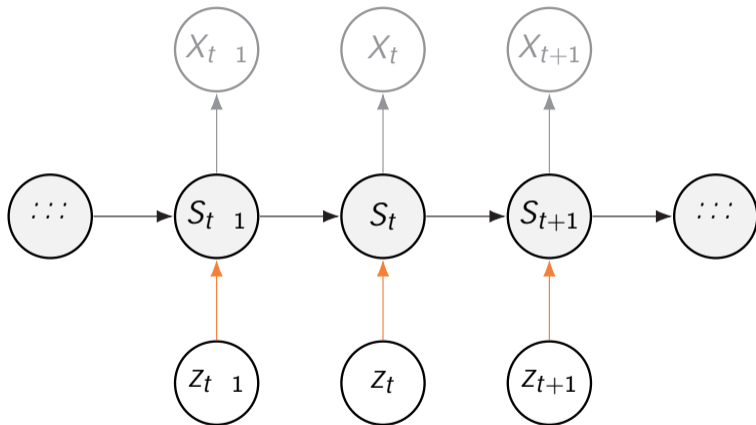
Langrock, Kneib, Sohn, and DeRuiter, 2015

## Markov-switching GAM



Langrock, Kneib, Glennie, and Michelot, 2017

## Smooth covariate effects on the state process



Feldmann et al., 2023



So this is where my spline adventure begins...

- fairly applied project:  $\binom{t}{ij}$  s(time of day, day)
- implementation with **xed** penalisation straightforward



So this is where my spline adventure begins...

- fairly applied project:  $\frac{(t)}{ij}$  s(time of day, day)
- implementation with **xed** penalisation straightforward



But how to find a good **penalty strength**??

# Smoothness selection for penalised splines

JOURNAL ARTICLE

## Nonparametric Inference in Hidden Markov Models Using P-Splines

Roland Langrock , [Thomas Kneib](#), Alexander Sohn, Stacy L. DeRuiter

*Biometrics*, Volume 71, Issue 2, June 2015, Pages 520–528,

<https://doi.org/10.1111/biom.12282>

**Published:** 13 January 2015    **Article history** ▼

## Markov-switching generalized additive models

Roland Langrock<sup>1</sup> · Thomas Kneib<sup>2</sup> · Richard Glennie<sup>3</sup> · Théo Michelot<sup>4</sup>



SPECIAL ISSUE ARTICLE |  Full Access

## Spline-based nonparametric inference in general state-switching models

Roland Langrock , Timo Adam, Vianey Leos-Barajas, Sina Mews, David L. Miller, Yannis P. Papastamatiou

## Smoothness selection for penalised splines

- cross-validation, AIC/BIC or subjective choice

## Smoothness selection for penalised splines

- cross-validation, AIC/BIC or subjective choice
- **grid search ! curse of dimensionality**
- painfully slow and (for me) under-smoothed results

## Smoothness selection for penalised splines

- cross-validation, AIC/BIC or subjective choice
- **grid search** / **curse of dimensionality**
- painfully slow and (for me) under-smoothed results
- in general, underdeveloped, most work done for GLMMs and GAMs

## Smoothness selection for penalised splines

- cross-validation, AIC/BIC or subjective choice
- **grid search** / **curse of dimensionality**
- painfully slow and (for me) under-smoothed results
- in general, underdeveloped, most work done for GLMMs and GAMs

Ideally: define the **penalised log-likelihood** / **automatic** smoothness-selection

# Smoothness selection for penalised splines

Comput Stat (2012) 27:757–777  
DOI 10.1007/s00180-011-0289-6

---

ORIGINAL PAPER

## **Density estimation and comparison with a penalized mixture approach**

**Christian Schellhase · Göran Kauermann**

- splines as **random effects**?



# Smoothness selection for penalised splines

Comput Stat (2012) 27:757–777  
DOI 10.1007/s00180-011-0289-6

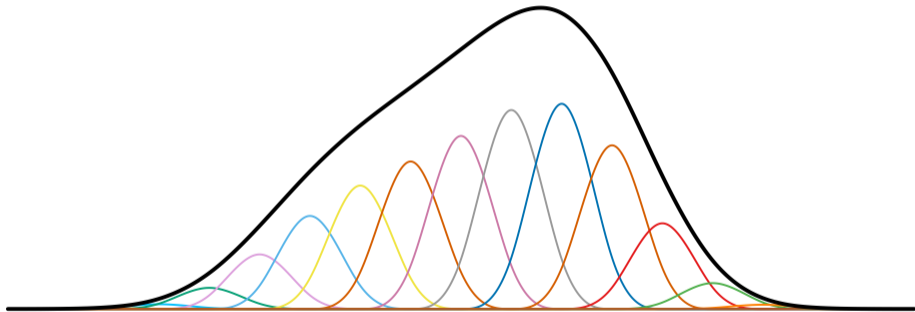
---

ORIGINAL PAPER

## **Density estimation and comparison with a penalized mixture approach**

**Christian Schellhase · Göran Kauermann**

- splines as **random effects**?
- Laplace approximation?



## Why are splines random effects?

Simple setting involving one penalised spline and no unpenalised parameters:

$$\eta_p(b; \lambda) = \eta(b) + \frac{1}{2} b^T S b;$$

- coefficients  $b = (b_1; \dots; b_k)$
- fixed penalty matrix  $S$
- penalty strength

## Why are splines random effects?

Simple setting involving one penalised spline and no unpenalised parameters:

$$\eta_p(b; \lambda) = \eta(b) - \frac{1}{2} b^T S b;$$

- coefficients  $b = (b_1; \dots; b_k)$
- fixed penalty matrix  $S$
- penalty strength

Penalised likelihood function

$$L_p(b; \lambda) = L(b) \exp \left\{ -\frac{1}{2} b^T S b \right\}$$

We see that

$$\exp \left\{ -\frac{1}{2} b^T S b \right\}$$

is proportional to a **multivariate Gaussian density**.

---

<sup>1</sup>ignoring potential issues with invertibility here

We see that

$$\exp \left\{ -\frac{1}{2} b^T S b \right\}$$

is proportional to a **multivariate Gaussian density**.

We might as well assume  $b \sim N(\mathbf{0}; S^{-1} = \Sigma)$  and add the missing **normalisation constant**<sup>1</sup>

$$L_j(b; \Sigma) = L(b) \frac{1}{(2\pi)^{k/2} \det(\Sigma)^{1/2}} \exp \left\{ -\frac{1}{2} b^T S b \right\}$$

---

<sup>1</sup>ignoring potential issues with invertibility here

We see that

$$\exp \left\{ -\frac{1}{2} \mathbf{b}^T \mathbf{S} \mathbf{b} \right\}$$

is proportional to a multivariate Gaussian density.

We might as well assume  $\mathbf{b} \sim N(\mathbf{0}; \mathbf{S}^{-1})$  and add the missing **normalisation constant**<sup>1</sup>

$$L_j(\mathbf{b}; \boldsymbol{\theta}) = L(\mathbf{b}) \left( \frac{1}{2\pi} \right)^{k/2} \det(\mathbf{S})^{1/2} \exp \left\{ -\frac{1}{2} \mathbf{b}^T \mathbf{S} \mathbf{b} \right\}$$

giving the joint log-likelihood

$$\ell_j(\mathbf{b}; \boldsymbol{\theta}) = \ell(\mathbf{b}) - \frac{k}{2} \log(2\pi) - \frac{1}{2} \log \det(\mathbf{S}) - \frac{1}{2} \mathbf{b}^T \mathbf{S} \mathbf{b}$$

---

<sup>1</sup>ignoring potential issues with invertibility here

## How to estimate models with random effects?

Joint likelihood of the **data** and the **random effect** as a function of

$$f(\mathbf{x}; \mathbf{b}) = f(\mathbf{x} | \mathbf{b}) f(\mathbf{b});$$

with  $f(\mathbf{x} | \mathbf{b}) = L(\mathbf{b})$  and  $f(\mathbf{b}) \sim N(0; \mathbf{S}^{-1})$ .



## How to estimate models with random effects?

Joint likelihood of the **data** and the **random effect** as a function of

$$f(\mathbf{x}; \mathbf{b}) = f(\mathbf{x} | \mathbf{b}) f(\mathbf{b});$$

with  $f(\mathbf{x} | \mathbf{b}) = L(\mathbf{b})$  and  $f(\mathbf{b}) \sim N(0; \Sigma^{-1})$ .

Marginal likelihood of the data by law of total probability

$$L(\mathbf{x}) = f(\mathbf{x}) = \int f(\mathbf{x}; \mathbf{b}) d\mathbf{b};$$

## How to estimate models with random effects?

Joint likelihood of the **data** and the **random effect** as a function of

$$f(\mathbf{x}; \mathbf{b}) = f(\mathbf{x} | \mathbf{b}) f(\mathbf{b});$$

with  $f(\mathbf{x} | \mathbf{b}) = L(\mathbf{b})$  and  $f(\mathbf{b}) \sim N(0; \Sigma^{-1})$ .

Marginal likelihood of the data by law of total probability

$$L(\mathbf{x}) = f(\mathbf{x}) = \int f(\mathbf{x}; \mathbf{b}) d\mathbf{b};$$

which we would like to maximise to find the MLE  $\hat{\mathbf{b}}$ .

# Marginal ML

# Marginal ML

## Marginal ML

In reality, this integral is intractable!

What can we do?

What can we do?

## The Laplace approximation

- ^ find the **mode** (in  $b$ ) of the joint log-likelihood

$$\ell_j(b; \mathbf{y}) = \frac{k}{2} \log(2\pi) + \frac{1}{2} \log \det(\mathbf{S}) - \frac{1}{2} \mathbf{b}^T \mathbf{S} \mathbf{b}$$

by penalised ML



## The Laplace approximation

- ^ find the **mode** (in  $b$ ) of the joint log-likelihood

$$\ell_j(b; \theta) = \frac{k}{2} \log(2\pi) + \frac{1}{2} \log \det(-S) - \frac{1}{2} (b - \hat{b})^T S (b - \hat{b})$$

by penalised ML

- ^ second-order **Taylor approximation** around the **mode**:

$$\ell_{\text{approx}}(b; \theta) = \ell_j(\hat{b}; \theta) - \frac{1}{2} (b - \hat{b})^T J(\theta)(b - \hat{b});$$

where  $J(\theta) = -\nabla^2 \ell_j(\hat{b}; \theta)$  is the (negative) Hessian of  $\ell_j(b; \theta)$  w.r.t.  $b$  at  $\hat{b}(\theta)$ .



^ now exponentiate to obtain likelihood and integrate out

$$\exp\{-j(\hat{b}; \cdot)\} = \int \exp\left\{-\frac{1}{2}(b - \hat{b})^T J(\cdot)(b - \hat{b})\right\} db$$

^ now exponentiate to obtain likelihood and integrate out

$$\exp\{-j(\hat{b}; y)\} = \int \exp\left\{-\frac{1}{2}(b - \hat{b})^T J(b - \hat{b})\right\} db$$

^ right side is a Gaussian integral and equals

$$(2\pi)^{k/2} |\det(J)|^{-1/2}$$

^ now exponentiate to obtain likelihood and integrate out

$$\exp \{ \eta_j(\hat{\mathbf{b}}; \boldsymbol{\beta}) \} = \int \exp \left\{ -\frac{1}{2} (\mathbf{b} - \hat{\mathbf{b}})' \mathbf{J}(\boldsymbol{\beta}) (\mathbf{b} - \hat{\mathbf{b}}) \right\} d\mathbf{b}$$

^ right side is a Gaussian integral and equals

$$(2\pi)^{k/2} |\det(\mathbf{J}(\boldsymbol{\beta}))|^{-1/2}$$

^ hence, approximate marginal log-likelihood becomes

$$\eta(\boldsymbol{\beta}) = \eta_j(\hat{\mathbf{b}}; \boldsymbol{\beta}) + \frac{k}{2} \log(2\pi) - \frac{1}{2} \log \det(\mathbf{J}(\boldsymbol{\beta}))$$





^ typically good approximation because large values dominate the integral



- ^ typically good approximation because large values dominate the integral
- ^ shape of the likelihood becomes more and more Gaussian as  $n \rightarrow \infty$

- ^ typically good approximation because large values dominate the integral
- ^ shape of the likelihood becomes more and more Gaussian as  $n \rightarrow \infty$
- ^ but each evaluation of marginal log-likelihood requires inner optimisation w.r.t.  $\theta$  ! leads to nested optimisation in general

## Optimising the marginal likelihood

Schellhase and Kauermann (2012) start with the (approximate) marginal log-likelihood

$$\ell(\beta) \approx \ell_j(\hat{\beta}; \beta) + \frac{k}{2} \log(2\pi) - \frac{1}{2} \log \det J(\beta) :$$

## Optimising the marginal likelihood

Schellhase and Kauermann (2012) start with the (approximate) marginal log-likelihood

$$\ell(\boldsymbol{\beta}) = \ell_j(\hat{\boldsymbol{\beta}}; \boldsymbol{\beta}) + \frac{k}{2} \log(2\pi) - \frac{1}{2} \log \det \mathbf{J}(\boldsymbol{\beta}) :$$

Writing out our joint log-likelihood, we have as our marginal log-likelihood

$$\ell(\boldsymbol{\beta}) = \frac{k}{2} \log(2\pi) + \frac{1}{2} \log \det(\mathbf{S}) + \ell(\hat{\boldsymbol{\beta}}) - \frac{1}{2} \hat{\boldsymbol{\beta}}^T \mathbf{S} \hat{\boldsymbol{\beta}} + \frac{k}{2} \log(2\pi) - \frac{1}{2} \log \det \mathbf{J}(\boldsymbol{\beta}) ;$$

## Optimising the marginal likelihood

Schellhase and Kauermann (2012) start with the (approximate) marginal log-likelihood

$$\ell(\boldsymbol{\beta}) = \ell_j(\hat{\boldsymbol{\beta}}; \boldsymbol{\beta}) + \frac{k}{2} \log(2\pi) - \frac{1}{2} \log \det \mathbf{J}(\boldsymbol{\beta}) :$$

Writing out our joint log-likelihood, we have as our marginal log-likelihood

$$\ell(\boldsymbol{\beta}) = \frac{k}{2} \log(2\pi) + \frac{1}{2} \log \det(\mathbf{S}) + \ell(\hat{\boldsymbol{\beta}}) - \frac{1}{2} \hat{\boldsymbol{\beta}}^T \mathbf{S} \hat{\boldsymbol{\beta}} + \frac{k}{2} \log(2\pi) - \frac{1}{2} \log \det \mathbf{J}(\boldsymbol{\beta}) ;$$

which we can now (partially) differentiate w.r.t.

## Optimising the marginal likelihood

$$\frac{\partial}{\partial \alpha} \frac{1}{2} \log \det(S) = \frac{\partial}{\partial \alpha} \frac{1}{2} \log \alpha^k \det(S) = \frac{\partial}{\partial \alpha} \frac{k}{2} \log(\alpha) + \frac{1}{2} \log \det(S) = \frac{k}{2}$$

$$\frac{\partial}{\partial \beta} \frac{1}{2} \beta^T S \beta = \frac{1}{2} \beta^T S \beta$$

$$\frac{\partial}{\partial \gamma} \frac{1}{2} \log \det J(\gamma) = \frac{1}{2} \text{tr} J(\gamma)^{-1} S$$

## Optimising the marginal likelihood

Hence in total

$$\frac{\partial \ell(\beta)}{\partial \beta} = \frac{1}{2} \mathbf{1}^T \mathbf{S} \hat{\beta} + \frac{k}{2} \frac{1}{2} \text{tr}(\mathbf{J}(\beta))^{-1} \mathbf{S} ;$$

## Optimising the marginal likelihood

Hence in total

$$\frac{\partial \ell(\beta)}{\partial \beta} = \frac{1}{2} \hat{\beta}^T S \hat{\beta} + \frac{k}{2} \frac{1}{2} \text{tr} [J(\beta)^{-1} S];$$

from which we can construct the estimating equation (omitting the details)

$$= \frac{\text{tr} [J(\beta)^{-1} J(\beta) = 0]}{\hat{\beta}^T S \hat{\beta}} = \frac{\text{dof}(\beta)}{\hat{\beta}^T S \hat{\beta}}.$$



## Optimising the marginal likelihood

Hence in total

$$\frac{\partial \ell(\beta)}{\partial \beta} = \frac{1}{2} \hat{\beta}^T S \hat{\beta} + \frac{k}{2} = \frac{1}{2} \text{tr} \{ J(\beta)^{-1} S \};$$

from which we can construct the estimating equation (omitting the details)

$$= \frac{\text{tr} \{ J(\beta)^{-1} J(\beta) \}}{\hat{\beta}^T S \hat{\beta}} = \frac{\text{dof}(\beta)}{\hat{\beta}^T S \hat{\beta}};$$

which yields the iterative procedure:

1. fit model via penalised ML
2. calculate Hessian at optimum
3. update penalty strength
4. repeat 1.-3. until convergence

- ^ 1st try ! worked much better than CV or AIC/ BIC, requiring only a few iterations for convergence ( 20 compared to hundreds or thousands for grid search)

- ^ first try ! worked much better than CV or AIC/ BIC, requiring only a few iterations for convergence ( 20 compared to hundreds or thousands for grid search)
- ^ but my situation is more complicated:
  - nested effects
  - multiple penalties

- ^ first try ! worked much better than CV or AIC/ BIC, requiring only a few iterations for convergence ( 20 compared to hundreds or thousands for grid search)
- ^ but my situation is more complicated:
  - fixed effects
  - multiple penalties
- ^ So why did it work??

## The full setting

$$p(\mathbf{a}; \mathbf{b}; \mathbf{S}) = p(\mathbf{a}; \mathbf{b}) \prod_i \frac{1}{\sqrt{2}} \exp\left(-\frac{1}{2} \mathbf{b}_i^T \mathbf{S} \mathbf{b}_i\right)$$

- ^ fixed effects  $\mathbf{a}$
- ^ multiple random effects  $\mathbf{b} = (\mathbf{b}_1; \dots; \mathbf{b}_p)$

$$\mathbf{b}_i \sim N(0; \mathbf{S}^{-1} \mathbf{1}_i)$$

Problem: If we integrate out  $\mathbf{b}$ , marginal likelihood is more complicated because of

## The full setting

$$p(\mathbf{a}; \mathbf{b}; \mathbf{X}) = p(\mathbf{a}; \mathbf{b}) \prod_i \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2} \mathbf{b}_i^T \mathbf{S} \mathbf{b}_i\right)$$

^ fixed effects  $\mathbf{a}$

^ multiple random effects  $\mathbf{b} = (b_1; \dots; b_p)$

$$b_i \sim N(0; \mathbf{S}^{-1} = \Sigma_i)$$

Problem: If we integrate out  $\mathbf{b}$ , marginal likelihood is more complicated because of

Solution: Assume  $a \sim N(0; 1)$  and integrate out both  $a$  and  $\mathbf{b}$ ! restricted maximum likelihood (REML), Laird and Ware, 1982

## Quasi REML

Then, marginal log-likelihood is a function of  $\beta = (\beta_1; \dots; \beta_p)$ .

## Quasi REML

Then, marginal log-likelihood is a function of  $(\beta_1; \dots; \beta_p)$ .

Partially differentiating w.r.t.  $\beta_i$  yields very similar result:

$$i = \frac{\text{tr} \left[ J(\hat{\beta})^{-1} J(\beta_i = 0) \right]_{ii}}{\hat{\beta}_i' S \hat{\beta}_i} = \frac{\text{dof}(\beta_i)}{\hat{\beta}_i' S \hat{\beta}_i}$$



## Quasi REML

Then, marginal log-likelihood is a function of  $\beta = (\beta_1; \dots; \beta_p)$ .

Partially differentiating w.r.t.  $\beta_i$  yields very similar result:

$$\beta_i = \frac{\text{tr} \left( J(\hat{\beta})^{-1} J(\beta_i = 0) \right)_{ii}}{\hat{\beta}_i^T S \hat{\beta}_i} = \frac{\text{dof}(\beta_i)}{\hat{\beta}_i^T S \hat{\beta}_i}$$

Yields the same iterative procedure: model fitting via penalised ML and updating penalty strength.

## Quasi REML

Then, marginal log-likelihood is a function of  $\beta = (\beta_1; \dots; \beta_p)$ .

Partially differentiating w.r.t.  $\beta_i$  yields very similar result:

$$\beta_i = \frac{\text{tr} \left( J(\hat{\beta})^{-1} J(\hat{\beta} = 0)_{ii} \right)}{\hat{\beta}_i^T S \hat{\beta}_i} = \frac{\text{dof}(\beta_i)}{\hat{\beta}_i^T S \hat{\beta}_i}$$

Yields the same iterative procedure: model fitting via penalised ML and updating penalty strength.

! smoothness selection procedure that makes nonparametric HMMs feasible!

## RTMB enters the picture

Here comes RTMB (Kristensen, 2024) with automatic differentiation, natively supporting the full Laplace approximation for models written in `plR` code

RTM enters the picture

So did we gain anything?

## RTM enters the picture

So did we gain anything?

- ^ with RTM (for inner optimisation), possible to implement `qreml()` very generally

## RTM enters the picture

So did we gain anything?

- ^ with RTM (for inner optimisation), possible to implement `qreml()` very generally
- ^ user only needs to specify penalised negative log-likelihood

## RTM enters the picture

So did we gain anything?

- ^ with RTM (for inner optimisation), possible to implement `qreml()` very generally
- ^ user only needs to specify penalised negative log-likelihood
- ^ qREML + AD ! efficiency skyrocketed!

## Practical usage

```
library(LaMa)
```

```
pnll <- function(par){ # penalised negative log-likelihood  
  ... # computing the negative log-likelihood  
  nll + penalty(splinePars, S, lambda)  
}
```



## Practical usage

```
library(LaMa)
```

```
pnll <- function(par){ # penalised negative log-likelihood
  ... # computing the negative log-likelihood
  nll + penalty(splinePars, S, lambda)
}
```

```
par <- list(..., splinePars = matrix(0, 2, 10)) # parameter list
dat <- list(..., S = S, lambda = rep(100, 2)) # data list
```

## Practical usage

```
library(LaMa)
```

```
pnll <- function(par){ # penalised negative log-likelihood
  ... # computing the negative log-likelihood
  nll + penalty(splinePars, S, lambda)
}
```

```
par <- list(..., splinePars = matrix(0, 2, 10)) # parameter list
dat <- list(..., S = S, lambda = rep(100, 2)) # data list
```

```
mod <- qreml(pnll, par, dat, random = "splinePars")
```

## Real-data example

## Bull sharks (Byrnes et al., 2023)

- Western Australia, extreme seasonal changes
  - seven bull sharks tagged (14K observations)
  - temperature, depth, and acceleration data
  - response: overall dynamic body acceleration
  - 2-state HMM: **low** and **high activity**
-

## Bull sharks (Byrnes et al., 2023)

- Western Australia, extreme seasonal changes
  - seven bull sharks tagged (14K observations)
  - temperature, depth, and acceleration data
  - response: overall dynamic body acceleration
  - 2-state HMM: **low** and **high activity**
- 

- $ODBA_k^{(t)} = \mu_k + s_k(\text{time}_t); k = 1; \dots; 7$  (ectotherms)

## Bull sharks (Byrnes et al., 2023)

- Western Australia, extreme seasonal changes
  - seven bull sharks tagged (14K observations)
  - temperature, depth, and acceleration data
  - response: overall dynamic body acceleration
  - 2-state HMM: **low** and **high activity**
- 

- $ODBA_k^{(t)} = \mu_k + s_k(\text{time}_t); k = 1; \dots; 7$  (ectotherms)
- $s_{ij}^{(t)} = \text{AvgTemp}_t + s(\text{tod}_t)$  (parametric in original paper)

## Bull sharks (Byrnes et al., 2023)

- Western Australia, extreme seasonal changes
  - seven bull sharks tagged (14K observations)
  - temperature, depth, and acceleration data
  - response: overall dynamic body acceleration
  - 2-state HMM: **low** and **high activity**
- 

- $ODBA_k^{(t)} = \mu_k + s_k(\text{time}_t); k = 1; \dots; 7$  (ectotherms)
- $s_{ij}^{(t)} = \text{AvgTemp}_t + s(\text{tod}_t)$  (parametric in original paper)
- **11** smooths in total, **162** parameters

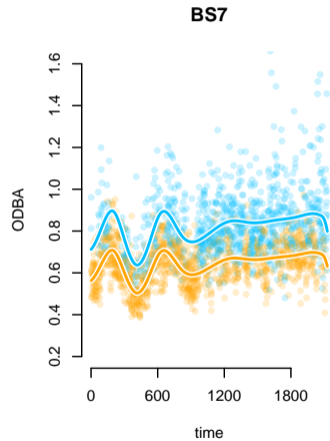
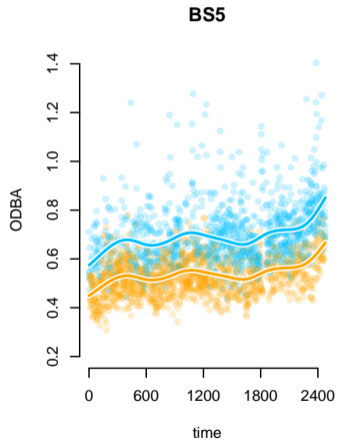
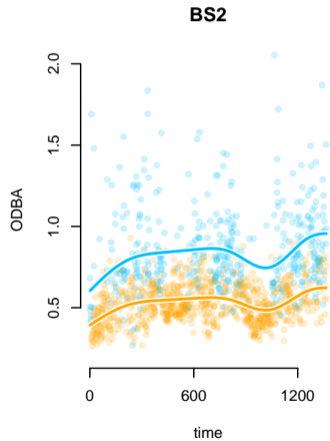
## Bull sharks (Byrnes et al., 2023)

- Western Australia, extreme seasonal changes
  - seven bull sharks tagged (14K observations)
  - temperature, depth, and acceleration data
  - response: overall dynamic body acceleration
  - 2-state HMM: **low** and **high activity**
- 

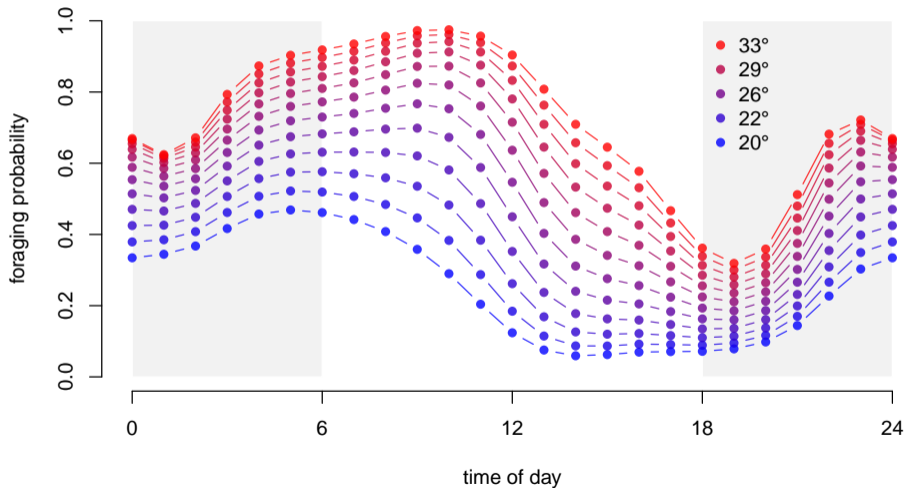
- $ODBA_k^{(t)} = \mu_k + s_k(\text{time}_t); k = 1; \dots; 7$  (ectotherms)
- $s_{ij}^{(t)} = s(\text{tod}_t) + \text{AvgTemp}_t - s(\text{tod}_t)$  (parametric in original paper)
- **11** smooths in total, **162** parameters
- model fit takes 5 minutes (32 penalised fits until convergence)



# Bull sharks (Byrnes et al., 2023)



## Bull sharks (Byrnes et al., 2023)



## Concluding remarks

- **qREML** is (of course) much more generally applicable than just to HMMs

## Concluding remarks

- **qREML** is (of course) much more generally applicable than just to HMMs
- you can find `qreml ()` in the package `LaMa`, vignette “*Penalised splines*”

## Concluding remarks

- **qREML** is (of course) much more generally applicable than just to HMMs
- you can find `qreml ()` in the package `LaMa`, vignette “*Penalised splines*”
- models with i.i.d. random effects can be fitted using the same approach

## Concluding remarks

- **qREML** is (of course) much more generally applicable than just to HMMs
  - you can find `qreml ()` in the package `LaMa`, vignette “*Penalised splines*”
  - models with i.i.d. random effects can be fitted using the same approach
- 
- `RTMB` will become an extremely valuable tool for fitting complex models

