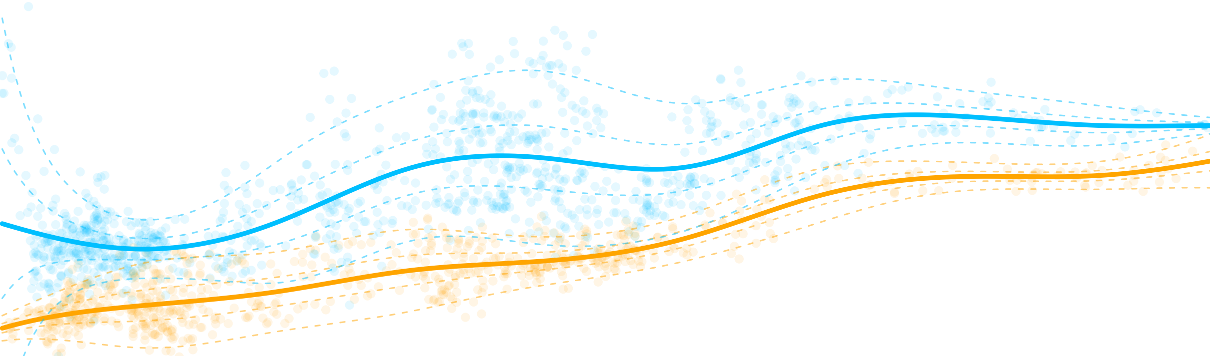
Statistics colloquium

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

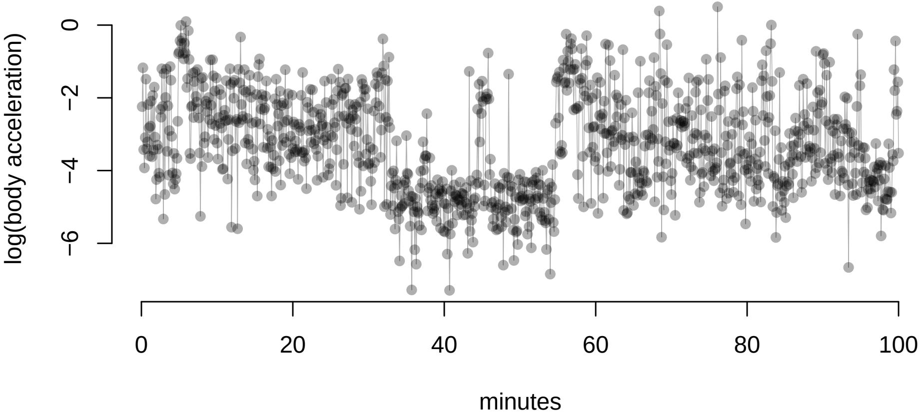
Jan-Ole Koslik

St Andrews, February 5, 2025

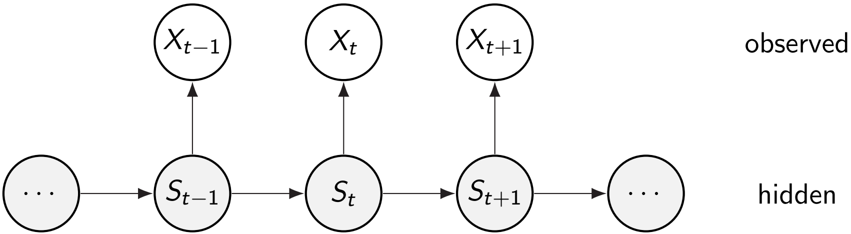


- PhD student in the **Statistics and Data Analysis Group** at Bielefeld University
- mostly working on **hidden Markov models** and their relatives

@olemole.bsky.social

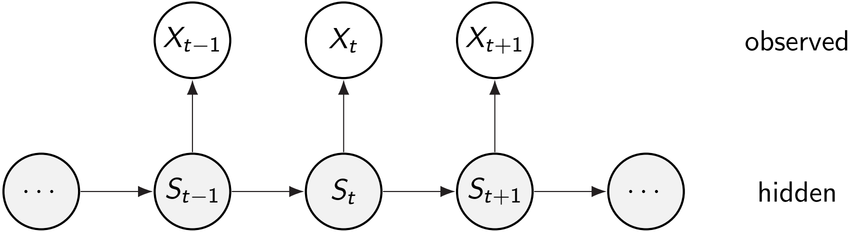

log(body acceleration)
0
−2
−4
−6
0
20
40
60
80
100
minutes

HMM — model formulation



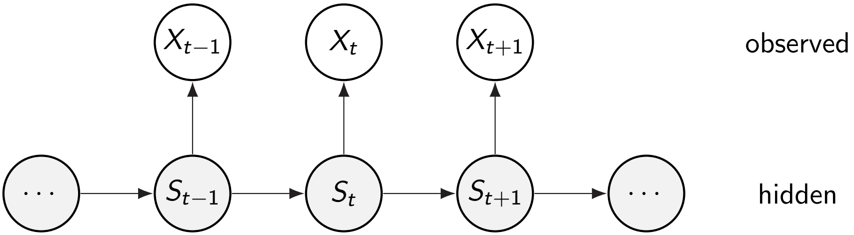
- every observation x_t is generated by one of N possible distributions $f_1, \ldots, f_N$

HMM — model formulation

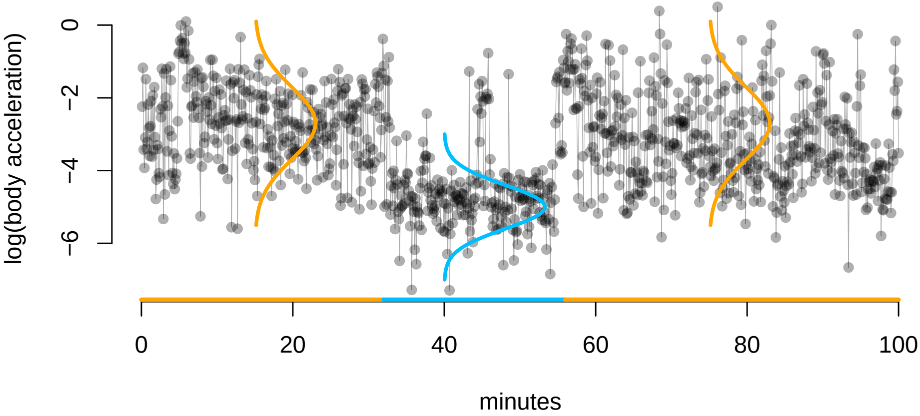


- every observation x_t is generated by one of N possible distributions $f_1, \ldots, f_N$
- hidden state process selects which distribution is active at time t

HMM — model formulation



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


log(body acceleration)
0
−2
−4
−6
0
20
40
60
80
100
minutes

Reminder: Markov chains

Markovian state process is fully characterised by the initial distribution

$$\boldsymbol{\delta}^{(1)} = \big(\Pr(S_1 = 1), \ldots, \Pr(S_1 = N)\big)$$

and the **transition probabilities**

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

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

$$\boldsymbol{\Gamma}^{(t)} = (\gamma_{ij}^{(t)})_{i,j=1,\ldots,N}.$$

Estimating HMMs

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

$$\mathcal{L}(\boldsymbol{\theta}) = \boldsymbol{\delta}^{(1)} \boldsymbol{P}(x_1) \boldsymbol{\Gamma}^{(1)} \boldsymbol{P}(x_2) \boldsymbol{\Gamma}^{(2)} \cdot \ldots \cdot \boldsymbol{\Gamma}^{(T-1)} \boldsymbol{P}(x_T) \mathbf{1},$$

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

Estimating HMMs

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

$$\mathcal{L}(\boldsymbol{\theta}) = \boldsymbol{\delta}^{(1)} \boldsymbol{P}(x_1) \boldsymbol{\Gamma}^{(1)} \boldsymbol{P}(x_2) \boldsymbol{\Gamma}^{(2)} \cdot \ldots \cdot \boldsymbol{\Gamma}^{(T-1)} \boldsymbol{P}(x_T) \mathbf{1},$$

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

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

Estimating HMMs

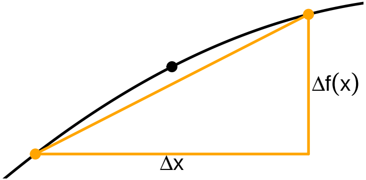
We can efficiently calculate the HMM likelihood using the **forward algorithm**

$$\mathcal{L}(\boldsymbol{\theta}) = \boldsymbol{\delta}^{(1)} \boldsymbol{P}(x_1) \boldsymbol{\Gamma}^{(1)} \boldsymbol{P}(x_2) \boldsymbol{\Gamma}^{(2)} \cdot \ldots \cdot \boldsymbol{\Gamma}^{(T-1)} \boldsymbol{P}(x_T) \mathbf{1},$$

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

With some adjustments, we can also calculate the **log-likelihood** $\ell(\boldsymbol{\theta})$ to avoid numerical underflow $\rightarrow$ 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
  $\rightarrow$ difficult as we can't do state-specific EDA

Why nonparametrics?

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

Why nonparametrics?

$=$ complicated

$\rightarrow$ 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

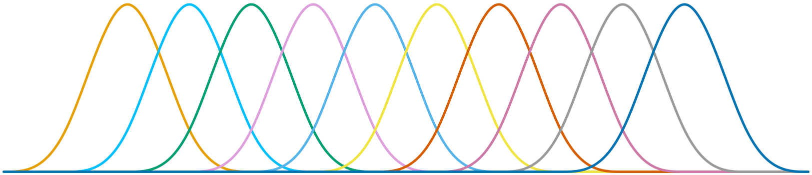
Jennifer Pohle, Roland Langrock, Floris M. van Beest, and Niels Martin Schmidt

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

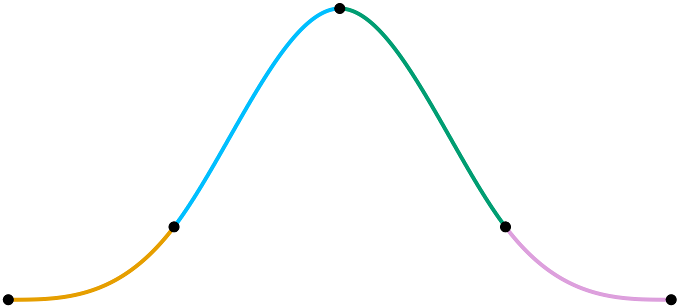
An informal introduction to penalised splines

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

$$s(x) = b_1 B_1(x) + b_2 B_2(x) + \ldots + b_k B_k(x) = \boldsymbol{b}^\top \boldsymbol{B}(x)$$

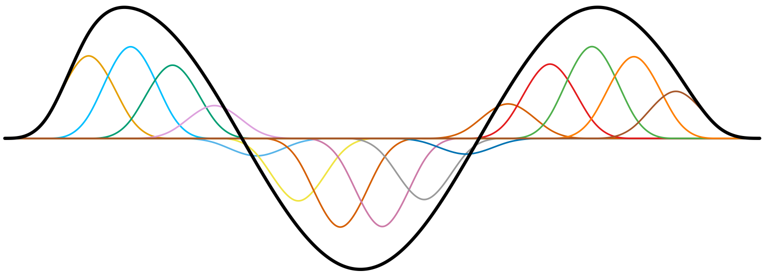


For example, when using **B-Splines** $B_k(x)$ is a **piecewise** polynomial

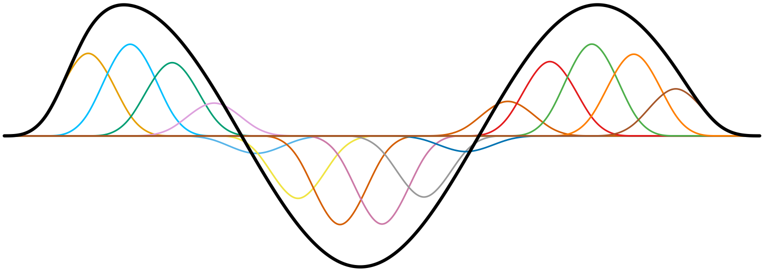


and zero outside the outer knots.

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



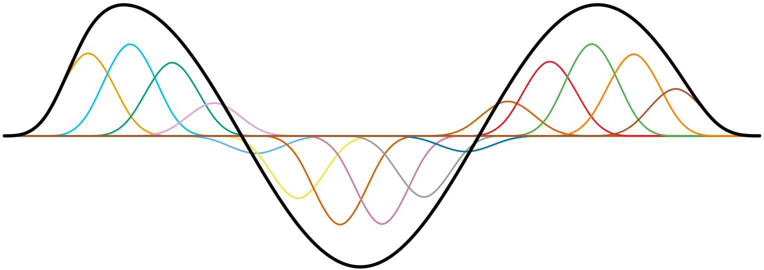
Approximate the true function with a sufficient number of basis functions:



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

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

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

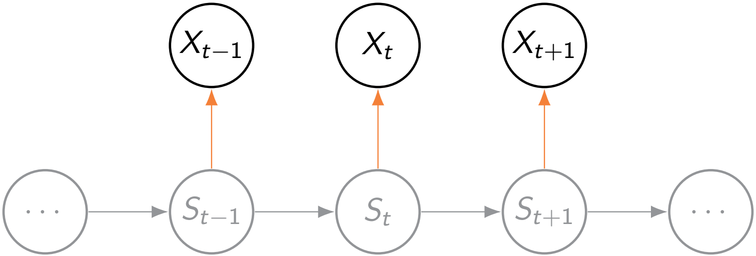


As s should not be too **wiggly**, we add the penalty

$$\lambda \int s''(x)^2 \, dx = \lambda \boldsymbol{b}^\top \boldsymbol{S} \boldsymbol{b},$$

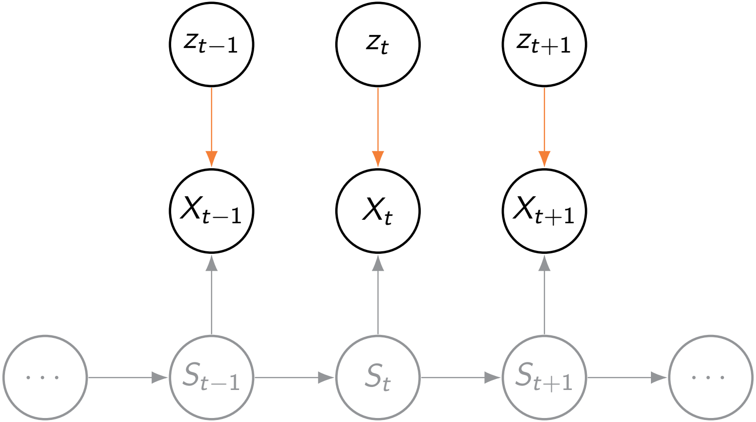
where $\boldsymbol{S}$ is fixed penalty matrix with entries $S_{ij} = \int B_i''(x) B_j''(x) \, dx$.

Nonparametric emission distributions



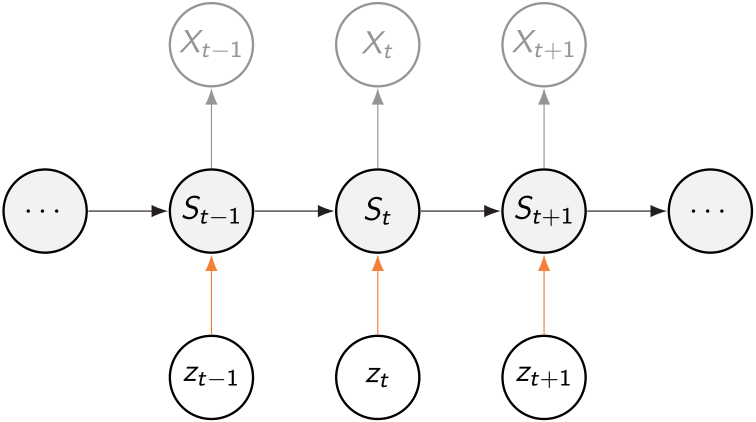
Langrock, Kneib, Sohn, and DeRuiter, 2015

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: $\gamma_{ij}^{(t)} \sim$ s(time of day, day)
- implementation with **fixed** penalisation straightforward

So this is where my spline adventure begins...

- fairly applied project: $\gamma_{ij}^{(t)} \sim \text{s(time of day, day)}$
- implementation with **fixed** 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

FREE

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** ▾

statistica neerlandica

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

Markov-switching generalized additive models

Roland Langrock[1] · Thomas Kneib[2] · Richard Glennie[3] · Théo Michelot[4]

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 $\rightarrow$ 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 $\rightarrow$ 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 $\rightarrow$ 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** $\rightarrow$ **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

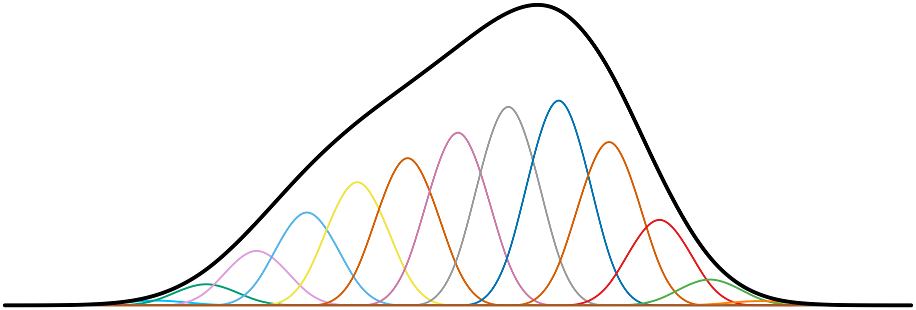
Comput Stat (2012) 27:757–777
DOI 10.1007/s00180-011-0289-6

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:

$$\ell_p(\boldsymbol{b}; \lambda) = \ell(\boldsymbol{b}) - \frac{1}{2}\lambda \boldsymbol{b}^\intercal \boldsymbol{S}\boldsymbol{b},$$

- coefficients $\boldsymbol{b} = (b_1, \dots, b_k)$
- fixed penalty matrix $\boldsymbol{S}$
- penalty strength λ

Why are splines random effects?

Simple setting involving one penalised spline and no unpenalised parameters:

$$\ell_p(\boldsymbol{b}; \lambda) = \ell(\boldsymbol{b}) - \frac{1}{2}\lambda \boldsymbol{b}^\intercal \boldsymbol{S}\boldsymbol{b},$$

- coefficients $\boldsymbol{b} = (b_1, \ldots, b_k)$
- fixed penalty matrix $\boldsymbol{S}$
- penalty strength λ

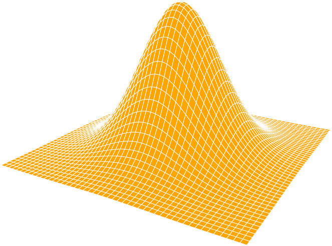
Penalised likelihood function

$$\mathcal{L}_p(\boldsymbol{b}; \lambda) = \mathcal{L}(\boldsymbol{b}) \cdot \exp\left(-\frac{1}{2}\lambda \boldsymbol{b}^\intercal \boldsymbol{S}\boldsymbol{b}\right)$$

We see that

$$\exp\left(-\frac{1}{2}\lambda \boldsymbol{b}^\intercal \boldsymbol{S}\boldsymbol{b}\right)$$

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

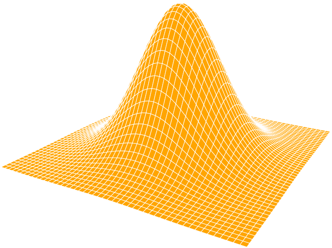


[1] ignoring potential issues with invertibility here

We see that

$$\exp\left(-\frac{1}{2}\lambda \boldsymbol{b}^\intercal \boldsymbol{S}\boldsymbol{b}\right)$$

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



We might as well assume $\boldsymbol{b} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{S}^{-1}/\lambda)$ and add the missing normalisation constant[1]

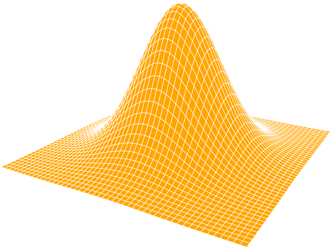
$$\mathcal{L}_j(\boldsymbol{b}; \lambda) = \mathcal{L}(\boldsymbol{b}) \cdot (2\pi)^{-k/2} \det(\lambda \boldsymbol{S})^{1/2} \exp\left(-\frac{1}{2}\lambda \boldsymbol{b}^\intercal \boldsymbol{S}\boldsymbol{b}\right)$$

[1] ignoring potential issues with invertibility here

We see that

$$\exp\left(-\frac{1}{2}\lambda \boldsymbol{b}^\mathsf{T} \boldsymbol{S} \boldsymbol{b}\right)$$

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



We might as well assume $\boldsymbol{b} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{S}^{-1}/\lambda)$ and add the missing normalisation constant[1]

$$\mathcal{L}_j(\boldsymbol{b}; \lambda) = \mathcal{L}(\boldsymbol{b}) \cdot (2\pi)^{-k/2} \det(\lambda \boldsymbol{S})^{1/2} \exp\left(-\frac{1}{2}\lambda \boldsymbol{b}^\mathsf{T} \boldsymbol{S} \boldsymbol{b}\right)$$

giving the joint log-likelihood

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

[1] 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 λ

$$f_\lambda(\boldsymbol{x}, \boldsymbol{b}) = f(\boldsymbol{x} \mid \boldsymbol{b}) \cdot f_\lambda(\boldsymbol{b}),$$

with $f(\boldsymbol{x} \mid \boldsymbol{b}) = \mathcal{L}(\boldsymbol{b})$ and $f_\lambda(\boldsymbol{b}) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{S}^{-1}/\lambda)$.

How to estimate models with random effects?

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

$$f_\lambda(\boldsymbol{x}, \boldsymbol{b}) = f(\boldsymbol{x} \mid \boldsymbol{b}) \cdot f_\lambda(\boldsymbol{b}),$$

with $f(\boldsymbol{x} \mid \boldsymbol{b}) = \mathcal{L}(\boldsymbol{b})$ and $f_\lambda(\boldsymbol{b}) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{S}^{-1}/\lambda)$.

Marginal likelihood of the data by law of total probability

$$\mathcal{L}(\lambda) = f_\lambda(\boldsymbol{x}) = \int f_\lambda(\boldsymbol{x}, \boldsymbol{b}) \, d\boldsymbol{b},$$

How to estimate models with random effects?

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

$$f_\lambda(\boldsymbol{x}, \boldsymbol{b}) = f(\boldsymbol{x} \mid \boldsymbol{b}) \cdot f_\lambda(\boldsymbol{b}),$$

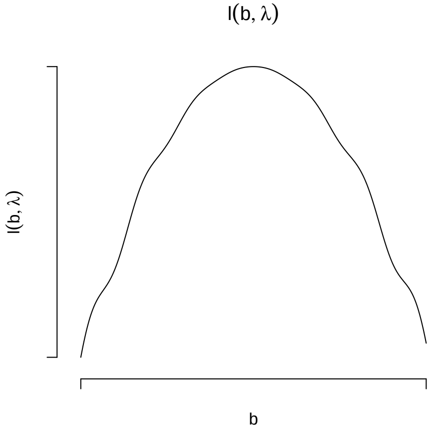
with $f(\boldsymbol{x} \mid \boldsymbol{b}) = \mathcal{L}(\boldsymbol{b})$ and $f_\lambda(\boldsymbol{b}) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{S}^{-1}/\lambda)$.

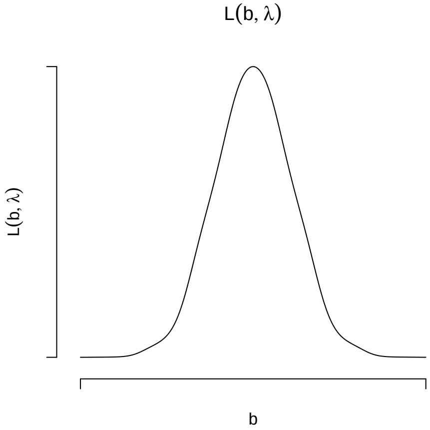
Marginal likelihood of the data by law of total probability

$$\mathcal{L}(\lambda) = f_\lambda(\boldsymbol{x}) = \int f_\lambda(\boldsymbol{x}, \boldsymbol{b})\, d\boldsymbol{b},$$

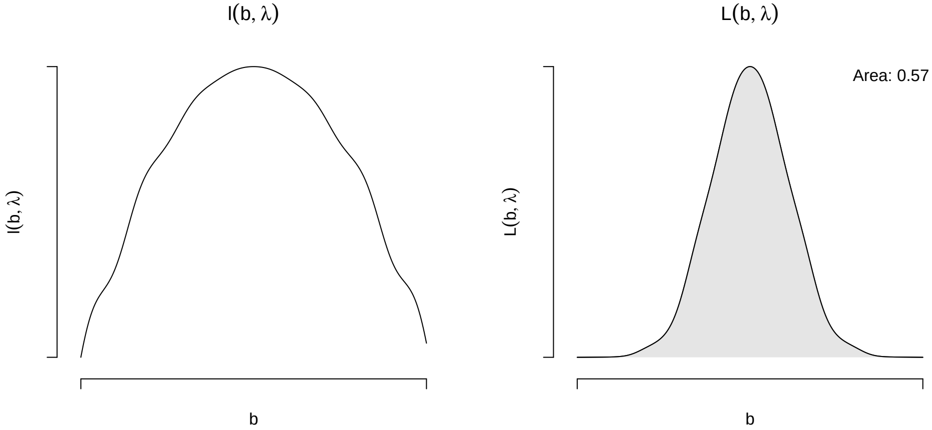
which we would like to maximise to find the **MLE** $\hat{\lambda}$.

Marginal ML

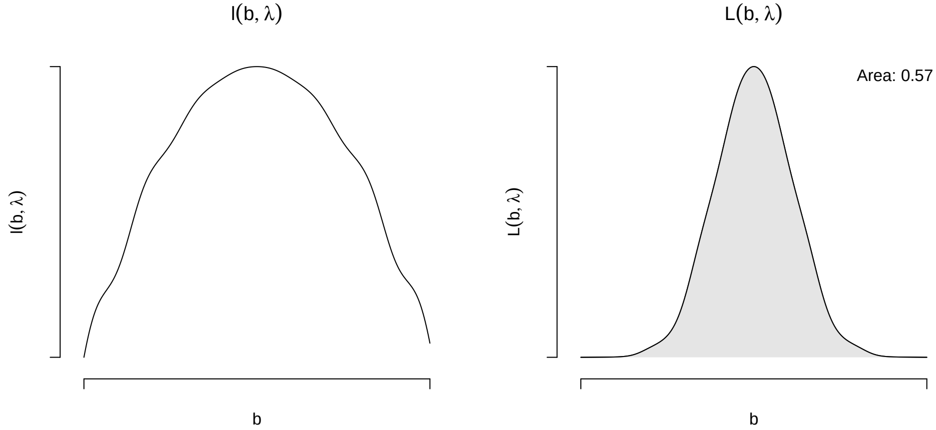




Marginal ML



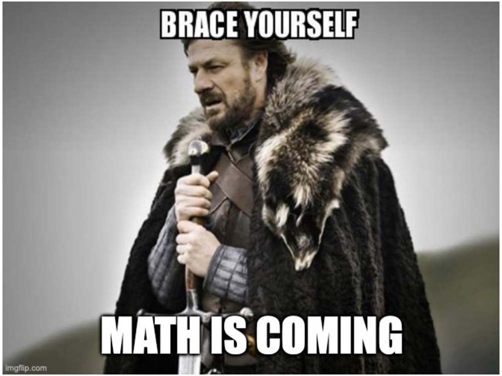
Marginal ML



In reality, this integral is intractable!

What can we do?

What can we do?



The Laplace approximation

- find the mode (in $\boldsymbol{b}$) of the joint log-likelihood

$$\ell_j(\boldsymbol{b}, \lambda) = -\frac{k}{2}\log(2\pi) + \frac{1}{2}\log\det(\lambda S) + \ell(\boldsymbol{b}) - \frac{1}{2}\lambda \boldsymbol{b}^\mathsf{T}\boldsymbol{S}\boldsymbol{b}$$

by **penalised ML**

The Laplace approximation

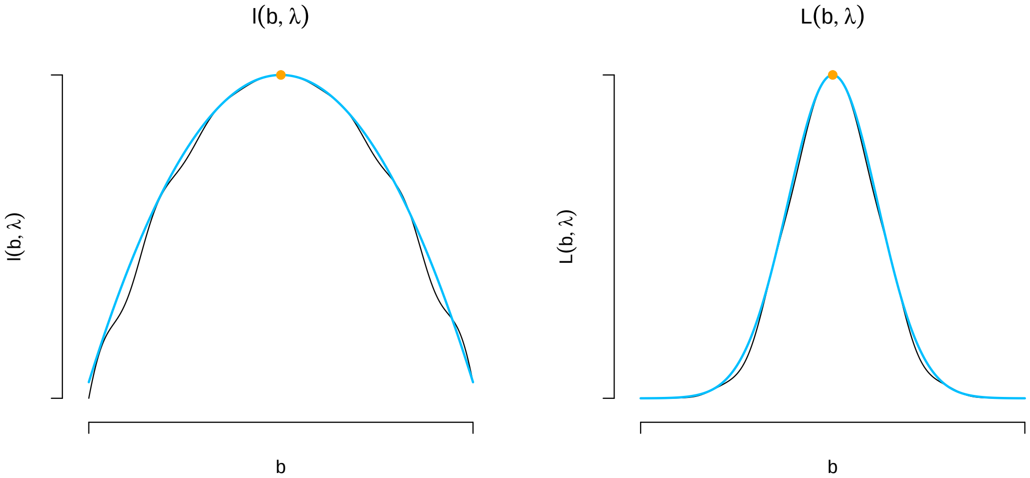
- find the mode (in $\boldsymbol{b}$) of the joint log-likelihood

$$\ell_j(\boldsymbol{b}, \lambda) = -\frac{k}{2}\log(2\pi) + \frac{1}{2}\log\det(\lambda S) + \ell(\boldsymbol{b}) - \frac{1}{2}\lambda \boldsymbol{b}^\mathsf{T}\boldsymbol{S}\boldsymbol{b}$$

  by **penalised ML**
- second-order **Taylor approximation** around the mode:

$$\ell_{approx}(\boldsymbol{b}, \lambda) = \ell_j(\hat{\boldsymbol{b}}, \lambda) - \frac{1}{2}(\boldsymbol{b} - \hat{\boldsymbol{b}})^\mathsf{T}\boldsymbol{J}(\lambda)(\boldsymbol{b} - \hat{\boldsymbol{b}}),$$

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



$l(b, \lambda)$
$L(b, \lambda)$
$l(b, \lambda)$
$L(b, \lambda)$
b
b

- now exponentiate to obtain likelihood and integrate out $\boldsymbol{b}$

$$\exp\bigl(\ell_j(\hat{\boldsymbol{b}}, \lambda)\bigr) \cdot \int \exp\bigl(-\frac{1}{2}(\boldsymbol{b} - \hat{\boldsymbol{b}})^\mathsf{T} \boldsymbol{J}(\lambda)(\boldsymbol{b} - \hat{\boldsymbol{b}})\bigr)\, d\boldsymbol{b}$$

- now exponentiate to obtain likelihood and integrate out $\boldsymbol{b}$

$$\exp\big(\ell_j(\hat{\boldsymbol{b}}, \lambda)\big) \cdot \int \exp\big(-\frac{1}{2}(\boldsymbol{b} - \hat{\boldsymbol{b}})^\mathsf{T} \boldsymbol{J}(\lambda)(\boldsymbol{b} - \hat{\boldsymbol{b}})\big)\, d\boldsymbol{b}$$

- right side is a Gaussian integral and equals

$$(2\pi)^{k/2} \cdot \det(\boldsymbol{J}(\lambda))^{-1/2}$$

- now exponentiate to obtain likelihood and integrate out $\boldsymbol{b}$

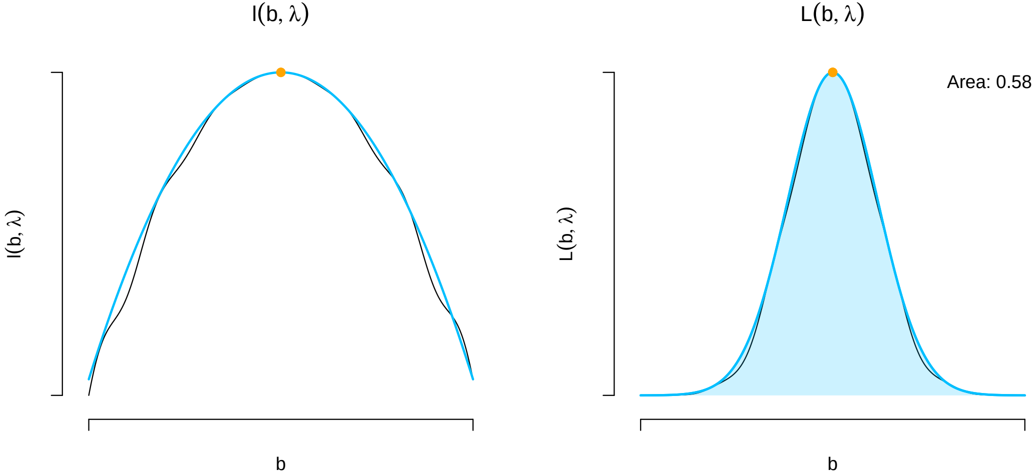
$$\exp\bigl(\ell_j(\hat{\boldsymbol{b}}, \lambda)\bigr) \cdot \int \exp\bigl(-\frac{1}{2}(\boldsymbol{b} - \hat{\boldsymbol{b}})^{\intercal}\boldsymbol{J}(\lambda)(\boldsymbol{b} - \hat{\boldsymbol{b}})\bigr)\, d\boldsymbol{b}$$

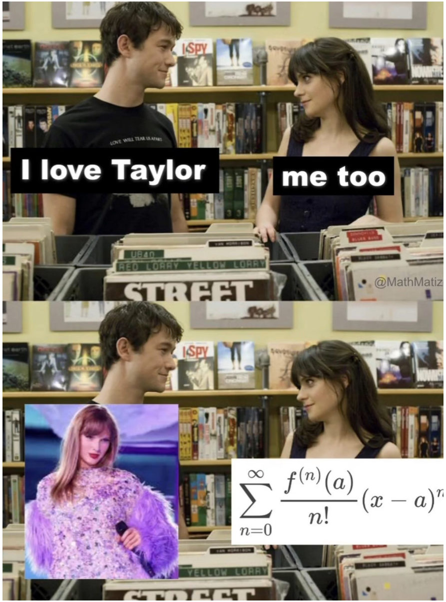
- right side is a Gaussian integral and equals

$$(2\pi)^{k/2} \cdot \det(\boldsymbol{J}(\lambda))^{-1/2}$$

- hence, approximate marginal log-likelihood becomes

$$\ell(\lambda) \approx \ell_j(\hat{\boldsymbol{b}}, \lambda) + \frac{k}{2}\log(2\pi) - \frac{1}{2}\log\det(\boldsymbol{J}(\lambda))$$


l(b, λ)
l(b, λ)
b
L(b, λ)
Area: 0.58
L(b, λ)
b


I love Taylor
me too
@MathMatiz
$$\sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n$$

- 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 $n \to \infty$

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

Optimising the marginal likelihood

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

$$\ell(\lambda) \approx \ell_j(\hat{\boldsymbol{b}}, \lambda) + \frac{k}{2}\log(2\pi) - \frac{1}{2}\log\det\big(\boldsymbol{J}(\lambda)\big).$$

Optimising the marginal likelihood

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

$$\ell(\lambda) \approx \ell_j(\hat{\boldsymbol{b}}, \lambda) + \frac{k}{2}\log(2\pi) - \frac{1}{2}\log\det\big(\boldsymbol{J}(\lambda)\big).$$

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

$$\ell(\lambda) \approx -\frac{k}{2}\log(2\pi) + \frac{1}{2}\log\det(\lambda S) + \ell(\hat{\boldsymbol{b}}) - \frac{1}{2}\lambda\hat{\boldsymbol{b}}^\intercal\boldsymbol{S}\hat{\boldsymbol{b}} + \frac{k}{2}\log(2\pi) - \frac{1}{2}\log\det\big(\boldsymbol{J}(\lambda)\big),$$

Optimising the marginal likelihood

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

$$\ell(\lambda) \approx \ell_j(\hat{\boldsymbol{b}}, \lambda) + \frac{k}{2}\log(2\pi) - \frac{1}{2}\log\det\big(\boldsymbol{J}(\lambda)\big).$$

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

$$\ell(\lambda) \approx -\frac{k}{2}\log(2\pi) + \frac{1}{2}\log\det(\lambda S) + \ell(\hat{\boldsymbol{b}}) - \frac{1}{2}\lambda\hat{\boldsymbol{b}}^\intercal \boldsymbol{S}\hat{\boldsymbol{b}} + \frac{k}{2}\log(2\pi) - \frac{1}{2}\log\det\big(\boldsymbol{J}(\lambda)\big),$$

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

Optimising the marginal likelihood

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

$$\frac{\partial}{\partial\lambda}\left(-\frac{1}{2}\lambda\hat{\boldsymbol{b}}^\intercal \boldsymbol{S}\hat{\boldsymbol{b}}\right) = -\frac{1}{2}\hat{\boldsymbol{b}}^\intercal \boldsymbol{S}\hat{\boldsymbol{b}}$$

$$\frac{\partial}{\partial\lambda}\left(-\frac{1}{2}\log\det\left(\boldsymbol{J}(\lambda)\right)\right) = -\frac{1}{2}\mathrm{tr}\left(\boldsymbol{J}(\lambda)^{-1}S\right)$$

Optimising the marginal likelihood

Hence in total

$$\frac{\partial \ell(\lambda)}{\partial \lambda} \approx -\frac{1}{2}\hat{\boldsymbol{b}}^\mathsf{T}\boldsymbol{S}\hat{\boldsymbol{b}} + \frac{k}{2\lambda} - \frac{1}{2}\mathrm{tr}\big(\boldsymbol{J}(\lambda)^{-1}\boldsymbol{S}\big),$$

Optimising the marginal likelihood

Hence in total

$$\frac{\partial \ell(\lambda)}{\partial \lambda} \approx -\frac{1}{2}\hat{\boldsymbol{b}}^\intercal \boldsymbol{S}\hat{\boldsymbol{b}} + \frac{k}{2\lambda} - \frac{1}{2}\mathrm{tr}\big(\boldsymbol{J}(\lambda)^{-1}\boldsymbol{S}\big),$$

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

$$\lambda = \frac{\mathrm{tr}\big(\boldsymbol{J}(\lambda)^{-1}\boldsymbol{J}(\lambda=0)\big)}{\hat{\boldsymbol{b}}^\intercal \boldsymbol{S}\hat{\boldsymbol{b}}} = \frac{\mathrm{dof}(\lambda)}{\hat{\boldsymbol{b}}^\intercal \boldsymbol{S}\hat{\boldsymbol{b}}}.$$

Optimising the marginal likelihood

Hence in total

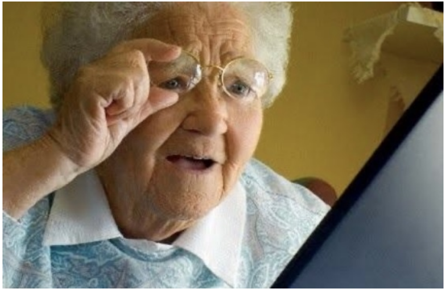
$$\frac{\partial \ell(\lambda)}{\partial \lambda} \approx -\frac{1}{2}\hat{\boldsymbol{b}}^\intercal \boldsymbol{S} \hat{\boldsymbol{b}} + \frac{k}{2\lambda} - \frac{1}{2}\mathrm{tr}\big(\boldsymbol{J}(\lambda)^{-1}\boldsymbol{S}\big),$$

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

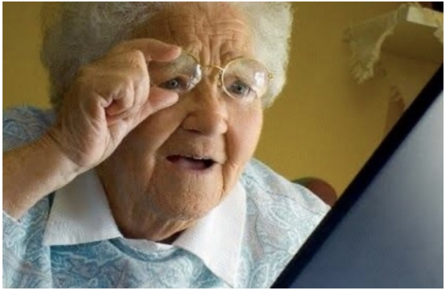
$$\lambda = \frac{\mathrm{tr}\big(\boldsymbol{J}(\lambda)^{-1}\boldsymbol{J}(\lambda = 0)\big)}{\hat{\boldsymbol{b}}^\intercal \boldsymbol{S} \hat{\boldsymbol{b}}} = \frac{\mathrm{dof}(\lambda)}{\hat{\boldsymbol{b}}^\intercal \boldsymbol{S} \hat{\boldsymbol{b}}}.$$

which yields the iterative procedure:

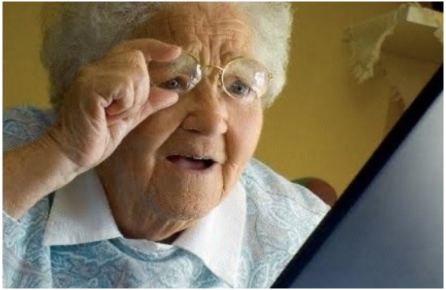
1. fit model via **penalised ML**
2. calculate **Hessian** at optimum
3. **update** penalty strength
4. repeat 1.-3. until convergence



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



- first try $\rightarrow$ worked much better than CV or AIC/ BIC, requiring only a few iterations for convergence ($\sim$ 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

$$\ell_p(\boldsymbol{a}, \boldsymbol{b}; \boldsymbol{\lambda}) = \ell(\boldsymbol{a}, \boldsymbol{b}) - \frac{1}{2} \sum_i \lambda_i \boldsymbol{b}_i^\intercal \boldsymbol{S} \boldsymbol{b}_i$$

- fixed effects $\boldsymbol{a}$
- multiple random effects $\boldsymbol{b} = (\boldsymbol{b}_1, \dots, \boldsymbol{b}_p)$

$$\boldsymbol{b}_i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{S}^{-1}/\lambda_i)$$

Problem: If we integrate out $\boldsymbol{b}$, marginal likelihood is more complicated beause of $\boldsymbol{a}$.

The full setting

$$\ell_p(\boldsymbol{a}, \boldsymbol{b}; \boldsymbol{\lambda}) = \ell(\boldsymbol{a}, \boldsymbol{b}) - \frac{1}{2} \sum_i \lambda_i \boldsymbol{b}_i^\top \boldsymbol{S} \boldsymbol{b}_i$$

- fixed effects $\boldsymbol{a}$
- multiple random effects $\boldsymbol{b} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_p)$

$$\boldsymbol{b}_i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{S}^{-1}/\lambda_i)$$

Problem: If we integrate out $\boldsymbol{b}$, marginal likelihood is more complicated beause of $\boldsymbol{a}$.

Solution: Assume $\boldsymbol{a} \sim \mathcal{N}(\mathbf{0}, \infty)$ and integrate out both $\boldsymbol{a}$ and $\boldsymbol{b}$ $\rightarrow$ restricted maximum likelihood (REML), Laird and Ware, 1982

Quasi REML

Then, marginal log-likelihood is a function of $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_p)$.

Quasi REML

Then, marginal log-likelihood is a function of $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_p)$.

Partially differentiating w.r.t. λ_i yields very similar result:

$$\lambda_i = \frac{\operatorname{tr}\left(\left[\boldsymbol{J}(\lambda)^{-1}\boldsymbol{J}(\lambda = 0)\right]_{ii}\right)}{\hat{\boldsymbol{b}}_i^\top \boldsymbol{S} \hat{\boldsymbol{b}}_i} = \frac{\operatorname{dof}(\lambda_i)}{\hat{\boldsymbol{b}}_i^\top \boldsymbol{S} \hat{\boldsymbol{b}}_i}$$

Quasi REML

Then, marginal log-likelihood is a function of $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_p)$.

Partially differentiating w.r.t. λ_i yields very similar result:

$$\lambda_i = \frac{\mathrm{tr}\left(\left[\boldsymbol{J}(\lambda)^{-1}\boldsymbol{J}(\lambda = 0)\right]_{ii}\right)}{\hat{\boldsymbol{b}}_i^\top \boldsymbol{S} \hat{\boldsymbol{b}}_i} = \frac{\mathrm{dof}(\lambda_i)}{\hat{\boldsymbol{b}}_i^\top \boldsymbol{S} \hat{\boldsymbol{b}}_i}$$

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

Quasi REML

Then, marginal log-likelihood is a function of $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_p)$.

Partially differentiating w.r.t. λ_i yields very similar result:

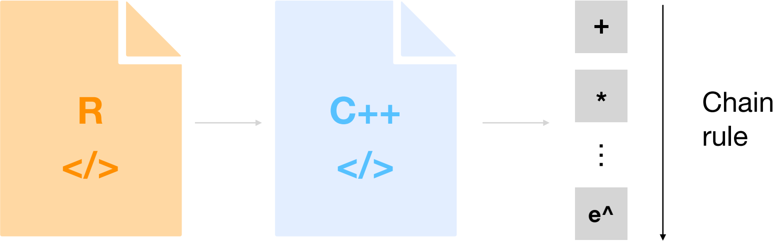
$$\lambda_i = \frac{\mathrm{tr}\Big(\big[\boldsymbol{J}(\lambda)^{-1}\boldsymbol{J}(\lambda = 0)\big]_{ii}\Big)}{\hat{\boldsymbol{b}}_i^\top \boldsymbol{S} \hat{\boldsymbol{b}}_i} = \frac{\mathrm{dof}(\lambda_i)}{\hat{\boldsymbol{b}}_i^\top \boldsymbol{S} \hat{\boldsymbol{b}}_i}$$

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

$\rightarrow$ 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 plain `R` code



RTMB enters the picture

So did we gain anything?

`RTMB` enters the picture

So did we gain anything?

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

RTMB enters the picture

So did we gain anything?

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

`RTMB` enters the picture

So did we gain anything?

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

Practical usage

```r
library(LaMa)

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

Practical usage

```r
library(LaMa)

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

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

Practical usage

```r
library(LaMa)

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

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

```r
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**

- $\text{ODBA}_k^{(t)} \mid \{S_t = i\} \sim \beta_k^{(i)} + s_k(\text{time}_t), k = 1, \ldots, 7$ (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**

- $\text{ODBA}_k^{(t)} \mid \{S_t = i\} \sim \beta_k^{(i)} + s_k(\text{time}_t), k = 1, \ldots, 7$ (ectotherms)
- $\gamma_{ij}^{(t)} \sim s(\text{tod}_t) + \text{AvgTemp}_t * s(\text{tod}_t)$ (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**

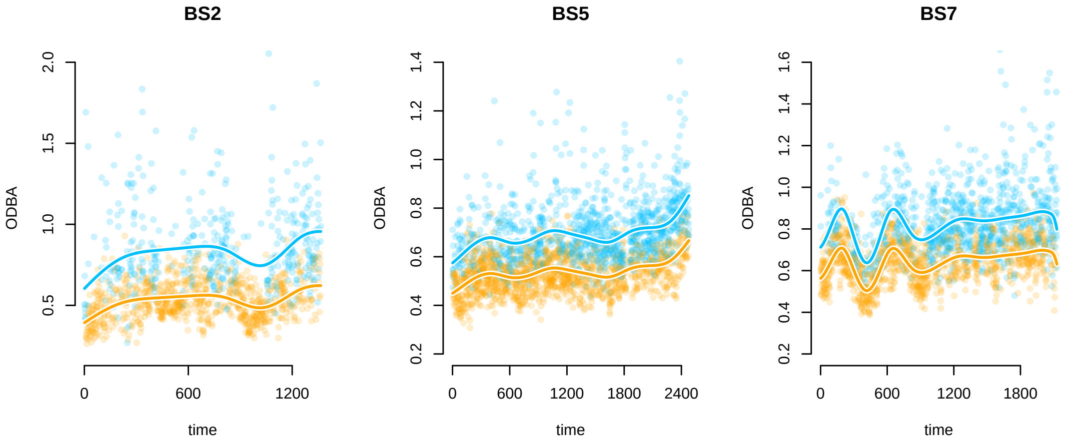
- $\text{ODBA}_k^{(t)} \mid \{S_t = i\} \sim \beta_k^{(i)} + s_k(\text{time}_t), k = 1, \ldots, 7$ (ectotherms)
- $\gamma_{ij}^{(t)} \sim s(\text{tod}_t) + \text{AvgTemp}_t * s(\text{tod}_t)$ (parametric in original paper)
- **11** smooths in total, **162** parameters

Bull sharks (Byrnes et al., 2023)

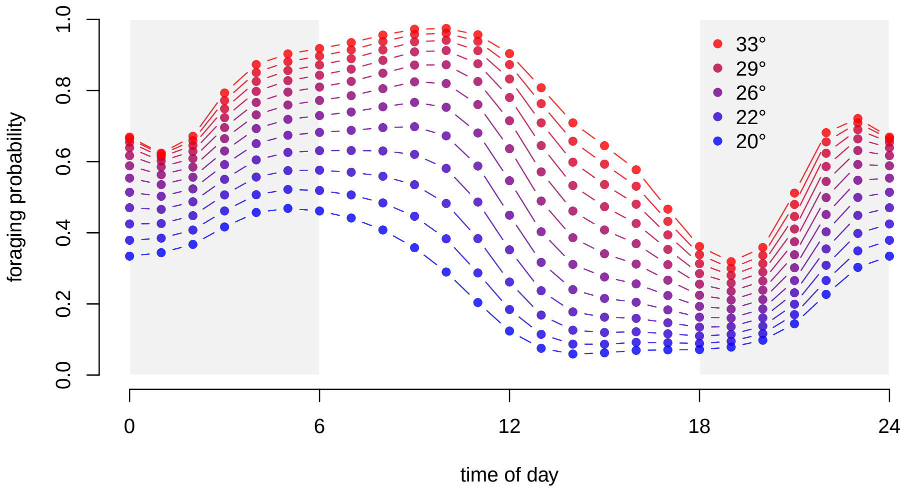
- Western Australia, extreme seasonal changes
- seven bull sharks tagged (14K observations)
- temperature, depth, and acceleration data
- response: overall dynamic body acceleration
- 2-state HMM: **low** and **high activity**

- $\text{ODBA}_k^{(t)} \mid \{S_t = i\} \sim \beta_k^{(i)} + s_k(\text{time}_t), k = 1, \dots, 7$ (ectotherms)
- $\gamma_{ij}^{(t)} \sim s(\text{tod}_t) + \text{AvgTemp}_t * s(\text{tod}_t)$ (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


THANK YOU FOR YOUR ATTENTION!
imgflip.com