Distributed Lag Non-linear Models (DLNMs)

Methodology and Application to Time Series Data Analysis

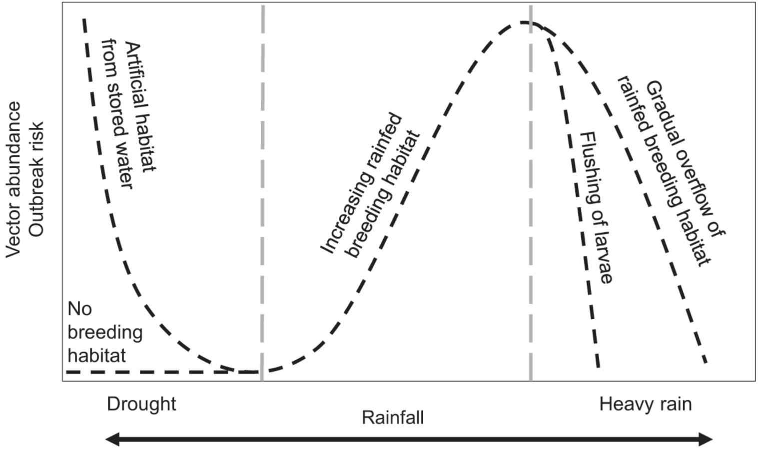
Vincent Arumadri

Department of Public Health

Erasmus MC

8th April, 2025

Non-linear association between rainfall and vector abundance and outbreak risk

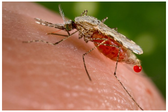


Caldwell et al. 2021, Nature Communications

DLNMs: Conceptual model

Rainfall
(Exposure)

Leads
$\longrightarrow$
Exposure-response



Malaria cases
(Response)

Rainfall
(Exposure)

Delayed effect?
$\longrightarrow$
Exposure-lag-response

Malaria cases
(Response)

Key definitions

- **Non-linear data associations**: data where there is no linear relationship between a dependent (**outcome/response**) and an independent (**exposure/predictor**) variable
- **Time series data**: a sequence of data points collected **over an interval of time** e.g daily rainfall measurements, weekly sales
- **Lag**: time difference between two observations in a sequence

Day	$Value$	$Lag-1$	$Lag-2$
1	10	NA	NA
2	20	10	NA
3	30	20	10
4	40	30	20
5	50	40	30

Key definitions

- **Basis**: known family of **functions/transformations** e.g polynomials, thresholds, splines etc **applied to a predictor X to generate basis variables**: $b_1(X), b_2(X) \dots b_k(X)$.

$$y = \beta_0 + \beta_1 b_1(x) + \beta_2 b_2(x) + \dots + \beta_k b_k(x) + \epsilon \quad (1)$$

- **Basis function** for polynomial takes the form:

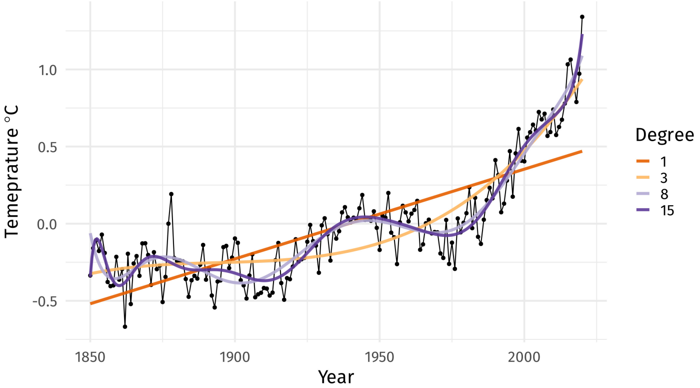
$$\underbrace{b_j(x)}_{\text{polynomial function of degree j}} = \underbrace{x^j}_{\text{Raise predictor x to degree j}} \quad (2)$$

- Substituting equation (2) in (1)

$$y = \beta_0 + \underbrace{\beta_1 x}_{\text{Degree 1 (linear term)}} + \underbrace{\beta_2 x^2}_{\text{Degree 2 polynomial}} + \overbrace{\beta_3 x^3}^{\text{Degree 3 polynomial}} + \dots + \underbrace{\beta_d x^d}_{\text{Degree d polynomial}} + \epsilon$$

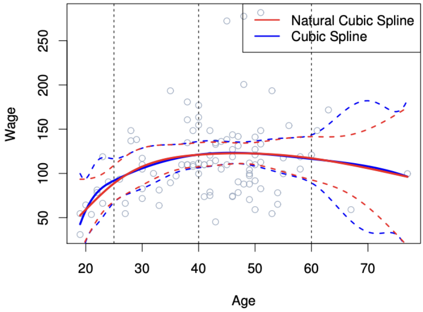
Modelling non-linear data associations

- **Polynomials** and **cubic splines (degree 3 polynomials)** are the most common **basis** used to fit non-linear associations
- Unusual to use *d* greater than **3 or 4, overfitting and wiggly**



Natural (restricted) cubic spline

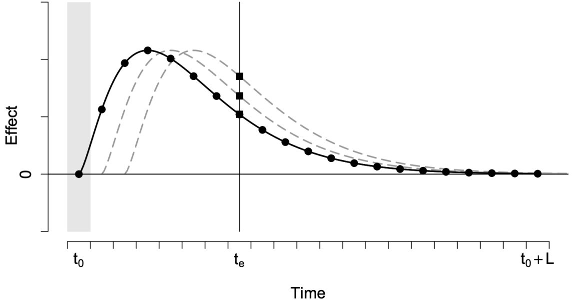
- **Natural cubic spline**: cubic spline with **additional boundary constraints**, enforcing linearity beyond boundary knots
- Produce more **stable estimates at boundaries** (narrower confidence intervals) than cubic splines



- **How are these concepts used in DLNMs?**

DLNMs: Modelling framework

- DLNMs capture a detailed representation of the **time-course** of the **exposure-lag-response relationship**
- Risk associated with individual exposure events at each lag **assigned a weight** that contributes to **overall cumulative risk**



- **Statistical issue is to model this risk!**

Basic model

- A general statistical model representation to describe the **time series of outcomes** Y_t with $t = 1, \ldots, n$ is given by:

$$\overbrace{g(\mu_t)}^{\text{Link function}} = \alpha + \sum_{j=1}^{J} \overbrace{s_j(x_{tj}; \beta)}^{\text{Smoothed predictor}} + \sum_{k=1}^{K} \underbrace{\gamma_k u_{tk}}_{\text{Other predictors with linear effects}}$$

- x_{tj} is the transformed (**non-linear/smoothed**) exposure at time t through basis function j

- β is (**linear**) unknown coefficient of x_{tj} to be estimated

Exposure-lag-response associations

- The risk is represented by a function $s(x, t)$ defined in terms of both **intensity** and **timing** of a series of **past exposures**:
  - an **exposure-response** function $f(x)$ for **exposure** x
  - a **lag-response** function $w(\ell)$ for **lag** ℓ
- Generating a **bi-dimensional exposure-lag-response** function

$$s(x, t) = f(x) \cdot w(\ell)$$

that describes **simultaneously** both the **intensity** and **timing** of **past exposures**

Erasmus MC
University Medical Center Rotterdam

Basis for exposure-response function

- Given, a timeseries of exposure X and assuming a **maximum lag of 2**, we can compute, q_{xt} (**vector of lagged exposure histories of X**)
- Applying a **linear transformation** to q_{xt} we get R_{xt} (**basis variables for lagged occurrences of X**)

t	x
1	10
2	20
3	30
4	40
5	50

$\Rightarrow$

lag 0	lag 1	lag 2
10	NA	NA
20	10	NA
30	20	10
40	30	20
50	40	30

$\Rightarrow$

$$\begin{bmatrix} 10 & NA & NA \\ 20 & 10 & NA \\ 30 & 20 & 10 \\ 40 & 30 & 20 \\ 50 & 40 & 30 \end{bmatrix}$$

Basis for lag-response function

- Applying **polynomial transformation of degree 2** to the lag vector, $\ell(0,1,2)$
- First step is to scale the lag vector by dividing by the maximum lag:

$$(0,1,2)/2 \Rightarrow (0,0.5,1)$$

- Obtaining C (**basis variables for each lag for polynomial degrees d = 0,1,2**)

x^d	x^0	x^1	x^2
lag 0 (0)	1	0	0
lag 1 (0.5)	1	0.5	0.25
lag 2 (1)	1	1	1

$$\Rightarrow C = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0.5 & 0.25 \\ 1 & 1 & 1 \end{bmatrix}$$

Special tensor product

- **Simultaneously** captures the **intensity and timing of past exposures**

$$A_{xt} = (1_{v\ell} \otimes R_{xt}) \odot (C \otimes 1_{vx})$$

Hadamard product

Kronecker product

- $1_{v\ell}$: Vector of 1's of dimensional length of **lag vector**

$$\ell(0,1,2) \Rightarrow 1_{v\ell} = [1,1,1]$$

- 1_{vx}: Vector of 1's of dimensional length of **exposure vector**

t	x
1	10
2	20
3	30
4	40
5	50

$$\Rightarrow 1_{vx} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$1_{v\ell} \otimes R_{xt}$

$$[1,1,1] \otimes \begin{bmatrix} 10 & NA & NA \\ 20 & 10 & NA \\ 30 & 20 & 10 \\ 40 & 30 & 20 \\ 50 & 40 & 30 \end{bmatrix} = \begin{bmatrix} 10 & 10 & 10 \\ 20 & 20 & 20 \\ 30 & 30 & 30 \\ 40 & 40 & 40 \\ 50 & 50 & 50 \\ NA & NA & NA \\ 10 & 10 & 10 \\ 20 & 20 & 20 \\ 30 & 30 & 30 \\ 40 & 40 & 40 \\ NA & NA & NA \\ NA & NA & NA \\ 10 & 10 & 10 \\ 20 & 20 & 20 \\ 30 & 30 & 30 \end{bmatrix}$$

$C \otimes 1_{vx}$

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 0.5 & 0.25 \\ 1 & 1 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0.5 & 0.25 \\ 1 & 0.5 & 0.25 \\ 1 & 0.5 & 0.25 \\ 1 & 0.5 & 0.25 \\ 1 & 0.5 & 0.25 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$(1_{v\ell} \otimes R_{xt}) \odot (C \otimes 1_{vx}) = A_{xt}$$

$$\begin{bmatrix} 10 & 10 & 10 \\ 20 & 20 & 20 \\ 30 & 30 & 30 \\ 40 & 40 & 40 \\ 50 & 50 & 50 \\ NA & NA & NA \\ 10 & 10 & 10 \\ 20 & 20 & 20 \\ 30 & 30 & 30 \\ 40 & 40 & 40 \\ NA & NA & NA \\ NA & NA & NA \\ 10 & 10 & 10 \\ 20 & 20 & 20 \\ 30 & 30 & 30 \end{bmatrix} \odot \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0.5 & 0.25 \\ 1 & 0.5 & 0.25 \\ 1 & 0.5 & 0.25 \\ 1 & 0.5 & 0.25 \\ 1 & 0.5 & 0.25 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \left[\begin{array}{ccc} 10 & 0 & 0 \\ 20 & 0 & 0 \\ 30 & 0 & 0 \\ 40 & 0 & 0 \\ 50 & 0 & 0 \\ \hline NA & NA & NA \\ 10 & 5 & 2.5 \\ 20 & 10 & 5 \\ 30 & 15 & 7.5 \\ 40 & 20 & 10 \\ \hline NA & NA & NA \\ NA & NA & NA \\ 10 & 10 & 10 \\ 20 & 20 & 20 \\ 30 & 30 & 30 \end{array}\right]$$

Cumulative risk of exposures across lags

- From Gasparrini et al 2010 "...array A_{xt} is then re-arranged **summing along the third dimension of lags** to obtain the **final matrix of cross-basis functions, w_{xt}**."

$$A_{x_t} \Rightarrow \begin{bmatrix} 10 & 0 & 0 \\ 20 & 0 & 0 \\ 30 & 0 & 0 \\ 40 & 0 & 0 \\ 50 & 0 & 0 \end{bmatrix} \oplus \begin{bmatrix} NA & NA & NA \\ 10 & 5 & 2.5 \\ 20 & 10 & 5 \\ 30 & 15 & 7.5 \\ 40 & 20 & 10 \end{bmatrix} \oplus \begin{bmatrix} NA & NA & NA \\ NA & NA & NA \\ 10 & 10 & 10 \\ 20 & 20 & 20 \\ 30 & 30 & 30 \end{bmatrix}$$

$$Direct - sum(\oplus) \Rightarrow \begin{bmatrix} NA & NA & NA \\ NA & NA & NA \\ 60 & 20 & 15 \\ 90 & 35 & 27.5 \\ 120 & 50 & 40 \end{bmatrix} = w_{xt}\beta$$

Crossbasis functions

```r
# Load package
pacman::p_load("dlnm")

# data
x <- data.frame(
  t = 1:5,
  value = c(10, 20, 30, 40, 50)
)
x
```

```
##   t value
## 1 1    10
## 2 2    20
## 3 3    30
## 4 4    40
## 5 5    50
```

```r
# crossbasis
cb.x <- crossbasis(x$value, lag=2,
                   argvar=list(fun = "lin"),
                   arglag=list(fun="poly", degree=2))

# crossbasis matrix
head(cb.x, 5)
```

```
##      v1.l1 v1.l2 v1.l3
## [1,]    NA    NA    NA
## [2,]    NA    NA    NA
## [3,]    60    20  15.0
## [4,]    90    35  27.5
## [5,]   120    50  40.0
```

Distributed lag non-linear models (DLNMs)

- **Bi-dimensional exposure-lag-response** function $f(x) \cdot w(\ell)$:

$$s(x,t) = \int_{\ell_0}^{L} f(x_{t-\ell}) \cdot w(\ell)\, d\ell$$

- Approximation obtained through a **discretization of the lag period into equally spaced time units,** q_{x_t}

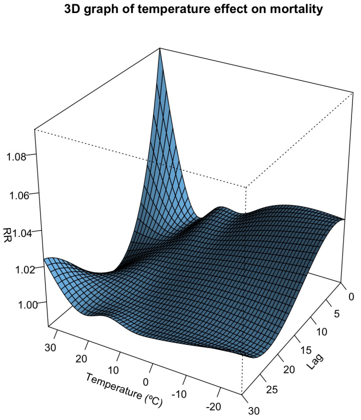
$$s(x_{t-\ell_0}, \ldots, x_{t-L}) \approx \sum_{\ell=\ell_0}^{L} f(x_{t-\ell}) \cdot w(\ell)$$

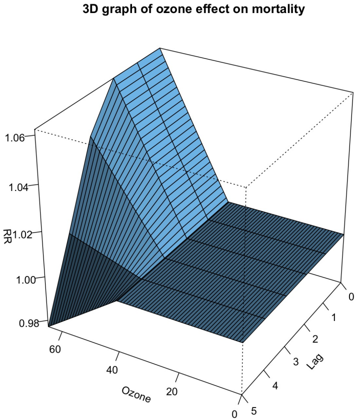
- The **problem reduces to choosing a basis function** for **exposure-response** (q_{x_t}) and **lag-response** (ℓ) space

Effect of temperature and ozone on mortality

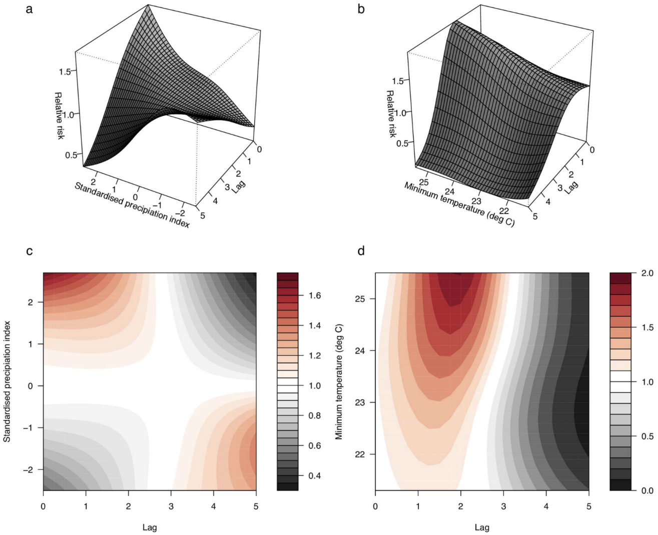
```
# Load packages and data
pacman::p_load("dlnm", "splines")
chicagoNMMAPS <- chicagoNMMAPS

# Objective: to investigate the effects of temperature and
# Ozone on mortality up to lag 30 and 5, respectively

# crossbasis ozone
cb.o3 <- crossbasis(chicagoNMMAPS$o3, lag=5,
                    argvar=list(fun="thr", thr=40.3),
                    arglag=list(fun="thr"))

# crossbasis temperature
cb.temp <- crossbasis(chicagoNMMAPS$temp, lag=30,
                      argvar=list(fun = "ns", df=5),
                      arglag=list(fun="bs"))

# model
model <- glm(death ~ cb.o3 + cb.temp + dow,
             family=quasipoisson(), chicagoNMMAPS)

# pred (extract estimated associations predicted by model)
pred.temp <- crosspred(cb.temp, model, cen=21)
pred.o3 <- crosspred(cb.o3, model, at=c(0:65,40.3))

# plots
plot(pred.temp, xlab="Temperature (ºC)", zlab="RR",
     main="3D graph of temperature effect on mortality")
```

Effect of temperature and ozone on mortality





Effect of rainfall and temperature on dengue risk



References

- A. Gasparrini, Armstrong, and Kenward 2010
- Antonio Gasparrini 2011
- Gareth James • Daniela Witten • Trevor Hastie and Robert Tibshirani 2013
- Antonio Gasparrini and Leone 2014
- Aßenmacher 2016
- Lowe et al. 2018

Thank you!

Slides: https://github.com/arumadri/dlnm