Distributional Regression
Computation, Model Choice and Variable Selection

Nikolaus Umlauf

https://eeecon.uibk.ac.at/~umlauf/
Overview

1 Introduction
2 Model specification
3 Model fitting
4 L1-type penalization
5 R package \textit{bamlss}
6 An application on the Bird Breeding Survey
Introduction

Zugspitze daily maximum temperature data (1900/08-2016/12)

\[ T \sim N(\mu, \sigma^2). \]
Zugspitze daily maximum temperature data (1900/08-2016/12)

\[ T \sim N(\mu = f(T_{t-1}), \log(\sigma^2) = \beta_0). \]
Introduction

Zugspitze daily maximum temperature data (1900/08-2016/12)

\[ T \sim N(\mu = f(T_{t-1}), \log(\sigma^2) = f(T_{t-1})) \]
Introduction

Zugspitze daily maximum temperature data (1900/08-2016/12)

\[ T \sim N(\mu = f(T_{t-1}), \log(\sigma^2) = f(T_{t-1})) \]

![Diagram showing the effect of times on log(\sigma^2)]
Model specification

Any parameter of a population distribution $D$ may be modeled by explanatory variables

$$y \sim D(h_1(\theta_1) = \eta_1, \ h_2(\theta_2) = \eta_2, \ldots, \ h_K(\theta_K) = \eta_K),$$

Each parameter is linked to a structured additive predictor

$$h_k(\theta_k) = \eta_k = \eta_k(x; \beta_k) = f_{1k}(x; \beta_{1k}) + \ldots + f_{J_kk}(x; \beta_{J_kk}),$$

$j = 1, \ldots, J_k$ and $k = 1, \ldots, K$ and $h_k(\cdot)$ are known monotonic link functions.

Vector of function evaluations $f_{jk} = (f_{jk}(x_1; \beta_{jk}), \ldots, f_{jk}(x_n; \beta_{jk}))^\top$

$$f_{jk} = \begin{pmatrix}
  f_{jk}(x_1; \beta_{jk}) \\
  \vdots \\
  f_{jk}(x_n; \beta_{jk})
\end{pmatrix} = f_{jk}(X_{jk}; \beta_{jk}).$$
Model specification

Nonlinear effects of continuous covariates

Two-dimensional surfaces

Spatially correlated effects $f(x) = f(s)$

Random intercepts $f(x) = f(id)$
Model specification

For simple linear effects $X_{jk} \beta_{jk}$: $p_{jk}(\beta_{jk}) \propto \text{const}$.

For the smooth terms:

$$p_{jk}(\beta_{jk}; \tau_{jk}, \alpha_{jk}) \propto d_{\beta_{jk}}(\beta_{jk} | \tau_{jk}, \alpha_{\beta_{jk}}) \cdot d_{\tau_{jk}}(\tau_{jk} | \alpha_{\tau_{jk}}).$$

Using a basis function approach a common choice is

$$d_{\beta_{jk}}(\beta_{jk} | \tau_{jk}, \alpha_{\beta_{jk}}) \propto |P_{jk}(\tau_{jk})|^{1/2} \exp \left(-\frac{1}{2} \beta_{jk}^\top P_{jk}(\tau_{jk}) \beta_{jk} \right).$$

Precision matrix $P_{jk}(\tau_{jk})$ derived from prespecified penalty matrices $\alpha_{\beta_{jk}} = \{K_{1jk}, \ldots, K_{Ljk}\}$.

The variances parameters $\tau_{jk}$ are equivalent to the inverse smoothing parameters in a frequentist approach.
Regularization in the GAMLSS framework

- A gradient boosting approach is provided by Mayr et al. (2012).
- Allows for variable selection within GAMLSS framework.
- Corresponding R-package `gamboostLSS` (Hofner et al., 2015).
- Provides a large number of pre-specified distributions.
- **New:** an alternative *gradient boosting* approach is implemented in the R-package `bamlss` (Umlauf et al., 2018b):
  - embeds many different approaches suggested in literature and software,
  - serves as unified conceptional “Lego toolbox” for complex regression models.
New model terms $f_{jk}(x; \beta_{jk})$ with LASSO-type penalties.
New model terms $f_{jk}(x; \beta_{jk})$ with LASSO-type penalties.
Model fitting

The main building block of regression model algorithms is the probability density function $d_y(y|\theta_1, \ldots, \theta_K)$.

Estimation typically requires to evaluate

$$\ell(\beta; y, X) = \sum_{i=1}^{n} \log d_y(y_i; \theta_{i1} = h_1^{-1}(\eta_{i1}(x_i, \beta_1)), \ldots, \theta_{iK} = h_K^{-1}(\eta_{iK}(x_i, \beta_K))),$$

with $\beta = (\beta_1^T, \ldots, \beta_K^T)^T$ and $X = (X_1, \ldots, X_K)$.

The log-posterior

$$\log \pi(\beta, \tau; y, X, \alpha) \propto \ell(\beta; y, X) + \sum_{k=1}^{K} \sum_{j=1}^{J_k} \left[ \log p_{jk}(\beta_{jk}; \tau_{jk}, \alpha_{jk}) \right],$$

where $\tau = (\tau_1^T, \ldots, \tau_K^T)^T = (\tau_{11}, \ldots, \tau_{J_11}, \ldots, \tau_{1K}, \ldots, \tau_{J_KK})^T$ (frequentist, penalized log-likelihood).
Model fitting

Posterior mode estimation, fortunately, partitioned updating is possible

\[
\begin{align*}
\beta_1^{(t+1)} &= U_1(\beta_1^{(t)}, \beta_2^{(t)}, \ldots, \beta_K^{(t)}) \\
\beta_2^{(t+1)} &= U_2(\beta_1^{(t+1)}, \beta_2^{(t)}, \ldots, \beta_K^{(t)}) \\
&\quad \vdots \\
\beta_K^{(t+1)} &= U_K(\beta_1^{(t+1)}, \beta_2^{(t+1)}, \ldots, \beta_K^{(t)}),
\end{align*}
\]

E.g., Newton-Raphson type updating

\[
\beta_k^{(t+1)} = U_k(\beta_k^{(t)}, \cdot) = \beta_k^{(t)} - H_{kk} \left( \beta_k^{(t)} \right)^{-1} s \left( \beta_k^{(t)} \right).
\]

Can be further partitioned for each function within parameter block \( k \). Moreover, using a basis function approach yields IWLS updates

\[
\beta_{jk}^{(t+1)} = (X_{jk}^\top W_{kk} X_{jk} + G_{jk}(\tau_{jk}))^{-1} X_{jk}^\top W_{kk} (z_k - \eta_{k,-j}^{(t)}).
\]
Model fitting

A simple generic algorithm for distributional regression models:

while(eps > ε & t < maxit) {
    for(k in 1:K) {
        for(j in 1:J[k]) {
            Compute \( \tilde{\eta} = \eta_k - f_{jk}. \)
            Obtain new \( (\beta_{jk}^*, \tau_{jk}^*)^\top = U_{jk}(X_{jk}, y, \tilde{\eta}, \beta_{jk}^{[t]}, \tau_{jk}^{[t]}). \)
            Update \( \eta_k. \)
        }
    }
    t = t + 1
    Compute new eps.
}

Functions \( U_{jk}(\cdot) \) could either return updates from an optimizing algorithm or proposals from a MCMC sampler.
L1-type penalization

**Idea**: depending on the type of covariate effects, subtract a combination of (parts of) the following penalty terms $\tau^{-1} J(\beta)$ from the log-likelihood.

**Classical LASSO** (Tibshirani, 1996): For a metric covariate $x_{jk}$ use

$$J_m(\beta_{jk}) = |\beta_{jk}|.$$  

**Group LASSO** (Meier et al., 2008): For a (dummy-encoded) categorical covariate $x_{jk}$ use

$$J_g(\beta_{jk}) = \|\beta_{jk}\|_2,$$

with vector $\beta_{jk}$ collecting all corresponding coefficients.
L1-type penalization

Alternatively, for categorical covariates often clustering of categories with implicit factor selection is desirable.

**Fused LASSO** (Gertheiss and Tutz, 2010): Depending on the nominal (left) or ordinal scale level (right) of the covariate, use

\[
J_f(\beta_{jk}) = \sum_{l > m} w^{(jk)}_{lm} |\beta_{jkl} - \beta_{jkm}| \quad \text{or} \quad J_f(\beta_{jk}) = \sum_{l=1}^{c_{jk}} w^{(jk)}_l |\beta_{jkl} - \beta_{jk,l-1}|
\]

where \( c_{jk} \) is the number of levels of categorical predictor \( x_{jk} \) and \( w^{(jk)}_{lm}, w^{(jk)}_l \) denote suitable weights. Choosing \( l = 0 \) as the reference, \( \beta_{jk0} = 0 \) is fixed.
L1-type penalization

Quadratic approximations of the penalties (compare Oelker & Tutz, 2017)

\[ J_{jk}(\beta_{jk}) \approx J_{jk}(\beta_{jk}^{(t)}) + \frac{1}{2} \left( \beta_{jk}^\top P_{jk}(\beta_{jk}) \beta_{jk} + (\beta_{jk}^{(t)})^\top P_{jk}(\beta_{jk}^{(t)}) \beta_{jk}^{(t)} \right), \]

with

\[ P_{jk}(\beta_{jk}^{(t)}) = q_j^\prime \left( \left\| a_j^\top \beta_{jk}^{(t)} \right\| N_{jk} \right) \cdot \frac{D_{jk}(a_j^\top \beta_{jk}^{(t)})}{a_j^\top \beta_{jk}^{(t)}} \cdot a_j a_j^\top. \]

E.g., \( \| \beta \|_1 = |\beta| \) is approximated by \( \sqrt{\beta^2 + c} \), hence, IWLS based updating functions \( U_{jk}(\cdot) \) are relatively easy to implement.
L1-type penalization

Example of the approximation of the $L_1$ norm.

Usually setting the constant to $c \approx 10^{-5}$ works well.
R package *bamlss*

The package is available at

https://CRAN.R-project.org/package=bamlss

Development version, in R simply type

```r
R> install.packages("bamlss",
+   repos = "http://R-Forge.R-project.org")
```
In principle, the setup does not restrict to any specific type of engine (Bayesian or frequentist).
## R package *bamlss*

<table>
<thead>
<tr>
<th>Type</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parser</td>
<td><code>bamlss.frame()</code></td>
</tr>
<tr>
<td>Transformer</td>
<td><code>bamlss.engine.setup()</code>, <code>randomize()</code></td>
</tr>
<tr>
<td>Optimizer</td>
<td><code>bfit()</code>, <code>opt()</code>, <code>cox.mode()</code>, <code>jm.mode()</code></td>
</tr>
<tr>
<td></td>
<td><code>boost()</code>, <code>lasso()</code></td>
</tr>
<tr>
<td>Sampler</td>
<td><code>GMCMC()</code>, <code>JAGS()</code>, <code>STAN()</code>, <code>BayesX()</code>,</td>
</tr>
<tr>
<td></td>
<td><code>cox.mcmc()</code>, <code>jm.mcmc()</code></td>
</tr>
<tr>
<td>Results</td>
<td><code>results.bamlss.default()</code></td>
</tr>
</tbody>
</table>

To implement new engines, only the building block functions have to be exchanged.
R package *bamlss*

Work in progress . . .

<table>
<thead>
<tr>
<th>Function</th>
<th>Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>beta_bamlss()</td>
<td>Beta distribution</td>
</tr>
<tr>
<td>binomial_bamlss()</td>
<td>Binomial distribution</td>
</tr>
<tr>
<td>cnorm_bamlss()</td>
<td>Censored normal distribution</td>
</tr>
<tr>
<td>cox_bamlss()</td>
<td>Continuous time Cox-model</td>
</tr>
<tr>
<td>gaussian_bamlss()</td>
<td>Gaussian distribution</td>
</tr>
<tr>
<td>gamma_bamlss()</td>
<td>Gamma distribution</td>
</tr>
<tr>
<td>gpareto_bamlss()</td>
<td>Generalized Pareto distribution</td>
</tr>
<tr>
<td>jm_bamlss()</td>
<td>Continuous time joint-model</td>
</tr>
<tr>
<td>multinomial_bamlss()</td>
<td>Multinomial distribution</td>
</tr>
<tr>
<td>mvn_bamlss()</td>
<td>Multivariate normal distribution</td>
</tr>
<tr>
<td>poisson_bamlss()</td>
<td>Poisson distribution</td>
</tr>
</tbody>
</table>

New families only require density, distribution, random number generator, quantile, score and hess functions.
R package *bamlss*

Wrapper function:

```r
R> f <- list(y ~ la(id,fuse=2), sigma ~ la(id,fuse=1))
R> b <- bamlss(f, family = "gaussian", sampler = FALSE,
+    optimizer = lasso, criterion = "BIC", multiple = TRUE)
```

Standard extractor and plotting functions:

`summary()`, `plot()`, `fitted()`, `residuals()`, `predict()`, `coef()`, `logLik()`, `DIC()`, `samples()`, ...
Bird Breeding Survey


- Long-term, large-scale, international avian monitoring program initiated in 1966 to track the status and trends of North American bird populations.

- Each year during the height of the avian breeding season, participants skilled in avian identification collect bird population data along roadside survey routes.

- At each stop, a 3-minute point count is conducted. During the count, every bird seen within a 0.25-mile radius or heard is recorded.
Bird Breeding Survey

Change of average richness over time?

![Histogram showing species counts and density over time](image1.png)
Bird Breeding Survey

Route specific effects?
Bird Breeding Survey

Model in R (potentially 344 parameters):

\[
R> f \leftarrow \text{list(}
+ \text{ counts} \sim \text{la(year,fuse=2) + la(route,fuse=1) + s(lon,lat,k=50),}
+ \text{ sigma} \sim \text{la(year,fuse=2) + la(route,fuse=1) + s(lon,lat,k=50)}
+ \text{ })
\]

\[
R> b \leftarrow \text{bamlss(f, data = bbs, sampler = FALSE, optimizer = lasso,}
+ \text{ criterion = "BIC", multiple = TRUE, nlambda = 50)}
\]

\[
R> \text{lasso.stop(b)}
\]

[1] 1781
attr(,"stats")
     logLik    logPost      BIC       edf     lambda.mu
-8326.43632 -13856.38433 17201.90386 69.45302     27.18282
lambda.sigma
   11.39852
Bird Breeding Survey

\[ R > \text{lasso.plot}(b, \text{which} = "\text{criterion}" \) 

\[ \mu \]
\[ \lambda = 27.18282 \]

\[ \sigma \]
\[ \lambda = 11.39852 \]
Bird Breeding Survey

```r
R> lasso.plot(b, which = "parameters", model = "mu")
```

\[ \lambda_{\text{opt}} = 27.18 \]
Bird Breeding Survey

R> lasso.plot(b, which = "parameters", model = "sigma")

\[
\lambda_0 = 11.4
\]

\[\beta_i\]

\[\log(\lambda)\]
$R > \text{lasso.plot}(b, \text{which} = "parameters", \text{model} = "mu")$
Bird Breeding Survey

\[ R> \text{lasso.plot}(b, \text{which} = "parameters", \text{model} = "sigma") \]

\[ \lambda_\sigma = 11.4 \]

\[ \beta_1 \]

\[ \log(\lambda) \]
Bird Breeding Survey

\begin{verbatim}
R> p <- predict(b, newdata = nd, model = "mu", +  term = "s(lon, lat)", mstop = lasso.stop(b))
\end{verbatim}
Bird Breeding Survey

R> p <- predict(b, newdata = nd, model = "sigma",
+   term = "s(lon,lat)", mstop = lasso.stop(b))
References & Software


Thank you for your attention!

Nikolaus Umlauf

https://eeecon.uibk.ac.at/~umlauf/