gamboostLSS: boosting generalized additive models for location, scale and shape

Benjamin Hofner

Joint work with Andreas Mayr, Nora Fenske, Thomas Kneib and Matthias Schmid

Department of Medical Informatics, Biometry and Epidemiology
FAU Erlangen-Nürnberg, Germany

useR! 2011

FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG
MEDIZINISCHE FAKULTÄT

Motivation: Munich rental guide

Aim:

- Provide precise point predictions and prediction intervals for the net-rent of flats in the city of Munich.

Data:

- Covariates: 325 (mostly) categorical, 2 continuous and 1 spatial
- Observations: 3016 flats

Problem:

- Heteroscedasticity found in the data

Idea

Model not only the expected mean but also the variance $\Rightarrow$ **GAMLSS**

The GAMLSS model class

Generalized Additive Models for Location, Scale and Shape

$$g_1(\mu) = \eta_\mu = \beta_{0\mu} + \sum_{j=1}^{p_1} f_{j\mu}(x_j) \qquad \text{“location”}$$

$$g_2(\sigma) = \eta_\sigma = \beta_{0\sigma} + \sum_{j=1}^{p_2} f_{j\sigma}(x_j) \qquad \text{“scale”}$$

$$\vdots$$

- Introduced by Rigby and Stasinopoulos (2005)
- Flexible alternative to generalized additive models (GAM)
- Up to four distribution parameters are regressed on the covariates.
- Every distribution parameter is modeled by its own predictor and an associated link function $g_k(\cdot)$.

Current fitting algorithm

The `gamlss` package

Fitting algorithms for a large amount of distribution families are provided by the **R** package `gamlss` (Stasinopoulus and Rigby, 2007).

- Estimation is based on a penalized likelihood approach.
- Modified versions of back-fitting (as for conventional GAMs) are used.

These algorithms work remarkably well in many applications, but:

- It is not feasible for high-dimensional data ($p \gg n$).
- No spatial effects are implemented.
- Variable selection is based on generalized AIC, which is known to be unstable.
  - ▷ *"More work needs to be done here"* (Stasinopoulus and Rigby, 2007).

Optimization problem for GAMLSS

- The task is to model the distribution parameters of the conditional density $f_{\text{dens}}(y|\mu, \sigma, \nu, \tau)$
  - The optimization problem can be formulated as

$$\underbrace{(\hat{\mu}, \hat{\sigma}, \hat{\nu}, \hat{\tau})}_{\boldsymbol{\theta}} \longleftarrow \underset{\eta_\mu, \eta_\sigma, \eta_\nu, \eta_\tau}{\operatorname{argmin}} \mathbb{E}_{Y,X}\Big[\rho\big(Y, \underbrace{\eta_\mu(X), \eta_\sigma(X), \eta_\nu(X), \eta_\tau(X)}_{\boldsymbol{\eta}}\big)\Big]$$

    with loss function $\rho = -l$, i.e., the negative log-likelihood of the response distribution:

$$l = \sum_{i=1}^{n} \log\left[f_{\text{dens}}(y_i|\boldsymbol{\theta}_i)\right] = \sum_{i=1}^{n} \log\left[f_{\text{dens}}(y_i|\mu_i, \sigma_i, \nu_i, \tau_i)\right]$$

  - Maximum likelihood approach

Alternative to ML: ▷ Component-wise boosting

Boosting

- minimizes empirical risk (e.g., **negative log likelihood**)
- in an iterative fashion
- via functional gradient descent (FGD).

In boosting iteration $m + 1$

- Compute (negative) gradient of the loss function and plug in the current estimate
$$u_i^{[m+1]} = - \left. \frac{\partial \rho(y_i, \eta)}{\partial \eta} \right|_{\eta = \hat{\eta}_i^{[m]}}$$
- Estimate $u_i^{[m+1]}$ via base-learners (i.e., simple regression models)
- Update: use only the **best-fitting base-learner**; add a small fraction ν of this estimated base-learner (e.g., 10%) to the model

▷ **Variable selection intrinsically within the fitting process**

Boosting for GAMLSS models

- Boosting was recently extended to risk functions with multiple components (Schmid et al., 2010)
- **Idea** ▷ Use partial derivatives instead of gradient
- Specify a **set of base-learners** — one base-learner per covariate
- Fit each of the base-learners **separately** to the partial derivatives
- **Cycle** through the partial derivatives within each boosting step

Boosting for GAMLSS models

- Boosting was recently extended to risk functions with multiple components (Schmid et al., 2010)
- **Idea** ▷ Use partial derivatives instead of gradient

- Specify a **set of base-learners** — one base-learner per covariate
- Fit each of the base-learners **separately** to the partial derivatives
- **Cycle** through the partial derivatives within each boosting step

$$\frac{\partial \rho}{\partial \eta_{\boldsymbol{\mu}}}(y_i, \hat{\mu}^{[m]}, \hat{\sigma}^{[m]}, \hat{\nu}^{[m]}, \hat{\tau}^{[m]}) \quad \xrightarrow[\text{best fitting BL}]{\text{update}} \quad \hat{\eta}_{\mu}^{[\mathbf{m+1}]} \Longrightarrow \hat{\mu}^{[\mathbf{m+1}]} \ ,$$

Boosting for GAMLSS models

- Boosting was recently extended to risk functions with multiple components (Schmid et al., 2010)
- **Idea** ▷ Use partial derivatives instead of gradient

- Specify a **set of base-learners** — one base-learner per covariate
- Fit each of the base-learners **separately** to the partial derivatives
- **Cycle** through the partial derivatives within each boosting step

$$\frac{\partial \rho}{\partial \eta_{\boldsymbol{\mu}}}(y_i, \hat{\mu}^{[m]}, \hat{\sigma}^{[m]}, \hat{\nu}^{[m]}, \hat{\tau}^{[m]}) \xrightarrow[\text{best fitting BL}]{\text{update}} \hat{\eta}_{\mu}^{[\mathbf{m+1}]} \Longrightarrow \hat{\mu}^{[\mathbf{m+1}]},$$

$$\frac{\partial \rho}{\partial \eta_{\boldsymbol{\sigma}}}(y_i, \hat{\mu}^{[\mathbf{m+1}]}, \hat{\sigma}^{[m]}, \hat{\nu}^{[m]}, \hat{\tau}^{[m]}) \xrightarrow[\text{best fitting BL}]{\text{update}} \hat{\eta}_{\sigma}^{[\mathbf{m+1}]} \Longrightarrow \hat{\sigma}^{[\mathbf{m+1}]},$$

Boosting for GAMLSS models

- Boosting was recently extended to risk functions with multiple components (Schmid et al., 2010)
- **Idea** ▷ Use partial derivatives instead of gradient

- Specify a **set of base-learners** — one base-learner per covariate
- Fit each of the base-learners **separately** to the partial derivatives
- **Cycle** through the partial derivatives within each boosting step

$$\frac{\partial \rho}{\partial \eta_{\boldsymbol{\mu}}}(y_i, \hat{\mu}^{[m]}, \hat{\sigma}^{[m]}, \hat{\nu}^{[m]}, \hat{\tau}^{[m]}) \quad \underset{\text{best fitting BL}}{\overset{\text{update}}{\longrightarrow}} \quad \hat{\eta}_{\mu}^{[\mathbf{m+1}]} \Longrightarrow \hat{\mu}^{[\mathbf{m+1}]} \ ,$$

$$\frac{\partial \rho}{\partial \eta_{\boldsymbol{\sigma}}}(y_i, \hat{\mu}^{[\mathbf{m+1}]}, \hat{\sigma}^{[m]}, \hat{\nu}^{[m]}, \hat{\tau}^{[m]}) \quad \underset{\text{best fitting BL}}{\overset{\text{update}}{\longrightarrow}} \quad \hat{\eta}_{\sigma}^{[\mathbf{m+1}]} \Longrightarrow \hat{\sigma}^{[\mathbf{m+1}]} \ ,$$

$$\frac{\partial \rho}{\partial \eta_{\boldsymbol{\nu}}}(y_i, \hat{\mu}^{[\mathbf{m+1}]}, \hat{\sigma}^{[\mathbf{m+1}]}, \hat{\nu}^{[m]}, \hat{\tau}^{[m]}) \quad \underset{\text{best fitting BL}}{\overset{\text{update}}{\longrightarrow}} \quad \hat{\eta}_{\nu}^{[\mathbf{m+1}]} \Longrightarrow \hat{\nu}^{[\mathbf{m+1}]} \ ,$$

Boosting for GAMLSS models

- Boosting was recently extended to risk functions with multiple components (Schmid et al., 2010)
- **Idea** ▷ Use partial derivatives instead of gradient

- Specify a **set of base-learners** — one base-learner per covariate
- Fit each of the base-learners **separately** to the partial derivatives
- **Cycle** through the partial derivatives within each boosting step

$$\frac{\partial \rho}{\partial \eta_{\boldsymbol{\mu}}}(y_i, \hat{\mu}^{[m]}, \hat{\sigma}^{[m]}, \hat{\nu}^{[m]}, \hat{\tau}^{[m]}) \quad \xrightarrow[\text{best fitting BL}]{\text{update}} \quad \hat{\eta}_{\mu}^{[\mathbf{m+1}]} \Longrightarrow \hat{\mu}^{[\mathbf{m+1}]} \;,$$

$$\frac{\partial \rho}{\partial \eta_{\boldsymbol{\sigma}}}(y_i, \hat{\mu}^{[\mathbf{m+1}]}, \hat{\sigma}^{[m]}, \hat{\nu}^{[m]}, \hat{\tau}^{[m]}) \quad \xrightarrow[\text{best fitting BL}]{\text{update}} \quad \hat{\eta}_{\sigma}^{[\mathbf{m+1}]} \Longrightarrow \hat{\sigma}^{[\mathbf{m+1}]} \;,$$

$$\frac{\partial \rho}{\partial \eta_{\boldsymbol{\nu}}}(y_i, \hat{\mu}^{[\mathbf{m+1}]}, \hat{\sigma}^{[\mathbf{m+1}]}, \hat{\nu}^{[m]}, \hat{\tau}^{[m]}) \quad \xrightarrow[\text{best fitting BL}]{\text{update}} \quad \hat{\eta}_{\nu}^{[\mathbf{m+1}]} \Longrightarrow \hat{\nu}^{[\mathbf{m+1}]} \;,$$

$$\frac{\partial \rho}{\partial \eta_{\boldsymbol{\tau}}}(y_i, \hat{\mu}^{[\mathbf{m+1}]}, \hat{\sigma}^{[\mathbf{m+1}]}, \hat{\nu}^{[\mathbf{m+1}]}, \hat{\tau}^{[m]}) \quad \xrightarrow[\text{best fitting BL}]{\text{update}} \quad \hat{\eta}_{\tau}^{[\mathbf{m+1}]} \Longrightarrow \hat{\tau}^{[\mathbf{m+1}]} \;.$$

Variable selection and shrinkage

- The main tuning parameter are the stopping iterations $m_{\text{stop},k}$. They control **variable selection** and the **amount of shrinkage**.
  - If boosting is stopped before convergence only the most important variables are included in the final model.
  - Variables that have never been selected in the updated step, are excluded.
  - Due to the small increments added in the update step, boosting incorporates shrinkage of effect sizes (compare to LASSO), leading to more stable predictions.
- For large $m_{\text{stop},k}$ boosting converges to the same solution as the original algorithm (in low-dimensional settings).
- The selection of $m_{\text{stop},k}$ is normally based on resampling methods, optimizing the predictive risk.

Data example: Munich rental guide

To deal with heteroscedasticity, we chose a three-parametric t-distribution with

$$\mathbb{E}(y) = \mu \qquad \text{and} \qquad \mathbb{V}\mathsf{ar}(y) = \sigma^2 \frac{\mathsf{df}}{\mathsf{df} - 2}$$

For each of the parameters μ, σ, and df, we consider the **candidate** predictors

$$\begin{aligned}
\eta_{\mu_i} &= \beta_{0\mu} + x_i^\top \beta_\mu + f_{1,\mu}(\mathsf{size}_i) + f_{2,\mu}(\mathsf{year}_i) + f_{\mathsf{spat},\mu}(s_i) \ , \\
\eta_{\sigma_i} &= \beta_{0\sigma} + x_i^\top \beta_\sigma + f_{1,\sigma}(\mathsf{size}_i) + f_{2,\sigma}(\mathsf{year}_i) + f_{\mathsf{spat},\sigma}(s_i) \ , \\
\eta_{\mathsf{df}_i} &= \beta_{0\mathsf{df}} + x_i^\top \beta_{\mathsf{df}} + f_{1,\mathsf{df}}(\mathsf{size}_i) + f_{2,\mathsf{df}}(\mathsf{year}_i) + f_{\mathsf{spat},\mathsf{df}}(s_i) \ .
\end{aligned}$$

Base-learners

- Categorical variables: Simple linear models
- Continuous variables: P-splines
- Spatial variable: Gaussian MRF (Markov random fields)

Package gamboostLSS

- Boosting for GAMLSS models is implemented in the R package **gamboostLSS** (▷ now available on CRAN).
- Package relies on the well tested and mature boosting package **mboost**.
- Lots of the **mboost** infrastructure is available in **gamboostLSS** as well (e.g., base-learners & convenience functions).

Package `gamboostLSS`

- Boosting for GAMLSS models is implemented in the R package **gamboostLSS** (▷ now available on CRAN).
- Package relies on the well tested and mature boosting package **mboost**.
- Lots of the **mboost** infrastructure is available in **gamboostLSS** as well (e.g., base-learners & convenience functions).

Now let's start and have a short look at some code!

Package gamboostLSS

```
> ## Install package mboost: (we use the R-Forge version as the
> ## bmrf base-learner is not yet included in the CRAN version)
> install.packages("mboost",
+                  repos = "http://r-forge.r-project.org")
> ## Install and load package gamboostLSS:
> install.packages("gamboostLSS")
> library("gamboostLSS")
```

(Simplified) code to fit the model

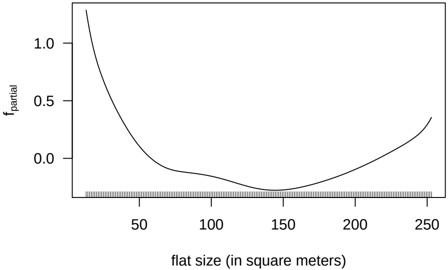
```
> ## Load data first, and load boundary file for spatial effects
> ## Now set up formula:
> form <- paste(names(data)[1], " ~ ",
                paste(names(data)[-c(1, 327, 328, 329)], collapse = " + "),
                " + bbs(wfl) + bbs(bamet) + bmrf(region, bnd = bound)")
> form <- as.formula(form)
> form

nmqms ~ erstbezg + dienstwg + gebmeist + gebgruen + hzkohojn +
    ... +
    bbs(wfl) + bbs(bamet) + bmrf(region, bnd = bound)

> ## Fit the model with (initially) 100 boosting steps
> mod <- gamboostLSS(formula = form, families = StudentTLSS(),
                     control = boost_control(mstop = 100,
                                             trace = TRUE),
                     baselearner = bols,
                     data = data)

[   1] .................................. -- risk: 3294.323
[  41] .................................. -- risk: 3091.206
[  81] ................
Final risk: 3038.919
```

(Simplified) code to fit the model (ctd.)

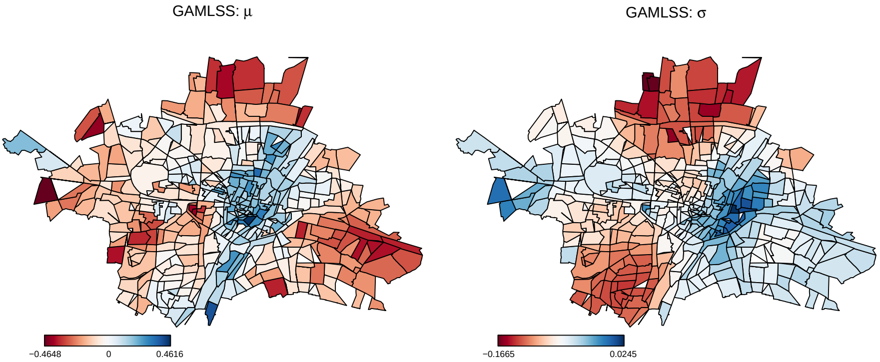
```
> ## optimal number of boosting iterations fund by 3-dimensional
> ## cross-validation on a logarithmic grid resulted in
> ## 750 (mu), 108 (sigma), 235 (df) steps;
> ## Let model run until these values:
> mod[c(750, 108, 235)]
>
> ## Let's look at the number of variables per parameter:
> sel <- selected(mod)
> lapply(sel, function(x) length(unique(x)))

$mu
[1] 115

$sigma
[1] 31

$df
[1] 7

> ## (Very) sparse model (only 115, 31 and 5 base-learners out of 328)
```

(Simplified) code to fit the model (ctd.)

```
> ## Now we can look at the estimated parameters
> ## e.g., the effect of roof terrace on the mean
> coef(mod, which = "dterasn", parameter = "mu")

$`bols(dterasn)`
 (Intercept)      dterasn
-0.004254606  0.293792997

> ## We can also easily plot the estimated smooth effects:
> plot(mod, which = "bbs(wfl)", parameter = "mu",
+      xlab = "flat size (in square meters)", type = "l")
```

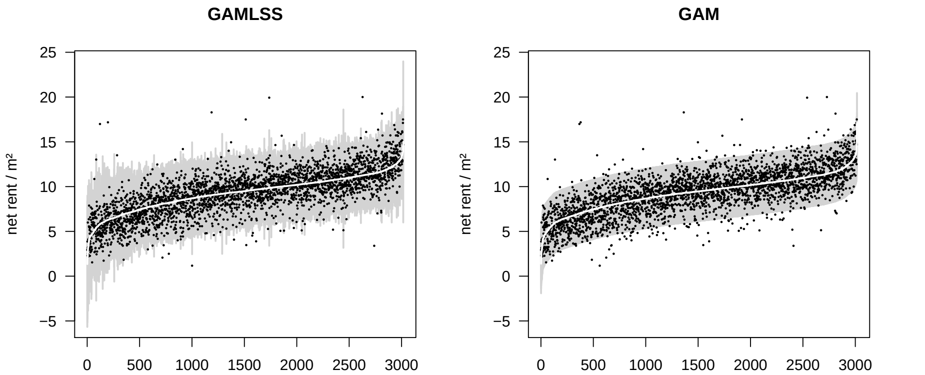


Results: spatial effects



Estimated spatial effects obtained for the high-dimensional GAMLSS for distribution parameters μ and σ. For the third parameter `df`, the corresponding variable was not selected.

Results: prediction intervals



95% prediction intervals based on the quantiles of the modeled conditional distribution. Coverage probability GAMLSS 93.93% (92.07-95.80); coverage probability GAM 92.23% (89.45-94.32).

Summary

- As **gamboostLSS** relies on **mboost**, we have a **well tested, mature back end**.
- The base-learners offer great flexibility when it comes to the **type of effects** (linear, non-linear, spatial, random, monotonic, . . .).
- Boosting is **feasible even if** $p \gg n$.
- **Variable selection** is included in the fitting process. Additional shrinkage leads to more stable results.

The algorithm is implemented in the **R** add-on package **gamboostLSS**
▷ **now** available on CRAN.

Further literature

- ▷ Mayr, A., N. Fenske, B. Hofner, T. Kneib and M. Schmid, (2010). GAMLSS for high-dimensional data – a flexible approach based on boosting. *Journal of the Royal Statistical Society, Series C (Applied Statistics)*, accepted.
- ▷ B. Hofner, A. Mayr, N. Fenske, and M. Schmid, (2010). gamboostLSS: Boosting Methods for GAMLSS Models. R package version 1.0-1.

- Bühlmann, P. and Hothorn, T., (2007). Boosting algorithms: Regularization, prediction and model fitting. *Statistical Science*, **22**, 477-522.
- Rigby, R. A. and Stasinopoulos, D. M., (2005). Generalized additive models for location, scale and shape. *Applied Statistics*, **54**, 507-554.
- Schmid, M., S. Potapov, A. Pfahlberg and T. Hothorn, (2010). Estimation and regularization techniques for regression models with multidimensional prediction functions . *Statistics and Computing*, **20**, 139-150.