

Structured Additive Regression Models: An R Interface to BayesX

Nikolaus Umlauf, Thomas Kneib, Stefan Lang,
Achim Zeileis

<http://eeecon.uibk.ac.at/~umlauf/>

Overview

- Introduction
- Structured Additive Regression Models (STAR)
- Installing the **BayesX** binary
- The main model fitting function
- More components of the interface
- Available additive terms
- Illustration
- Outlook
- References

Introduction: What is BayesX?

The free software **BayesX** is a standalone program comprising powerful tools for Bayesian and mixed model based inference in complex semiparametric regression models with structured additive predictor (STAR).

- Generalized additive models (GAM).
- Generalized additive mixed models (GAMM).
- Generalized geosadditive mixed models (GGAMM).
- Dynamic models.
- Varying coefficient models (VCM).
- Geographically weighted regression.

BayesX is written in C++ and utilizes numerically efficient (sparse) matrix architectures.

Introduction: What is BayesX?

In **BayesX**, estimation of regression parameters is based on three inferential concepts:

- ➊ Full Bayesian inference via MCMC.
- ➋ Inference via a mixed model representation.
- ➌ Penalized likelihood including variable selection.

BayesX provides functionality for the following types of responses:

- Univariate exponential family.
- Categorical responses with unordered responses.
- Categorical responses with ordered responses.
- Continuous time survival models.
- Continuous time multi-state models.

Introduction: The R interface

Problems: **BayesX** only provides limited functionality for

- handling/manipulating data sets,
- handling/manipulating geographical maps,
- exploring/visualizing estimation results.

Therefore, the R package **BayesX** (available at CRAN) was developed, which provides functionality for exploring and visualizing estimation results.

However, estimating models from **BayesX** with special program files and handling estimation outputs within R is still time consuming and not straightforward.

Introduction: The R interface

Now: New interface package **R2BayesX** for

- specifying/estimating STAR models with **BayesX** directly from R,
- standard methods and extractor functions for **BayesX** fitted model objects, e.g. producing high level graphics of estimated effects, model diagnostic plots, summary statistics and more.

In addition:

- Run already existing **BayesX** input program files from R.
- Automatically import **BayesX** output files into R.

To install the package directly within R type:

```
install.packages("R2BayesX",  
  repos = "http://R-Forge.R-project.org")
```

Introduction: Example

Dataset on malnutrition in Zambia:

The main interest is to model the dependence of `stunting` of newborn children on covariates including

- the age of the child in months (`agechild`),
- the mother's bmi (`mbmi`)
- and the `district` the mother lives in.

We start with the following model:

$$\text{stunting}_i = \gamma_0 + f_1(\text{agechild}_i) + f_2(\text{mbmi}_i) + f_{\text{spat}}(\text{district}_i) + \varepsilon_i,$$

with $\varepsilon_i \sim N(0, \sigma^2)$.

Introduction: Example

Loading the data and boundary object

```
R> data("ZambiaNutrition", "ZambiaBnd", package = "R2BayesX")
```

The model is specified using R's formula language definition

```
R> f <- stunting ~ sx(agechild) + sx(mbmi) + sx(district,  
+      bs = "gk", map = ZambiaBnd, full = TRUE)
```

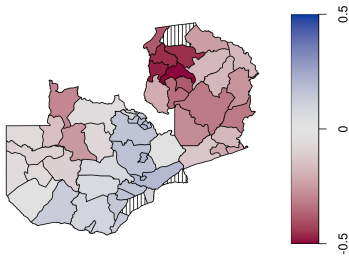
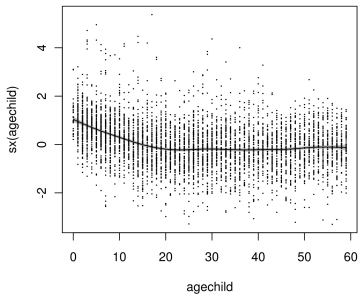
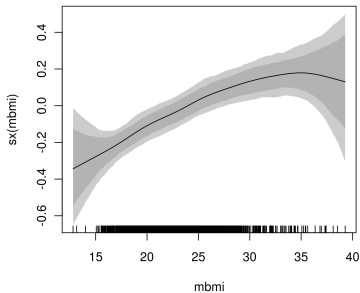
estimated by

```
R> b <- bayesx(f, family = "gaussian", method = "MCMC",  
+      data = ZambiaNutrition)
```

and plotted, e.g. by typing

```
R> plot(b, map = ZambiaBnd)
```


Introduction: Example



STAR models

Distributional and structural assumptions, given covariates and parameters, are based on generalized linear models with

$$E(y|\mathbf{x}, \mathbf{z}, \boldsymbol{\gamma}, \boldsymbol{\theta}) = h^{-1}(\eta)$$

and structured additive predictor

$$\eta = f_1(\mathbf{z}) + \dots + f_p(\mathbf{z}) + \mathbf{x}'\boldsymbol{\gamma}$$

- $\mathbf{x}'\boldsymbol{\gamma}$ parametric part of the predictor.
- \mathbf{z} represents a generic vector of all nonlinear modeled covariates, e.g. may include continuous covariates, time scales, location or unit or cluster indexes.
- The vector $\boldsymbol{\theta}$ comprises all parameters of the functions f_1, \dots, f_p .
- f_j one-/two-/higher-dimensional, not necessarily continuous functions.

STAR models: Modeling the functions f_j

The functions f_j are possibly smooth functions comprising effects (and combinations) as e.g. given by:

- Nonlinear effects of continuous covariates.
- Two-dimensional surfaces.
- Spatially correlated effects.
- Varying coefficients.
- Spatially varying effects.
- Random intercepts.
- Random slopes.

STAR models: General form

- Vector of function evaluations $\mathbf{f}_j = (f_j(\mathbf{z}_1), \dots, f_j(\mathbf{z}_n))$ of the $i = 1, \dots, n$ observations can be written in matrix notation

$$\mathbf{f}_j = \mathbf{Z}_j \beta_j,$$

with \mathbf{Z}_j as the design matrix, where β_j are unknown regression coefficients.

- Form of \mathbf{Z}_j only depends on the functional type chosen.
- Penalized least squares:

$$\text{PLS}(\beta, \gamma) = \|\mathbf{y} - \boldsymbol{\eta}\|^2 + \lambda_1 \beta_1' \mathbf{K}_1 \beta_1 + \dots + \lambda_p \beta_p' \mathbf{K}_p \beta_p$$

STAR models: General form

- Prior for β in the corresponding Bayesian approach

$$p(\beta_j | \tau_j^2) \propto \exp \left(-\frac{1}{2\tau_j^2} \beta_j' \mathbf{K}_j \beta_j \right),$$

τ_j^2 variance parameter, governs the smoothness of f_j .

- Structure of \mathbf{K}_j also depends on the type of covariates and on assumptions about smoothness of \mathbf{f}_j .
- The variance parameter τ_j^2 is equivalent to the inverse smoothing parameter in a frequentist approach. Utilizing mixed model technology, restricted maximum likelihood (REML) forms a basis for determination. From a Bayesian perspective, this yields empirical Bayes/posterior mode estimates for the STAR models.

Installing the BayesX binary

For model fitting with package **R2BayesX**, the binary command-line version of **BayesX** needs to be installed and linked to R first.

```
R> library("R2BayesX")  
R> install.bayesx(inst.dir = "/path/to/bin", source.dir = NULL)
```

Then by setting e.g.

```
R> options(bayesx.bin = "/path/to/bin/BayesX")
```

the model fitting function of **R2BayesX** will know the location of the binary using function `getOption()`.

Note: on Windows, function `install.bayesx()` will download and execute an installer, which also installs the GUI version of **BayesX**.

The main model fitting function

The arguments of the main model fitting function are

```
bayesx(formula, data, weights = NULL, subset = NULL,  
        offset = NULL, na.action = na.fail, contrasts = NULL,  
        family = "gaussian", method = "MCMC",  
        control = bayesx.control(...), ...)
```

Families:

```
"binomial", "binomialprobit", "gamma", "gaussian",  
"multinomial", "poisson", "cox", "cumprobit", "multistate",  
"binomialcomploglog", "cumlogit", "multinomialcatsp",  
"multinomialprobit", "seqlogit", "seqprobit".
```

Methods:

```
"MCMC", "REML", "STEP".
```

Note: family objects are currently not supported.

More components of the interface

Internally, function `bayesx()` calls the following functions:

- 1 `parse.bayesx.input()`
- 2 `write.bayesx.input()`
- 3 `run.bayesx()`
- 4 `read.bayesx.output()`

These functions are operating independently and may also be called by the R user.

The functionality is especially helpful for already existing **BayesX** program and output files.

Moreover, function `read.bayesx.output()` also returns objects of class "bayesx".

Available additive terms

The main model term constructor function is function `sx()`, with arguments:

```
sx(x, z = NULL, bs = "ps", by = NA, ...)
```

`sx()` is simply an interface to function `s()` from package **mgcv**.

```
s(..., k = -1, bs = "ps", m = NA, by = NA, xt = NULL)
```

Random effects are included in the models using function `r()`.

```
r(id, by = NA, xt = NULL)
```

Basis/term types:

```
"rw1", "rw2", "season", "ps" ("psplinerw1", "psplinerw2"),  
"te" ("pspline2dimrw1"), "kr" ("kriging"), "gk"  
("geokriging"), "gs" ("geospline"), "mrf" ("spatial"), "bl"  
("baseline"), "factor", "ridge", "lasso", "nigmix".
```

Available additive terms

Additional options within “...” and xt for each basis/term type and method may be looked up using function `bayes.term.options()`, e.g.

```
R> bayesx.term.options(bs = "ps", method = "MCMC")
```

possible options for 'bs = "ps"':

degree: the degree of the B-spline basis functions.

Default: integer, 'degree = 3'.

knots: number of inner knots.

Default: integer, 'knots = 20'.

order: only if 'bs = "ps"', the order of the difference penalty.

Default: integer, 'order = 2'.

.
.
.

Illustration

Following Kandala, Lang, Klasen and Fahrmeir (2001), the task is to model **stunting** of newborn children on the following covariates:

Variable	Description
stunting	Standardized Z-score for stunting.
mbmi	Body mass index of the mother.
agechild	Age of the child in months.
district	District where the mother lives.
memployment	Is the mother employed?
meducation	Mother's educational status.
urban	Is the domicile in an urban region?
gender	Gender of the child.

The predictor of the STAR model is given by

$$\eta = \gamma_0 + \gamma_1 \text{memploymentyes} + \gamma_2 \text{urbanno} + \gamma_3 \text{genderfemale} + \gamma_4 \text{meducationno} + \gamma_5 \text{meducationprimary} + f_1(\text{mbmi}) + f_2(\text{agechild}) + f_{str}(\text{district}) + f_{unstr}(\text{district})$$

Illustration

The formula is set with

```
R> f <- stunting ~ memployment + urban + gender + meducation +  
+      sx(mbmi) + sx(agechild) + sx(district, bs = "mrf",  
+      map = ZambiaBnd) + r(district)
```

The model is then fitted using MCMC by calling

```
R> set.seed(321)  
R> zm <- bayesx(f, family = "gaussian", method = "MCMC",  
+      data = ZambiaNutrition, iterations = 12000, burnin = 2000,  
+      step = 10)
```

Model summary

```
R> summary(zm)
```

Illustration

Call:

```
bayesx(formula = f, data = ZambiaNutrition, family = "gaussian",  
        method = "MCMC", iterations = 12000, burnin = 2000, step = 10)
```

Fixed effects estimation results:

Parametric Coefficients:

	Mean	Sd	2.5%	50%	97.5%
(Intercept)	0.0991	0.0475	0.0046	0.1018	0.1863
memploymentno	-0.0084	0.0135	-0.0359	-0.0084	0.0170
urbanno	-0.0895	0.0217	-0.1306	-0.0893	-0.0450
genderfemale	0.0582	0.0133	0.0320	0.0578	0.0850
meducationno	-0.1722	0.0269	-0.2248	-0.1719	-0.1163
meducationprimary	-0.0611	0.0262	-0.1115	-0.0614	-0.0091

Smooth terms variances:

	Mean	Sd	2.5%	50%	97.5%	Min	Max
sx(agechild)	0.0062	0.0060	0.0014	0.0042	0.0233	0.0007	0.0570
sx(district)	0.0360	0.0191	0.0094	0.0325	0.0813	0.0025	0.1784
sx(mbmi)	0.0019	0.0028	0.0003	0.0011	0.0081	0.0002	0.0468

Illustration

Random effects variances:

	Mean	Sd	2.5%	50%	97.5%	Min	Max
r(district)	0.0076	0.0064	0.0008	0.0062	0.0226	0.0003	0.0701

Scale estimate:

	Mean	Sd	2.5%	50%	97.5%
Sigma2	0.8023	0.0163	0.7721	0.8017	0.836

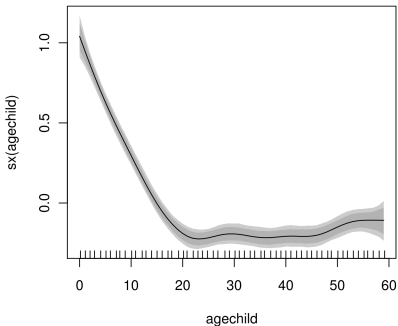
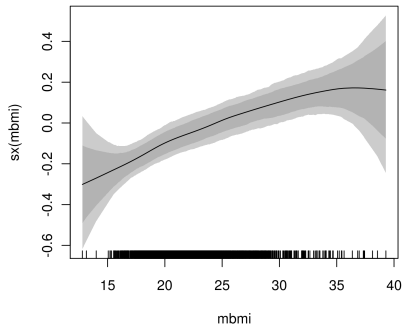
N = 4847 burnin = 2000 DIC = 4899.506 pd = 50.41262

method = MCMC family = gaussian iterations = 12000 step = 10

Illustration

Plotting of specific terms

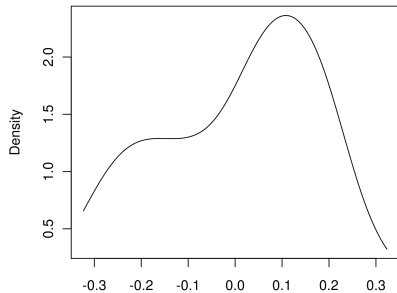
```
R> plot(zm, term = c("sx(mbmi)", "sx(agechild)"))
```



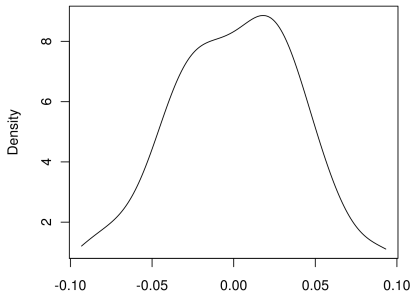
Illustration

Spatial effects, kernel density estimates

```
R> plot(zm, term = c("sx(district)", "r(district)"))
```



N = 58 Bandwidth = 0.06664

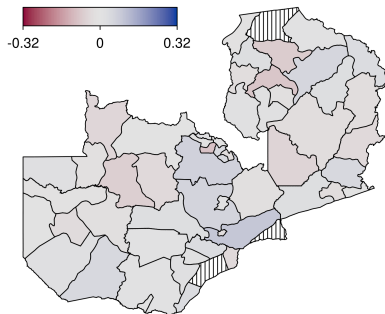
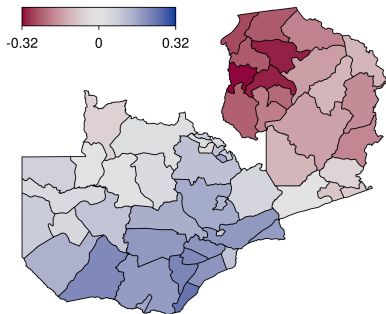


N = 55 Bandwidth = 0.01658

Illustration

Spatial effects, map effect plots

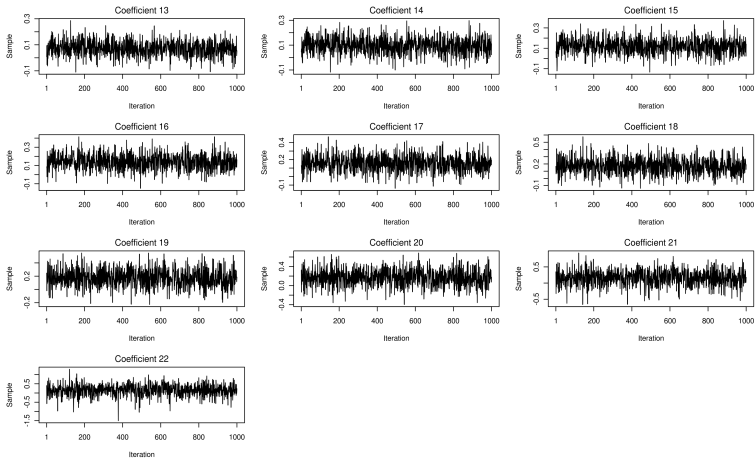
```
R> plot(zm, term = "sx(district)", map = ZambiaBnd)  
R> range <- c(-0.32, 0.32)  
R> plot(zm, term = "r(district)", map = ZambiaBnd, range = range,  
+       lrange = range)
```



Illustration

Diagnostic plots, sampling paths

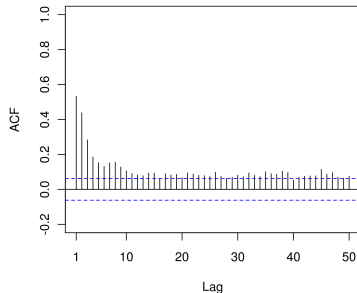
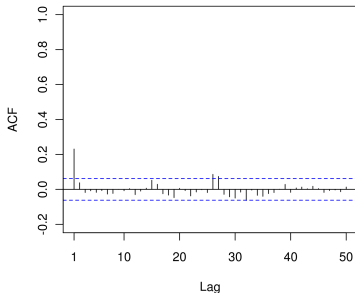
```
R> plot(zm, term = "sx(mbmi)", which = "coef-samples")
```



Illustration

Diagnostic plots, autocorrelation functions and maximum autocorrelation of parameters

```
R> plot(zm, term = "sx(mbmi)", which = "var-samples", acf = TRUE)  
R> plot(zm, which = "max-acf")
```



Further inspection through extractor function `samples()`, e.g. with package **coda** is possible.

Illustration

Inspecting the log-file of the **BayesX** binary

```
R> bayesx_logfile(zm)

> bayesreg b
> map ZambiaBnd
> ZambiaBnd.infile using /tmp/Rtmpa3Z6WF/bayesx/ZambiaBnd.bnd
NOTE: 57 regions read from file /tmp/Rtmpa3Z6WF/bayesx/ZambiaBnd.bnd
> dataset d
> d.infile using /tmp/Rtmpa3Z6WF/bayesx/bayesx.estim.data.raw
NOTE: 14 variables with 4847 observations read from file
/tmp/Rtmpa3Z6WF/bayesx/bayesx.estim.data.raw

> b.outfile = /tmp/Rtmpa3Z6WF/bayesx/bayesx.estim
> b.regress stunting = mbmi(psplinerw2,nrknots=20,degree=3) +
  agechild(psplinerw2,nrknots=20,degree=3) +
  district(spatial,map=ZambiaBnd) + district(random) + memploymentyes +
  urbanno + genderfemale + meducationno + meducationprimary,
  family=gaussian iterations=12000 burnin=2000 step=10 setseed=123
predict using d
.
.
.
```

Outlook

- Beta testing and bug fixing.
- Facilitate installation of **BayesX** binary across platforms.
- Release the package through CRAN.
- Enhance functionality of the package, i.e. support spatial objects (e.g. from **sp**), more options for visualization etc.

The slides together with a package vignette, the R code and demos are available at:

<http://bayesr.R-Forge.R-project.org/>

References

Belitz C, Brezger A, Kneib T, Lang S (2011). **BayesX** – Software for Bayesian Inference in Structured Additive Regression. Models. Version 2.0.1.

URL <http://www.stat.uni-muenchen.de/~bayesx/>

Brezger A, Kneib T, Lang S (2005). “**BayesX**: Analyzing Bayesian Structured Additive Regression Models”. *Journal of Statistical Software*, **14**(11), 1–22.

URL <http://www.jstatsoft.org/v14/i11/>

Fahrmeir L, Kneib T, Lang S (2009). *Regression – Modelle, Methoden und Anwendungen*. 2nd edition. Springer, Berlin.

Kandala NB, Lang S, Klasen S, Fahrmeir L (2001). “Semiparametric Analysis of the Socio-Demographic and Spatial Determinants of Undernutrition in Two African Countries”. *Research in Official Statistics*, **1**, 81–100.

Wood SN (2006). *Generalized Additive Models: An Introduction with R*. Chapman & Hall/CRC, Boca Raton.

Wood SN (2011). **mgcv**: GAMs with GCV/AIC/REML Smoothness Estimation and GAMMs by PQL. R package version 1.7-6. URL <http://CRAN.R-project.org/package=mgcv>