An Introduction to the Estimation of GPLMs and Data Examples for the R `gplm` Package

Marlene Müller

April 2014

Contents

1 Introduction to (Generalized) Partial Linear Models

The generalized linear model (GLM) is a regression model that can be written as

$$E(Y|\boldsymbol{X}) = G(\boldsymbol{X}^T\beta),$$

where Y is the dependent variable, $\boldsymbol{X}$ a vector of explanatory variables, $\boldsymbol{\beta}$ the unknown parameter vector and $G(\bullet)$ a known function (the inverse link function). The *generalized partial linear model* (GPLM) extends this GLM by a nonparametric component:

$$E(Y|\boldsymbol{X},\boldsymbol{T}) = G\{\boldsymbol{X}^T\beta + m(\boldsymbol{T})\}.$$

Here, it is assumed that the explanatory variables split into two vectors, $\boldsymbol{X}$ and $\boldsymbol{T}$. The vector $\boldsymbol{X}$ denotes a p-variate random vector which typically covers discrete covariables or variables that are known to influence the index in a linear way. The vector $\boldsymbol{T}$ is a q-variate random vector of continuous covariables to be modeled in a nonparametric way. As before, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^T$ denotes a finite dimensional unknown parameter, while $m(\bullet)$ is an unknown q-variate smooth function.

As for the GLM, the dependent variable Y may originate from an exponential family. This means to assume that the variance $\mathrm{Var}(Y|\boldsymbol{X},\boldsymbol{T})$ may depend on the predictor $\boldsymbol{X}^T\beta + m(T)$ and on a dispersion parameter σ^2, i.e.

$$\mathrm{Var}(Y|\boldsymbol{X},\boldsymbol{T}) = \sigma^2 V\left(G\{X^T\beta + m(T)\}\right).$$

Possible distributions for Y are for example (McCullagh and Nelder; 1989):

- discrete distributions: Bernoulli, Binomial, Poisson, Geometric, Negative Binomial,
- continuous distributions: Normal, Exponential, Gamma, Inverse Normal.

It is easy to see that GPLM covers other semi-non-parametric models, as for example:

- the *partial linear model* (PLM), i.e. $Y = \boldsymbol{X}^T\boldsymbol{\beta} + m(\boldsymbol{T}) + \varepsilon$ with $\varepsilon \sim N(0, \sigma^2)$ implying $E(Y|\boldsymbol{X},\boldsymbol{T}) = \boldsymbol{X}^T\boldsymbol{\beta} + m(\boldsymbol{T})$ and $\mathrm{Var}(Y|\boldsymbol{X},\boldsymbol{T}) = \sigma^2$, (Speckman; 1988; Robinson; 1988).
- the *generalized additive model* (GAM) with a linear and a single nonparametric component function, i.e. $E(Y|\boldsymbol{X},\boldsymbol{T}) = G\{\boldsymbol{X}^T\beta + m(\boldsymbol{T})\}$ (Hastie and Tibshirani; 1990).

2 Estimation Methods

The estimation approaches for the GPLM are essentially based on the idea that an estimate $\widehat{\boldsymbol{\beta}}$ can be found for known $m(\bullet)$, and an estimate $\widehat{m}(\bullet)$ can be found for known $\boldsymbol{\beta}$. This section introduces two different estimation methods, a *generalized Speckman* estimator and the classical *backfitting* approach. A more detailed presentation of these methods can be found in Müller (2001) and Härdle et al. (2004, Chapters 4, 5 and 7).

Recall the first two conditional moments of Y are specified as

$$\begin{aligned} E(Y|\boldsymbol{X},\boldsymbol{T}) &= \mu = G(\eta) = G\{\boldsymbol{X}^T\boldsymbol{\beta} + m(\boldsymbol{T})\}, \\ \mathrm{Var}(Y|\boldsymbol{X},\boldsymbol{T}) &= \sigma^2 V(\mu). \end{aligned}$$

We will now discuss the estimation of $\boldsymbol{\beta}$ and $m(\bullet)$ by means of the sample values $\{y_i, \boldsymbol{x}_i, \boldsymbol{t}_i\}$, $i = 1, \ldots, n$. It should be pointed out that we focus on the estimation of $\boldsymbol{\beta}$ and $m(\bullet)$. The additional scale parameter σ can be obtained from

$$\widehat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^{n} \frac{(y_i - \widehat{\mu}_i)^2}{V(\widehat{\mu}_i)}, \tag{1}$$

when we denote $\widehat{\mu}_i = G(\widehat{\eta}_i)$ estimated from $\widehat{\eta}_i = \boldsymbol{x}_i^T \widehat{\boldsymbol{\beta}} + \widehat{m}(\boldsymbol{t}_i)$.

2.1 Speckman Type Estimation

The Speckman estimator (Speckman; 1988) was originally derived for the PLM. In our setup that means to consider a model with the identity link G and normally distributed error terms:

$$Y = \boldsymbol{X}^T \beta + m(\boldsymbol{T}) + \varepsilon\,, \quad \varepsilon \sim N(0, \sigma^2)\,.$$

Taking the conditional expectation w.r.t. $\boldsymbol{T}$ and differencing the two equations leads to (Härdle et al.; 2004, Section 7.1):

$$\underbrace{Y - E(Y|\boldsymbol{T})}_{\widetilde{Y}} = \big\{ \underbrace{\boldsymbol{X} - E(\boldsymbol{X}|\boldsymbol{T})}_{\widetilde{\boldsymbol{X}}} \big\}^{\top} \boldsymbol{\beta} + \underbrace{\varepsilon - E(\varepsilon|\boldsymbol{T})}_{\widetilde{\varepsilon}},$$

i.e. a modified regression equation in $\widetilde{\boldsymbol{X}}$ and $\widetilde{Y}$ that allows to separately estimate the parameter vector $\boldsymbol{\beta}$. The modified variables $\widetilde{\boldsymbol{X}}$ and $\widetilde{Y}$ are calculated using the fact that the conditional expectation $E(\bullet|\boldsymbol{T})$ can be estimated through a (nonparametric) regression on the explanatory variable $\boldsymbol{T}$.

To formulate the estimation procedure we introduce the following terms based on the sample values $\{y_i, \boldsymbol{x}_i, \boldsymbol{t}_i\}$:

$$\boldsymbol{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}, \quad \mathcal{X} = \begin{pmatrix} x_{11} & \ldots & x_{1p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \ldots & x_{np} \end{pmatrix}, \quad \boldsymbol{m} = \begin{pmatrix} m(t_1) \\ \vdots \\ m(t_n) \end{pmatrix}$$

If we denote the regression operator (hat matrix) by $\mathcal{S}$, then Speckman's estimation method for the PLM can be summarized as follows:

Speckman estimation for the PLM

- *calculate*
$$\widetilde{\mathcal{X}} = (\mathcal{I} - \mathcal{S})\mathcal{X} \quad \text{and} \quad \widetilde{\boldsymbol{y}} = (\mathcal{I} - \mathcal{S})\boldsymbol{y}$$
- *estimate* $\boldsymbol{\beta}$
$$\widehat{\boldsymbol{\beta}} = (\widetilde{\mathcal{X}}^T \widetilde{\mathcal{X}})^{-1} \widetilde{\mathcal{X}}^T \widetilde{\boldsymbol{y}}$$
- *estimate* $\boldsymbol{m}$
$$\widehat{\boldsymbol{m}} = \mathcal{S}(\boldsymbol{y} - \mathcal{X}\widehat{\boldsymbol{\beta}})$$

Optionally, the estimation steps for $\boldsymbol{\beta}$ and $\boldsymbol{m}$ can be used as updating step in an iteration up to convergence.

In case of a Nadaraya–Watson kernel type regression (Härdle et al.; 2004, Section 4.1), we consider the smoother matrix $\mathcal{S}$ with elements

$$\mathcal{S}_{ij} = \frac{\mathcal{K}_{\boldsymbol{h}}(\boldsymbol{t}_i - \boldsymbol{t}_j)}{\sum_{k=1}^{n} \mathcal{K}_{\boldsymbol{h}}(\boldsymbol{t}_k - \boldsymbol{t}_j)}, \tag{2}$$

where $\mathcal{K}$ denotes a multidimensional kernel function and $\boldsymbol{h}$ the respective bandwidth vector (the dimension being equal to the dimension of $\boldsymbol{T}$). By $\mathcal{K}_{\boldsymbol{h}}$ we abbreviate the componentwise rescaled kernel function

$$\mathcal{K}_{\boldsymbol{h}}(\boldsymbol{u}) = \frac{1}{h_1 \cdot \ldots \cdot h_q} \mathcal{K}\left(\frac{u_1}{h_1}, \ldots, \frac{u_q}{h_q}\right)$$

for q-dimensional vectors $\boldsymbol{u}$ and $\boldsymbol{h}$. (See Subsection 2.3.2 for more details on possible kernel functions.) For the modified regression terms this leads to the calculation of

$$\widetilde{\boldsymbol{x}}_j = \boldsymbol{x}_j - \frac{\sum_{i=1}^{n} \mathcal{K}_{\boldsymbol{h}}(\boldsymbol{t}_i - \boldsymbol{t}_j)\, \boldsymbol{x}_i}{\sum_{k=1}^{n} \mathcal{K}_{\boldsymbol{h}}(\boldsymbol{t}_k - \boldsymbol{t}_j)} \quad \text{and} \quad \widetilde{y}_j = y_j - \frac{\sum_{i=1}^{n} \mathcal{K}_{\boldsymbol{h}}(\boldsymbol{t}_i - \boldsymbol{t}_j)\, y_i}{\sum_{k=1}^{n} \mathcal{K}_{\boldsymbol{h}}(\boldsymbol{t}_k - \boldsymbol{t}_j)} \quad \text{for } j = 1, \ldots, n\,.$$

The nonparametric component function of the PLM is thus estimated by

$$\widehat{m}_j = \widehat{m}(\boldsymbol{t}_j) = \frac{\sum_{i=1}^{n} \mathcal{K}_{\boldsymbol{h}}(\boldsymbol{t}_i - \boldsymbol{t}_j)\, (y_i - \boldsymbol{x}_i^T \widehat{\boldsymbol{\beta}})}{\sum_{i=1}^{n} \mathcal{K}_{\boldsymbol{h}}(\boldsymbol{t}_i - \boldsymbol{t}_j)} \quad \Longleftrightarrow \quad \widehat{\boldsymbol{m}} = \mathcal{S}(\boldsymbol{y} - \mathcal{X}\widehat{\boldsymbol{\beta}})\,. \tag{3}$$

In an analogous way, estimates of $m(\bullet)$ can be calculated at arbitrary design points by replacing $\boldsymbol{t}_j$ in (3).

In the context of a generalized partial linear model (GPLM) we have to take the function G (inverse link function) and the distribution of Y into account. The estimation method thus combines the estimation algorithm in a GLM (IRLS = iteratively reweighted least squares, see McCullagh and Nelder; 1989; Müller; 2012, Chapter III.24) and the Speckman approach for the PLM.

Generalized linear models (GLMs) are estimated by maximum likelihood (or equivalently minimum deviance) estimation. We now denote the individual log-likelihood in a GPLM by

$$\ell(\mu_i, y_i) = \ell\left(G\left\{\boldsymbol{x}_i^T \beta + m(\boldsymbol{t}_i)\right\}\,,\, y_i\right)$$

such that estimating for the full sample means to maximize $\ell(\boldsymbol{\mu}, \boldsymbol{y}) = \sum_{i=1}^{n} \ell(\mu_i, y_i)$. The difference between the GLM and the GPLM is just the nonparametric part $m(\boldsymbol{t}_i)$.

In a GLM, the parameter vector $\boldsymbol{\beta}$ would be estimated by the IRLS algorithm through an iterative updating $\widehat{\boldsymbol{\beta}}^{new} = (\mathcal{X}^T \mathcal{W} \mathcal{X})^{-1} \mathcal{X}^T \mathcal{W} \boldsymbol{z}$, where $\boldsymbol{z}$ denotes the adjusted dependent variable vector $\boldsymbol{z} = \mathcal{X}\widehat{\boldsymbol{\beta}} + \mathcal{W}^{-1}\boldsymbol{v}$. The vector $\boldsymbol{v}$ denotes the vector of the first derivatives of $\ell(\boldsymbol{\mu}, \boldsymbol{y})$ w.r.t. the indices $\eta_i = \boldsymbol{x}_i^\top \boldsymbol{\beta}$ evaluated at $\boldsymbol{x}_i^T \widehat{\boldsymbol{\beta}}$. Analogously, $\mathcal{W} = \mathrm{diag}(w_{11}, \ldots, w_{nn})$ is the diagonal matrix containing the corresponding second derivatives.

In the case of a GPLM, the adjusted dependent variable is extended by the nonparametric component:

$$\boldsymbol{z} = \mathcal{X}\widehat{\boldsymbol{\beta}} + \widehat{\boldsymbol{m}} - \mathcal{W}^{-1}\boldsymbol{v}, \tag{4}$$

where $\boldsymbol{v}$ and $\mathcal{W}$ are defined as before but now evaluated at $\boldsymbol{x}_i^T\widehat{\boldsymbol{\beta}} + \widehat{m}_i$. The elements of the smoother matrix $\mathcal{S}$ have to be modified to

$$\mathcal{S}_{ij} = \frac{\mathcal{K}_{\mathbf{H}}(\boldsymbol{t}_i - \boldsymbol{t}_j)\, w_{ii}}{\sum_{i=1}^{n} \mathcal{K}_{\mathbf{H}}(\boldsymbol{t}_i - \boldsymbol{t}_j)\, w_{ii}}, \tag{5}$$

where we recall that w_{ii} is the i-th diagonal element of $\mathcal{W}$.

The iterative Speckman type estimation algorithm for the GPLM can then be summarized by the following calculation in each iteration step:

Generalized Speckman estimation for the GPLM

- *calculate (in each step)*

$$\widetilde{\mathcal{X}} = (\mathcal{I} - \mathcal{S})\mathcal{X} \quad \text{and} \quad \widetilde{\boldsymbol{z}} = \widetilde{\mathcal{X}}\,\widehat{\boldsymbol{\beta}} - \mathcal{W}^{-1}\boldsymbol{v}$$

- *updating step for* $\boldsymbol{\beta}$

$$\widehat{\boldsymbol{\beta}}^{new} = \left(\widetilde{\mathcal{X}}^T\mathcal{W}\widetilde{\mathcal{X}}\right)^{-1}\widetilde{\mathcal{X}}^T\mathcal{W}\widetilde{\boldsymbol{z}}$$

- *updating step for* $\boldsymbol{m}$

$$\widehat{\boldsymbol{m}}^{new} = \mathcal{S}\left(\widetilde{\boldsymbol{z}} - \mathcal{X}\widehat{\boldsymbol{\beta}}^{new}\right)$$

It is easy to see, that Speckman's method for the PLM is a special case of this generalized algorithm: In the case of a PLM with normal errors $\varepsilon \sim N(0, \sigma^2)$ we find the derivatives of the log-likelihoods as $v_i = -(y_i - \boldsymbol{x}_i^T\boldsymbol{\beta} - m_j)/\sigma^2$ and $w_{ii} \equiv -1/\sigma^2$. Thus the adjusted dependent variable $\widetilde{\boldsymbol{z}}$ equals $\widetilde{\boldsymbol{y}}$ and the elements of $\mathcal{W}$ cancel out in $\widehat{\boldsymbol{\beta}}^{new}$ and $\mathcal{S}$.

Some further calculations (e.g. Müller; 2001) show that $\mathcal{X}\widehat{\boldsymbol{\beta}} + \widehat{\boldsymbol{m}} = \mathcal{R}^S\boldsymbol{z}$ with

$$\mathcal{R}^S = \widetilde{\mathcal{X}}\left(\widetilde{\mathcal{X}}^T\mathcal{W}\widetilde{\mathcal{X}}\right)^{-1}\widetilde{\mathcal{X}}^T\mathcal{W}(\mathcal{I} - \mathcal{S}) + \mathcal{S}. \tag{6}$$

where all terms are evaluated at the estimates at convergence. This means that the estimation method is linear (in $\boldsymbol{z}$), thus the trace of the hat matrix $\mathcal{R}^S$ can be used to approximate the degrees of freedom of the model estimate:

$$df^{res} = n - \operatorname{trace}\left(\mathcal{R}^S\right)$$

(see e.g. Hastie and Tibshirani; 1990, for an analogous calculation in the backfitting case).

2.2 Backfitting Estimation

The backfitting method has been suggested as an iterative algorithm to fit an additive model (see Buja et al.; 1989; Hastie and Tibshirani; 1990). Its main idea is to regress the additive

components separately on partial residuals. The PLM is a again special case, consisting of only two additive functions. We denote now by $\mathcal{P}$ the projection matrix $\mathcal{P} = \mathcal{X}(\mathcal{X}^T\mathcal{X})^{-1}\mathcal{X}^T$ from a linear regression model and by $\mathcal{S}$ the smoother matrix as before. Then backfitting means to resolve

$$\begin{aligned} \mathcal{X}\widehat{\boldsymbol{\beta}} &= \mathcal{P}(\boldsymbol{y} - \widehat{\boldsymbol{m}}) \\ \widehat{\boldsymbol{m}} &= \mathcal{S}(\boldsymbol{y} - \mathcal{X}\widehat{\boldsymbol{\beta}}) \end{aligned}$$

as $\boldsymbol{y} - \widehat{\boldsymbol{m}}$ are the residuals from a nonparametric fit and $\boldsymbol{y} - \mathcal{X}\widehat{\boldsymbol{\beta}}$ the residuals from a linear regression. In this simple case an iteration is not necessary (Hastie and Tibshirani; 1990, Section 5.3) and the explicit solution is

$$\begin{aligned} \widehat{\boldsymbol{\beta}} &= \{\mathcal{X}^T(\mathcal{I} - \mathcal{S})\mathcal{X}\}^{-1}\mathcal{X}^T(\mathcal{I} - \mathcal{S})\boldsymbol{y}, \\ \widehat{\boldsymbol{m}} &= \mathcal{S}(\boldsymbol{y} - \mathcal{X}\widehat{\boldsymbol{\beta}}). \end{aligned}$$

These estimators differ from the Speckman estimators in only a subtle detail: the Speckman estimator for $\boldsymbol{\beta}$ shows $(\mathcal{I} - \mathcal{S})^\top(\mathcal{I} - \mathcal{S})$ instead of $(\mathcal{I} - \mathcal{S})$.

The extension of the PLM estimation method to the GPLM follows the same idea as in the Speckman case: apply the PLM approach using a weighted smoother matrix on the adjusted dependent variable (Hastie and Tibshirani; 1990, Section 6.7):

Backfitting estimation for the GPLM

- *calculate (in each step)*

$$\widetilde{\mathcal{X}} = (\mathcal{I} - \mathcal{S})\mathcal{X} \quad \text{and} \quad \widetilde{\boldsymbol{z}} = \widetilde{\mathcal{X}}\,\widehat{\boldsymbol{\beta}} - \mathcal{W}^{-1}\boldsymbol{v}$$

- *updating step for* $\boldsymbol{\beta}$

$$\widehat{\boldsymbol{\beta}}^{new} = \left(\mathcal{X}^T\mathcal{W}\widetilde{\mathcal{X}}\right)^{-1}\mathcal{X}^T\mathcal{W}\widetilde{\boldsymbol{z}},$$

- *updating step for* $\boldsymbol{m}$

$$\widehat{\boldsymbol{m}}^{new} = \mathcal{S}\left(\widetilde{\boldsymbol{z}} - \mathcal{X}\widehat{\boldsymbol{\beta}}^{new}\right)$$

As for the generalized Speckman estimation, we also find here a linear hat matrix $\mathcal{X}\widehat{\boldsymbol{\beta}} + \widehat{\boldsymbol{m}} = \mathcal{R}^B\boldsymbol{z}$, this time

$$\mathcal{R}^B = \widetilde{\mathcal{X}}\left(\mathcal{X}^T\mathcal{W}\widetilde{\mathcal{X}}\right)^{-1}\mathcal{X}^T\mathcal{W}(\mathcal{I} - \mathcal{S}) + \mathcal{S}, \tag{7}$$

(cf. Hastie and Tibshirani; 1990, Section 6.15).

2.3 Computational Aspects

The estimation approaches presented in Section 2 are implemented in the R package `gplm`. We refer to some general computational issues here while specific examples can be found in the following sections.

2.3.1 GPLM Iteration

► *Smoother matrix* $\mathcal{S}$
The choice of the matrix determines the way of smoothing that is used to estimate the non-parametric component function $m(\bullet)$. Even though we have used the Nadaraya–Watson type smoother for Section 2 any in R available linear smoother (that is able to incorporate weights) can be used instead. The `gplm` package covers two possible smoothers:

- `kgplm` using Nadaraya–Watson type smoothing:
  The Speckman and backfitting estimation methods require to compute terms of the form
  $$\sum_{i=1}^{n} \delta_i \mathcal{K}_{\mathbf{H}}(\boldsymbol{t}_i - \boldsymbol{t}), \tag{8}$$
  where the δ_i may be the y_i or the weight values w_{ii}. This computation has to be done at least for all sample observations $\boldsymbol{t}_j$ $(j = 1, \ldots, n)$ since updated values of $m(\bullet)$ at all observations are required in the updating step for $\boldsymbol{\beta}$. Thus $O(n^2)$ operations are necessary. In addition, for example for presentation purposes it can be useful to estimate the function $m(\bullet)$ on an additional grid. The `gplm` package provides the function `convol` to calculate (8). Different kernel functions are possible: product or spherical kernels based on triangle, uniform (rectangular), Epanechnikov, biweight (quartic), triweight and Gaussian (normal) univariate kernel function (see Subsection 2.3.2).
- `sgplm1` using spline smoothing:
  This routine uses the `smooth.spline` function in R, a linear spline smoother which allows the above mentioned incorporation of weights. In contrast to the kernel smoother, only $O(n)$ operations for the smoother plus $O(n \log(n))$ for sorting the data are necessary. On the other hand, this function does only allow for one-dimensional smoothing.

► *Initialization*
The presented iterative algorithms (for the GPLM) require first an initialization step. Different strategies to initialize the iterative algorithm are possible:

- Start with $\widetilde{\boldsymbol{\beta}}$, $\widetilde{m}(\bullet)$ from a parametric (GLM) fit.
- Alternatively, start with $\boldsymbol{\beta} = 0$ and $m(t_j) = G^{-1}(y_j)$ (for example with the adjustment $m_j = G^{-1}\{(y_j + 0.5)/2\}$ for binary responses).
- Backfitting procedures often use $\boldsymbol{\beta} = 0$ and $m(t_j) \equiv G^{-1}(\overline{y})$.

In practice, we do not find large differences between the approaches. `kgplm` and `sgplm1` use the third variant. It is however possible to set the initial values in a different manner by using the optional parameter `b.start` and `m.start`.

2.3.2 Kernel Functions

The R functions `kde` (kernel density estimation), `kreg` (kernel regression) and `kgplm` use multidimensional kernel functions to implement the smoothing procedure.

► *Product kernel functions*
The q-dimensional product kernels are obtained by multiplying one-dimensional kernel

functions K:

$$\mathcal{K}(\boldsymbol{u}) = K(u_1) \cdot \ldots \cdot K(u_q)\,.$$

- *Spherical kernel functions*
  The q-dimensional spherical (or radially symmetric) kernels are obtained by applying a one-dimensional kernel functions on the Euclidean norm $\|\boldsymbol{u}\|$ of the vector $\boldsymbol{u}$:

$$\mathcal{K}(\boldsymbol{u}) = c_{\mathcal{K},q}\, K(\|\boldsymbol{u}\|)\,.$$

  The constant $c_{\mathcal{K},q}$ is a scaling factor that ensures that the resulting function is a density, i.e. integrates to 1.

The following table shows the kernel functions implemented in the package `gplm`. The function $\mathrm{I}(\bullet)$ denotes the indicator function and the constant v_q denotes the volume of the q-dimensional unit sphere:

$$v_q = \frac{\pi^{q/2}}{\Gamma\,(1 + q/2)}\,.$$

	one-dimensional $K(u)$	spherical q-dimensional $\mathcal{K}(\boldsymbol{u})$
`triangle`	$(1 - \lvert u\rvert)\ \mathrm{I}(\lvert u\rvert \le 1)$	$\frac{q+1}{v_q}\,(1 - \|u\|)\ \mathrm{I}(\|u\| \le 1)$
`uniform`	$\frac{1}{2}\ \mathrm{I}(\lvert u\rvert \le 1)$	$\frac{1}{v_q}\ \mathrm{I}(\|u\| \le 1)$
`epanechnikov`	$\frac{3}{4}\,(1 - u^2)\ \mathrm{I}(\lvert u\rvert \le 1)$	$\frac{q+2}{2\,v_q}\,(1 - \|\boldsymbol{u}\|^2)\ \mathrm{I}(\|u\| \le 1)$
`biweight`	$\frac{15}{16}\,(1 - u^2)^2\ \mathrm{I}(\lvert u\rvert \le 1)$	$\frac{(q+2)(q+4)}{8\,v_q}\,(1 - \|\boldsymbol{u}\|^2)^2\ \mathrm{I}(\|u\| \le 1)$
`triweight`	$\frac{35}{32}\,(1 - u^2)^3\ \mathrm{I}(\lvert u\rvert \le 1)$	$\frac{(q+2)(q+4)(q+6)}{48\,v_q}\,(1 - \|\boldsymbol{u}\|^2)^3\ \mathrm{I}(\|u\| \le 1)$
`gaussian`	$\frac{1}{\sqrt{2\pi}}\ \exp(-u^2/2)$	$\frac{1}{(2\pi)^{q/2}}\ \exp(-\|\boldsymbol{u}\|^2/2)$

Note that in the case of a Gaussian kernel the q-dimensional product and spherical kernels coincide. For references on these kernel functions see for example Silverman (1986, Section 4.2), Fahrmeir and Hamerle (1984, Section 2.5) and Wand and Jones (1995, Section 4.5 and Appendix B).

3 Examples for Kernel Density Estimation and Kernel Regression

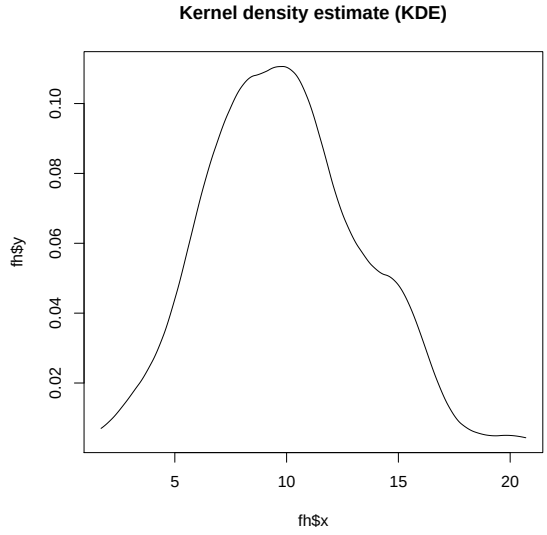
We start with some simple examples for kernel density and regression estimation. First we load the `airquality` data which are part of the R `base` package.

```
> data(airquality)
> attach(airquality)
```

For the variable `Wind` we estimate the kernel density estimator and graph it:

```
> library(gplm)
> fh <- kde(Wind)
> plot(fh, type="l", main="Kernel density estimate (KDE)")
```

This displays the following figure:



By default `kde` calculates the kernel density estimator using the quartic (biweight) kernel on an equidistant grid within the range of the data. The default bandwidth is calculated by Scott's rule of thumb (Scott; 1992, Section 6.3) as `kde` is indented to work with multivariate data (Härdle et al.; 2004, Section 3.6). The estimated bandwidth can be obtained by:

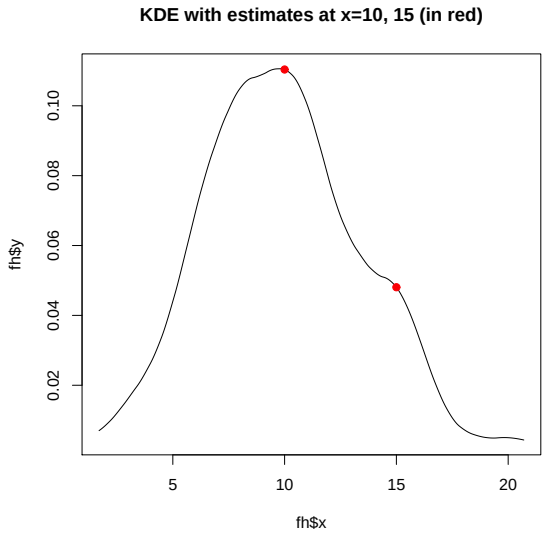
```
> fh$bandwidth
```

```
biweight
2.914528
```

The function `kde` is also able to calculate the density estimate on a grid of given vlaues. For example, to add the kernel density estimate at the value $x = 10$ and $x = 15$ to our figure we could type for example:

```
> fh.10 <- kde(Wind, grid=c(10,15))
> points(fh.10, col="red", pch=19)
> title("KDE with estimates at x=10, 15 (in red)")
```

This modifies our figure in the following way:

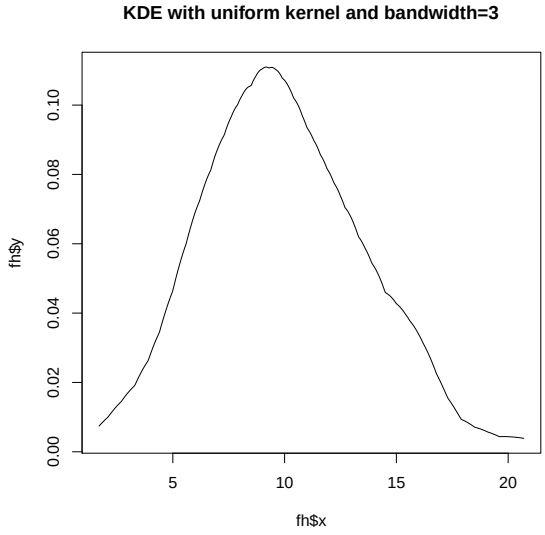


The bandwidth can be modified by setting the `bandwidth` parameter. Also the kernel function could be changed:

```
> fh <- kde(Wind, bandwidth=3, kernel="epanechnikov")
> fh$bandwidth
```

```
[1] 3
```

This provides us with:



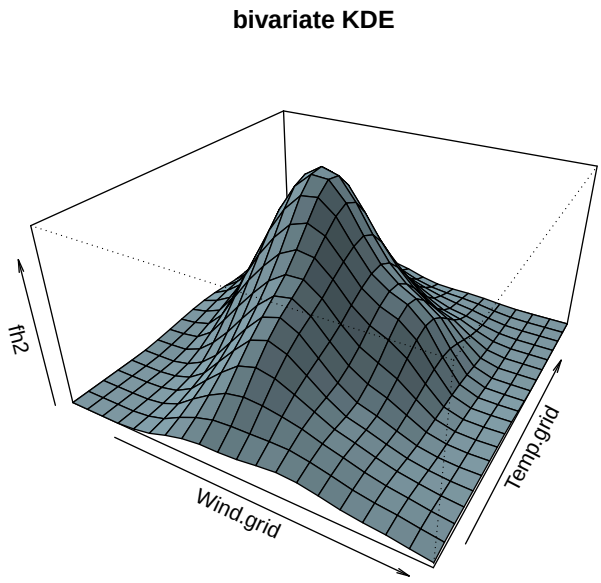
A two-dimensional kernel density estimate is computed in a similar way:

```
> fh.biv <- kde(cbind(Wind,Temp))
```

To obtain a plot of the resulting density surface requires a little more effort:

```
> Wind.grid <- unique(fh.biv$x[,1])  ## extract grids
> Temp.grid <- unique(fh.biv$x[,2])
> o <- order(fh.biv$x[,2],fh.biv$x[,1])  ## order by 2nd column
> fh2 <- matrix(fh.biv$y[o],length(Wind.grid),length(Temp.grid))
> persp(Wind.grid,Temp.grid,fh2,main="bivariate KDE",
+       theta=30,phi=30,expand=0.5,col="lightblue",shade=0.5)
```

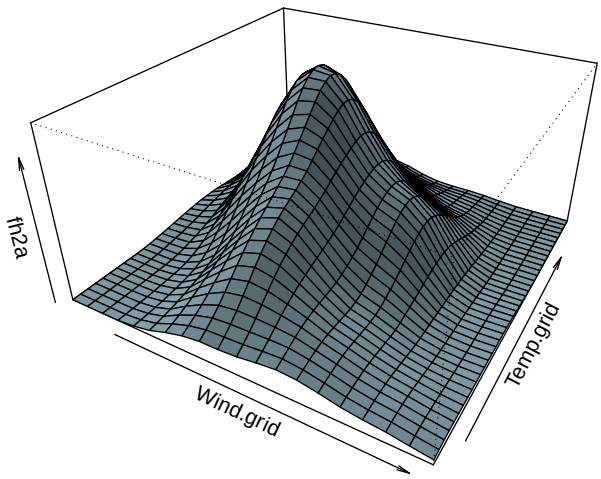
The resulting surface is shown as:



In order to use different grid sizes we could apply for example:

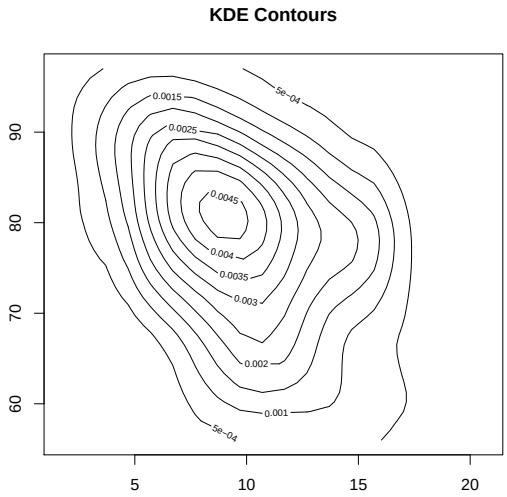
```
> Wind.grid <- seq(min(Wind),max(Wind),length=20)  ## define grid
> Temp.grid <- seq(min(Temp),max(Temp),length=40)
> fh.biv <- kde(cbind(Wind,Temp), grid=create.grid(list(Wind.grid,Temp.grid)))
> o <- order(fh.biv$x[,2],fh.biv$x[,1])  ## order by 2nd column
> fh2a <- matrix(fh.biv$y[o],length(Wind.grid),length(Temp.grid))
> persp(Wind.grid,Temp.grid,fh2a,main="bivariate KDE",
+       theta=30,phi=30,expand=0.5,col="lightblue",shade=0.5)
```

The now resulting surface shows a finer grid in the `Temp` dimension:



An alternative graphical display is a contour plot computed from:

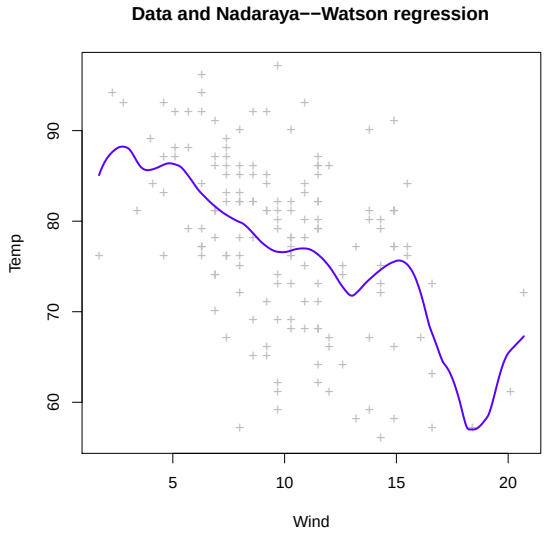
```
> contour(Wind.grid,Temp.grid,fh2a, main="KDE Contours")
```



The function `kreg` implements the (multivariate) Nadaraya–Watson estimator for estimating the regression function $m(\boldsymbol{x}) = E(Y|\boldsymbol{x})$. As a univariate regression example we use the regression:

```
> mh <- kreg(Wind, Temp)
> plot(Wind,Temp, col="grey", pch="+")
> lines(mh, col="blue", lwd=2)
> title("Data and Nadaraya--Watson regression")
```

The default bandwidth is calculated by Scott's rule of thumb. All other options are practically identical to `kde`. The considered code displays the figure:

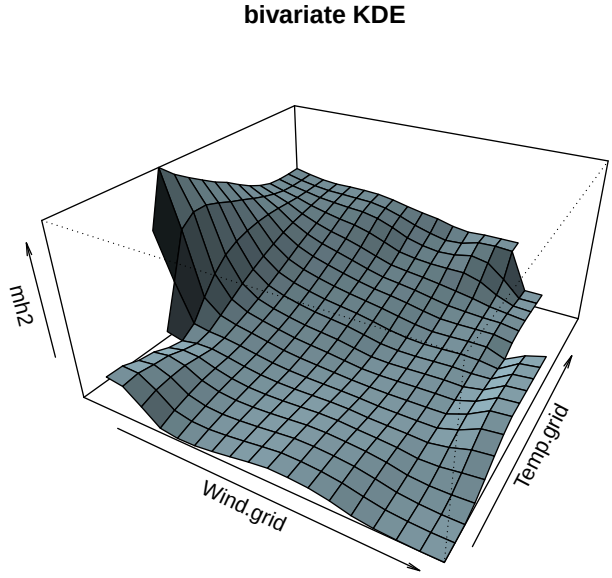


In the same way as before we can estimate bivariate (or higher dimensional) regression functions:

```
> airquality2 <- airquality[!is.na(Ozone),] ## delete NA's
> detach(airquality)  ## detach previous data
> attach(airquality2)
> bandwidth <- bandwidth.scott(cbind(Wind,Temp))
> mh.biv <- kreg(cbind(Wind,Temp),Ozone, bandwidth=bandwidth)
> Wind.grid <- unique(mh.biv$x[,1])  ## extract grids
```

```
> Temp.grid <- unique(mh.biv$x[,2])
> o <- order(mh.biv$x[,2],mh.biv$x[,1])  ## order by 2nd column
> mh2 <- matrix(mh.biv$y[o],length(Wind.grid),length(Temp.grid))
> persp(Wind.grid,Temp.grid,mh2,main="bivariate KDE",
+       theta=30,phi=30,expand=0.5,col="lightblue",shade=0.5)
```

The estimated regression surface is:



4 Examples for a Partial Linear Model (PLM)

To illustrate the PLM we use the data from the Current Population Survey 1985 in package `AER`:

```
> library(AER)
> data(CPS1985)
> str(CPS1985)  ## show data structure
```

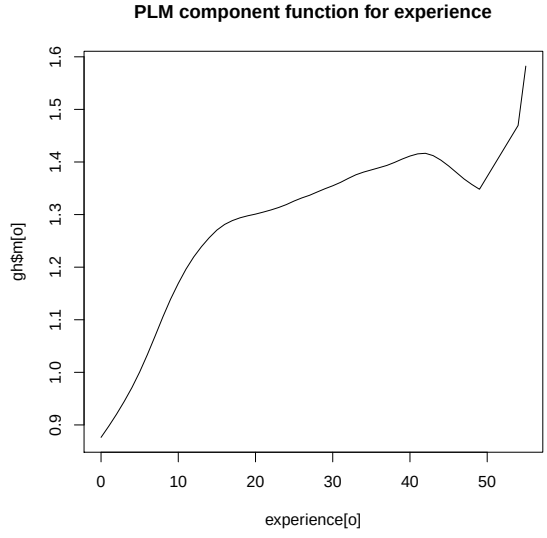
```
'data.frame':        534 obs. of  11 variables:
 $ wage      : num  5.1 4.95 6.67 4 7.5 ...
 $ education : num  8 9 12 12 12 13 10 12 16 12 ...
 $ experience: num  21 42 1 4 17 9 27 9 11 9 ...
 $ age       : num  35 57 19 22 35 28 43 27 33 27 ...
 $ ethnicity : Factor w/ 3 levels "cauc","hispanic",..: 2 1 1 1 1 1 1 1 1 1 ...
 $ region    : Factor w/ 2 levels "south","other": 2 2 2 2 2 2 1 2 2 2 ...
 $ gender    : Factor w/ 2 levels "male","female": 2 2 1 1 1 1 1 1 1 1 ...
 $ occupation: Factor w/ 6 levels "worker","technical",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ sector    : Factor w/ 3 levels "manufacturing",..: 1 1 1 3 3 3 3 3 1 3 ...
 $ union     : Factor w/ 2 levels "no","yes": 1 1 1 1 1 2 1 1 1 1 ...
 $ married   : Factor w/ 2 levels "no","yes": 2 2 1 1 2 1 1 1 2 1 ...
```

```
> attach(CPS1985)
```

It is often assumed that the `log(wage)` of a person depends linearly on `education` and nonlinearly on `experience`. We can check that by estimating a PLM using `gplm`:

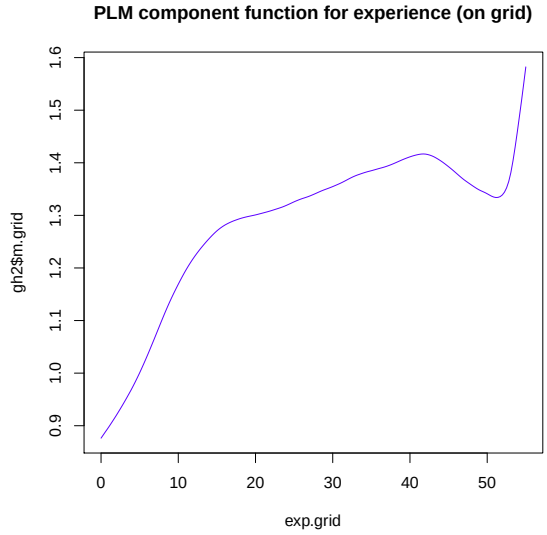
```
> library(gplm)
> bandwidth <- bandwidth.scott(experience)
> gh <- kgplm(x=cbind(gender,education),t=experience,y=log(wage),h=bandwidth)
> o <- order(experience)  ## sort curve estimate by experience
> plot(experience[o], gh$m[o], type="l")
> title("PLM component function for experience")
```

Indeed we find an inverse U-shaped function showing some artefacts (due to sparse data) at the right boundary:



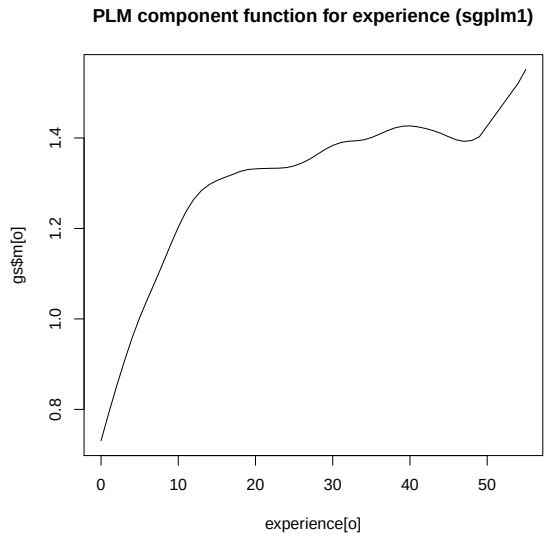
As before we can compute the estimated function on a finer grid which also generates a more smooth picture:

```
> exp.grid <- seq(min(experience),max(experience),length=200)
> gh2 <- kgplm(x=cbind(gender,education),t=experience,y=log(wage),
+              h=bandwidth,grid=exp.grid)
> plot(exp.grid, gh2$m.grid, type="l", col="blue")
> title("PLM component function for experience (on grid)")
```



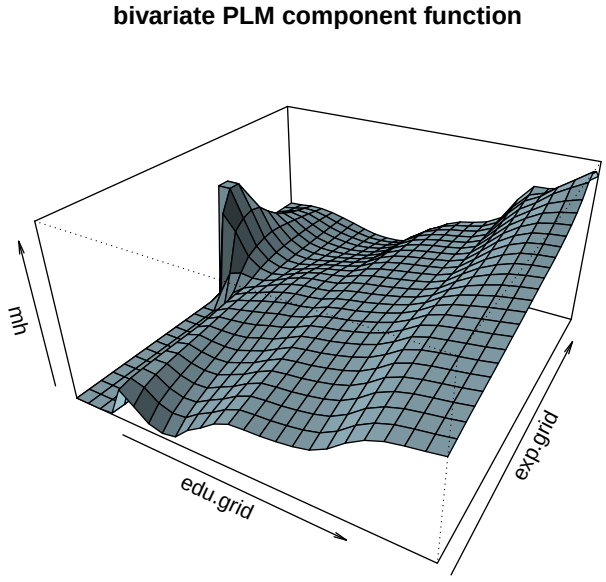
The R function name `kgplm` indicates that this function is based on kernel smoothing. An alternative to `kgplm` is the function `sgplm1` based on splines. This function however does only estimate univariate PLM component functions. An estimate similar to that by `kgplm` is found by:

```
> gs <- sgplm1(x=cbind(gender,education),t=experience,y=log(wage),df=8)
> o <- order(experience)  ## sort curve estimate by experience
> plot(experience[o], gs$m[o], type="l")
> title("PLM component function for experience (sgplm1)")
```



Using `kgplm` we can also estimate a bivariate PLM component function for education and experience. Technically this follows the examples for a bivariate kernel density or regression. The function `kgplm` does however not automatically define a grid. We therefore specify the grid first and then display the function using `persp` as already shown:

```
> bandwidth <- 1.5*bandwidth.scott(cbind(education,experience))
> edu.grid <- seq(min(education),max(education),length=25)
> exp.grid <- seq(min(experience),max(experience),length=25)
> grid  <- create.grid(list(edu.grid,exp.grid))
> gh <- kgplm(x=(gender=="female"),t=cbind(education,experience),y=log(wage),
+             h=bandwidth,grid=grid)
> o <- order(grid[,2],grid[,1])
> mh <- matrix(gh$m.grid[o],length(edu.grid),length(exp.grid))
> persp(edu.grid,exp.grid,mh,
+       theta=30,phi=30,expand=0.5,col="lightblue",shade=0.5)
> title("bivariate PLM component function")
```



5 Examples for a Generalized Partial Linear Model (GPLM)

For illustrating the GPLM we use another dataset from the `AER` package, the data on extramarital affairs. In order to consider a binary response we transform the number of affairs to a binary indicator variable `y`:

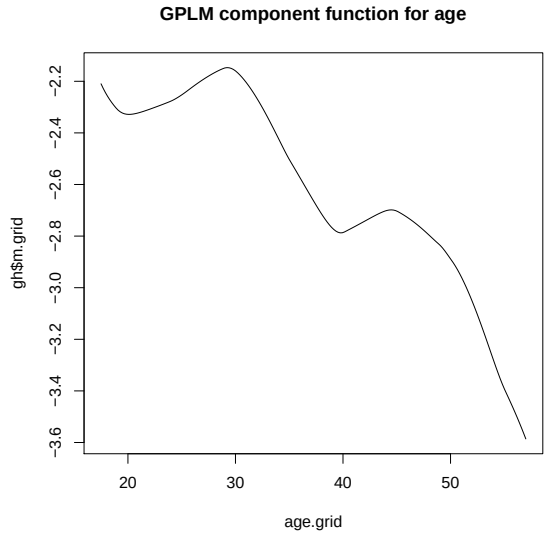
```
> library(AER)
> data(Affairs)
> str(Affairs)  ## show data structure
```

```
'data.frame':        601 obs. of  9 variables:
 $ affairs      : num  0 0 0 0 0 0 0 0 0 0 ...
 $ gender       : Factor w/ 2 levels "female","male": 2 1 1 2 2 1 1 2 1 2 ...
 $ age          : num  37 27 32 57 22 32 22 57 32 22 ...
 $ yearsmarried : num  10 4 15 15 0.75 1.5 0.75 15 15 1.5 ...
 $ children     : Factor w/ 2 levels "no","yes": 1 1 2 2 1 1 1 2 2 1 ...
 $ religiousness: int  3 4 1 5 2 2 2 2 4 4 ...
 $ education    : num  18 14 12 18 17 17 12 14 16 14 ...
 $ occupation   : int  7 6 1 6 6 5 1 4 1 4 ...
 $ rating       : int  4 4 4 5 3 5 3 4 2 5 ...
```

```
> attach(Affairs)
> y <- (affairs > 0)
```

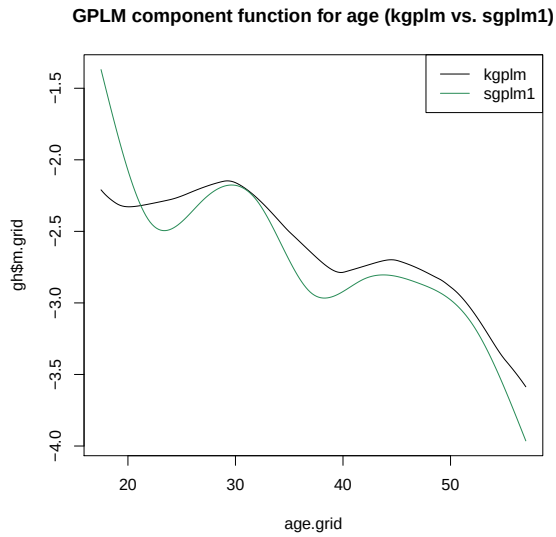
The GPLM estimation is again very similar to the PLM estimation. Basically, the only main difference is the specification of the distribution family for the response and the respective link function. We are now estimate a logit-type GPLM for the response `y`:

```
> library(gplm)
> bandwidth <- 1.5*bandwidth.scott(age)
> age.grid <- seq(min(age),max(age),length=200)
> gh <- kgplm(x=cbind(gender,education,yearsmarried),t=age,y=y,h=bandwidth,
+             grid=age.grid,family="bernoulli",link="logit")
> plot(age.grid, gh$m.grid, type="l")
> title("GPLM component function for age")
```

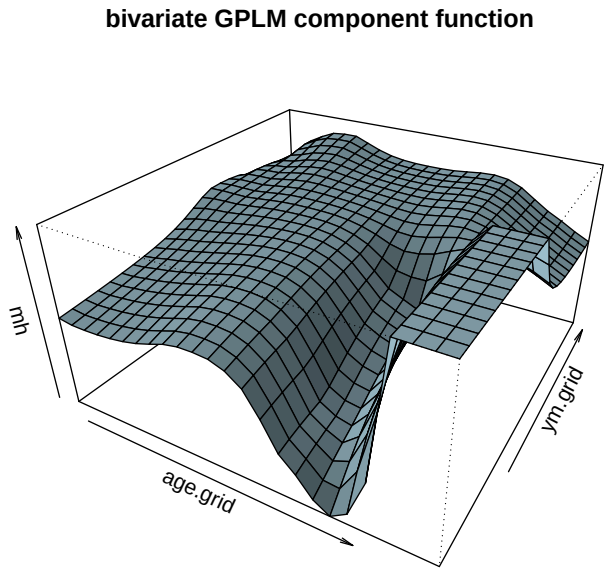


Alternatively we can now use `sgplm1` again and compare both estimates:

```
> gs <- sgplm1(x=cbind(gender,education,yearsmarried),t=age,y=y,df=7,
+              grid=age.grid,family="bernoulli",link="logit")
> ylim <- range(gh$m.grid, gs$m.grid)
> plot(age.grid, gh$m.grid, type="l", ylim=ylim)
> lines(age.grid, gs$m.grid, col="seagreen")
> title("GPLM component function for age (kgplm vs. sgplm1)")
> legend("topright",c("kgplm","sgplm1"),lwd=1,col=c("black","seagreen"))
```



The previous figures indicated that extramarital affairs are less likely for higher age. This changes if we estimate a bivariate GPLM component function. The following figure shows that the propensity varies with `age` and `yearsmarried`:



The following code was used to generate this figure:

```
> bandwidth <- 1.5*bandwidth.scott(cbind(age,yearsmarried))
> age.grid <- seq(min(age),max(age),length=25)
> ym.grid <- seq(min(yearsmarried),max(yearsmarried),length=25)
> grid  <- create.grid(list(age.grid,ym.grid))
> gh <- kgplm(x=(gender=="female"),t=cbind(age,yearsmarried),y=y,
+             h=bandwidth,grid=grid,family="bernoulli",link="logit")
> o <- order(grid[,2],grid[,1])
> mh <- matrix(gh$m.grid[o],length(age.grid),length(ym.grid))
> persp(age.grid,ym.grid,mh,
+       theta=30,phi=30,expand=0.5,col="lightblue",shade=0.5)
> title("bivariate GPLM component function")
```

References

Buja, A., Hastie, T. J. and Tibshirani, R. J. (1989). Linear smoothers and additive models (with discussion), *Annals of Statistics* **17**: 453–555.

Fahrmeir, L. and Hamerle, A. (1984). *Multivariate Statistische Verfahren*, De Gruyter, Berlin.

Härdle, W., Müller, M., Sperlich, S. and Werwatz, A. (2004). *Nonparametric and Semiparametric Modeling: An Introduction*, Springer, New York.

Hastie, T. J. and Tibshirani, R. J. (1990). *Generalized Additive Models*, Vol. 43 of *Monographs on Statistics and Applied Probability*, Chapman and Hall, London.

McCullagh, P. and Nelder, J. A. (1989). *Generalized Linear Models*, Vol. 37 of *Monographs on Statistics and Applied Probability*, 2 edn, Chapman and Hall, London.

Müller, M. (2001). Estimation and testing in generalized partial linear models — a comparative study, *Statistics and Computing* **11**: 299–309.

Müller, M. (2012). Generalized linear models, *in* J. Gentle, W. Härdle and Y. Mori (eds), *Handbook of Computational Statistics*, 2 edn, Springer, Heidelberg.

Robinson, P. M. (1988). Root n–consistent semiparametric regression, **56**: 931–954.

Scott, D. W. (1992). *Multivariate Density Estimation: Theory, Practice, and Visualization*, John Wiley & Sons, New York, Chichester.

Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis*, Vol. 26 of *Monographs on Statistics and Applied Probability*, Chapman and Hall, London.

Speckman, P. E. (1988). Regression analysis for partially linear models, *Journal of the Royal Statistical Society, Series B* **50**: 413–436.

Wand, M. P. and Jones, M. C. (1995). *Kernel Smoothing*, Vol. 60 of *Monographs on Statistics and Applied Probability*, Chapman and Hall, London.