

Müller, Marlene

Working Paper

Generalized partial linear models

SFB 373 Discussion Paper, No. 2000,52

Provided in Cooperation with:

Collaborative Research Center 373: Quantification and Simulation of Economic Processes,
Humboldt University Berlin

Suggested Citation: Müller, Marlene (2000) : Generalized partial linear models, SFB 373 Discussion Paper, No. 2000,52, Humboldt University of Berlin, Interdisciplinary Research Project 373: Quantification and Simulation of Economic Processes, Berlin,
<https://nbn-resolving.de/urn:nbn:de:kobv:11-10047788>

This Version is available at:

<https://hdl.handle.net/10419/62202>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.

Generalized Partial Linear Models

Marlene Müller

A **generalized linear model** (GLM) is a regression model of the form

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

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

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

In the following we describe how to use the XploRe `gplm` quantlib for estimating generalized partial linear models. The `gplm` quantlib is highly related to the `glm` quantlib for GLM in XploRe. Names of routines and the functionality in both quantlibs correspond to each other. It is recommended to start reading with the GLM tutorial (Härdle, Klinke, and Müller 2000, Chapter 7). Parts of the features which are also available in GLM are not explained in detail here.

1 Estimating GPLMs

As mentioned above, a GPLM has the form

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

where $E(Y|X, T)$ denotes the expected value of the dependent variable Y given X , T which are vectors of explanatory variables. The index $X^T\beta + m(T)$ is linked to the dependent variable Y via a known function $G(\bullet)$ which is called the **link** function in analogy to generalized linear models (GLM). The parameter vector β and the function $m(\bullet)$ need to be estimated. Typically, generalized partial linear models are considered for Y from an exponential family. We therefore assume for the variance $Var(Y|X, T) = \sigma^2 V[G\{X^T\beta + m(T)\}]$, i.e. a dependence on the index $X^T\beta + m(T)$ and on a dispersion parameter σ^2 .

1.1 Models

It is easy to see that GPLM covers a range of semiparametric models, as for example:

- **Partial linear regression**

The model $Y = X^T\beta + m(T) + \varepsilon$ with $\varepsilon \sim N(0, \sigma^2)$ implies $E(Y|X, T) = G\{X^T\beta + m(T)\}$ and $Var(Y|X, T) = \sigma^2$. This gives a GPLM with **identity** link function $G(\bullet) = \bullet$ and variance function $V(\bullet) = 1$.

- **Generalized additive model (GAM) with linear and nonparametric component**

This is commonly written as $E(Y|X, T) = G\{c + X^T\beta + f(T)\}$ where $Ef(T) = 0$ is assumed. By defining $m(t) = c + f(t)$ we arrive at the above GPLM.

1.2 Semiparametric Likelihood

The estimation methods for the GPLM are based on the idea that an estimate $\hat{\beta}$ can be found for known $m(\bullet)$, and an estimate $\hat{m}(\bullet)$ can be found for known β . The `gplm` quantlib implements **profile likelihood** estimation and **backfitting**. Details on the estimation procedure can be found in Hastie and Tibshirani (1990), Severini and Staniswalis (1994), Härdle, Mammen and Müller (1998), Müller (1997).

The default numerical algorithm for likelihood maximization is the Newton-Raphson iteration. Optionally, a Fisher scoring can be chosen.

Profile Likelihood Denote by $L(\mu, y)$ the individual log-likelihood or (if the distribution of Y does not belong to an exponential family) quasi-likelihood function

$$L(\mu, y) = \int_{\mu}^y \frac{(s - y)}{V(s)} ds.$$

The **profile likelihood** method considered in Severini and Wong (1992) and Severini and Staniswalis (1994) is based on the fact, that the conditional distribution of Y given X and T is parametric. The essential method for estimation is to fix the parameter β and to estimate the least favorable nonparametric

function in dependence of this fixed β . The resulting estimate for $m_\beta(\bullet)$ is then used to construct the profile likelihood for β .

Suppose, we have observations $\{y_i, x_i, t_i\}$, $i = 1, \dots, n$. Denote the individual log- or quasi-likelihood in y_i by

$$\ell_i(\eta) = L\{G(\eta), y_i\}.$$

In the following, ℓ'_i and ℓ''_i denote the derivatives of $\ell_i(\eta)$ with respect to η . Abbreviate now $m_j = m_\beta(t_j)$ and define S^P the smoother matrix with elements

$$S^P_{ij} = \frac{\ell''_i(x_i^T \beta + m_j) K_H(t_i - t_j)}{\sum_{i=1}^n \ell''_i(x_i^T \beta + m_j) K_H(t_i - t_j)} \quad (1)$$

and let X be the design matrix with rows x_i^T . Denote further by I the identity matrix, by v the vector and by W the diagonal matrix containing the first (ℓ'_i) and second (ℓ''_i) derivatives of $\ell_i(x_i^T \beta + m_i)$, respectively.

The Newton-Raphson estimation algorithm (see Severini and Staniswalis 1994) is then as follows.

Profile Likelihood Algorithm	
<ul style="list-style-type: none"> • <i>updating step for β</i> 	$\beta^{new} = (\tilde{X}^T W \tilde{X})^{-1} \tilde{X}^T W \tilde{z}$
<p style="margin-left: 20px;">with</p>	$\begin{aligned} \tilde{X} &= (I - S^P)X, \\ \tilde{z} &= \tilde{X}\beta - W^{-1}v. \end{aligned}$
<ul style="list-style-type: none"> • <i>updating step for m_j</i> 	$m_j^{new} = m_j - \frac{\sum_{i=1}^n \ell'_i(x_i^T \beta + m_j) K_H(t_i - t_j)}{\sum_{i=1}^n \ell''_i(x_i^T \beta + m_j) K_H(t_i - t_j)}.$

The variable \tilde{z} is a sort of adjusted dependent variable. From the formula for

β^{new} it becomes clear, that the parametric part of the model is updated by a parametric method (with a nonparametrically modified design matrix X).

Alternatively, the functions ℓ_i'' can be replaced by their expectations (w.r.t. to y_i) to obtain a Fisher scoring type procedure.

Generalized Speckman Estimator The profile likelihood estimator is particularly easy to derive in case of a model with identity link and normally distributed y_i . Here, $\ell_i' = y_i - x_i^T \beta - m_j$ and $\ell_i'' \equiv -1$. The latter yields the smoother matrix S with elements

$$S_{ij} = \frac{K_H(t_i - t_j)}{\sum_{i=1}^n K_H(t_i - t_j)}. \quad (2)$$

Moreover, the update for m_j simplifies to

$$m^{new} = S(y - X\beta)$$

using the vector notation $y = (y_1, \dots, y_n)^T$, $m^{new} = (m_1^{new}, \dots, m_n^{new})^T$. The parametric component is determined by

$$\beta^{new} = (\tilde{X}^T \tilde{X})^{-1} \tilde{X}^T \tilde{y}$$

with $\tilde{X} = (I - S)X$ and $\tilde{y} = (I - S)y$. These estimators for the partial linear model were proposed by Speckman (1988).

Recall that each iteration step of a GLM is a weighted least squares regression on an adjusted dependent variable (McCullagh and Nelder 1989). Hence, in the partial linear model the weighted least squares regression could be replaced by an partial linear fit on the adjusted dependent variable

$$z = X\beta + m - W^{-1}v. \quad (3)$$

Again, denote v a vector and W a diagonal matrix containing the first (ℓ_i') and second (ℓ_i'') derivatives of $\ell_i(x_i^T \beta + m_i)$, respectively. Then, the Newton-Raphson type Speckman estimator (see Müller 1997) for the GPLM can be written as:

Generalized Speckman Algorithm

- *updating step for β*

$$\beta^{new} = (\tilde{X}^T W \tilde{X})^{-1} \tilde{X}^T W \tilde{z},$$

- *updating step for m*

$$m^{new} = S(z - X\beta),$$

using the notations

$$\begin{aligned} \tilde{X} &= (I - S)X, \\ \tilde{z} &= (I - S)z = \tilde{X}\beta - W^{-1}v. \end{aligned}$$

The basic simplification of this approach consists in using the smoothing matrix S with elements

$$S_{ij} = \frac{\ell_i''(x_i^T \beta + m_i) K_H(t_i - t_j)}{\sum_{i=1}^n \ell_i''(x_i^T \beta + m_i) K_H(t_i - t_j)} \quad (4)$$

instead of the matrix S^P from (1). As before, a Fisher scoring type procedure is obtained by replacing ℓ_i'' by their expectations.

Backfitting The backfitting method was suggested as an iterative algorithm to fit an additive model (Hastie and Tibshirani 1990). The key idea is to regress the additive components separately on partial residuals. The ordinary partial linear model (with identity link function)

$$E(Y|X, T) = X^T \beta + m(T)$$

is a special case, consisting of only two additive functions. Denote P the projection matrix $P = X(X^T X)^{-1} X^T$ and S a smoother matrix. Abbreviate $m = (m_1, \dots, m_n)^T = (m(t_1), \dots, m(t_n))^T$. Then backfitting means to solve

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

For a GPLM, backfitting means now to perform an additive fit on the adjusted dependent variable z which was defined in (3), see Hastie and Tibshirani (1990). We use again the kernel smoother matrix S from (4).

Backfitting Algorithm	
• <i>updating step for β</i>	$\beta^{new} = (X^T W \tilde{X})^{-1} X^T W \tilde{z},$
• <i>updating step for m</i>	$m^{new} = S(z - X\beta),$
using the notations	
	$\tilde{X} = (I - S)X,$
	$\tilde{z} = (I - S)z = \tilde{X}\beta - W^{-1}v.$

As for profile likelihood and Speckman estimation, we obtain a Newton-Raphson or Fisher scoring type algorithm by using ℓ'_i or $E(\ell'_i)$, respectively.

2 Data Preparation

2.1 General

All estimation quantlets in the `gplm` quantlib have as input parameters:

- x A $n \times p$ matrix containing observations of explanatory variables for the linear part,
- t A $n \times q$ matrix containing observations of explanatory variables for the nonparametric part,
- y A $n \times 1$ vector containing the observed responses.

There should be no vector of 1 concatenated to the matrix \mathbf{x} . A constant is contained automatically in the nonparametric estimate for $m(\bullet)$. Neither the matrices \mathbf{x} , \mathbf{t} nor the vector \mathbf{y} should contain missing values (NaN) or infinite values (Inf, -Inf).

2.2 Credit Scoring Example

In the following, we will use credit scoring data to illustrate the GPLM estimation. For details on the file `kredit` see Fahrmeir and Tutz (1994) or Fahrmeir and Hamerle (1984). We use a subsample on loans for cars and furniture, which has a sample size of $n = 564$ out of 1000.

		Yes	No	(in %)	
Y	credit worthy	75.7	24.3		
X_1	previous credits o.k.	36.2	63.8		
X_2	employed	77.0	23.0		
		Min	Max	Mean	S.D.
X_3	duration (months)	4	72	20.90	11.41
T_1	amount (DM)	338	15653	3200.00	2467.30
T_2	age (years)	19	75	34.46	10.96

Table 1: Descriptive statistics for credit data.

Descriptive statistics for this subsample and a selection of covariates can be found in Table 1. The covariate **previous credit o.k.** indicates that previous loans were repaid without problems. The variable **employed** means that the person taking the loan has been employed by the same employer for at least one year.

The following XploRe code creates matrices \mathbf{x} , \mathbf{t} and \mathbf{y}

```

library("stats")
file=read("kredit")
file=paf(file, (file[,5]>=1)&&(file[,5]<=3))
                                ; purpose=car/furniture
y=file[,1]
x=(file[,4]>2)                    ; previous loans o.k.
x=x~(file[,8]>2)                  ; employed (>=1 year)


```



```

x=x~(file[,3])                ; duration of loan
t=(file[,6])                  ; amount of loan
t=t~(file[,14])               ; age of client
xvars="previous"|"employed"|"duration"
tvars="amount"|"age"
summarize(y~x~t,"y"|xvars|tvars)

```

 gplm01.xpl

and produces the summary statistics:

[2,]	Minimum	Maximum	Mean	Median	Std.Error
[3,]					
[4,] y	0	1	0.75709	1	0.42922
[5,] previous	0	1	0.3617	0	0.48092
[6,] employed	0	1	0.7695	1	0.42152
[7,] duration	4	72	20.902	18	11.407
[8,] amount	338	15653	3200	2406	2467.3
[9,] age	19	75	34.463	32	10.964

Note that in the following statistical analysis we took logarithms of **amount** and **age** and transformed these values linearly to the interval $[0, 1]$.

3 Computing GPLM Estimates

Currently six types of distributions are supported by the `gplm` quantlib: Binomial, Normal (Gaussian), Poisson, Gamma (includes Exponential), Inverse Gaussian and Negative Binomial (includes Geometric). Table 2 summarizes the models which are available.

The quantlet in the `gplm` quantlib which is mainly responsible for GPLM estimation is `gplmest`.

3.1 Estimation

```

g = gplmest (code, x, t, y, h {, opt})
  estimates a GPLM

```

Distribution	Model Code	Link Function
Gaussian	"noid"	identity link (canonical)
	"nopow"	power link
Binomial	"bilo"	Logistic link (Logit, canonical)
	"bipro"	Gaussian link (Probit)
	"bicll"	complementary log-log link
Poisson	"polog"	logarithmic link (canonical)
	"popow"	power link
Gamma	"gac1"	reciprocal link (canonical)
	"gapow"	power link
Inv. Gaussian	"igc1"	squared reciprocal link (canonical)
	"igpow"	power link
Neg. Binomial	"nbc1"	canonical link
	"igpow"	power link

Table 2: Supported models.

The quantlet `gplmest` provides a convenient way to estimate a GPLM. The standard call is quite simple, for example

```
g=gplmest("bipro",x,t,y,h)
```

estimates a probit model (binomial with Gaussian cdf link). For `gplmest` the short code of the model (here "bipro") needs to be given, this is the same short code as for the `glm` quantlib. Additionally to the data, a bandwidth parameter `h` needs to be given (a vector corresponding to the dimension of `t` or just a scalar).

The result of the estimation is assigned to the variable `g` which is a list containing the following output:

```
g.b
  the estimated parameter vector

g.bv
  the estimated covariance of g.b


g.m
  the estimated nonparametric function
```

`g.stat`
contains the statistics (see Section 5).

Recalling our credit scoring example from Subsection 2.2, the estimation—using a logit link—would be done as follows:

```
t=log(t) ; logs of amount and age
trange=max(t)-min(t)
t=(t-min(t))./trange ; transformation to [0,1]
library("gplm")

h=0.4
g=gplmest("bilo",x,t,y,h)
g.b
```


 `gplm02.xpl`

Now we can inspect the estimated coefficients in `g.b`

```
Contents of b
[1,] 0.96516
[2,] 0.74628
[3,] -0.049835
```

A graphical output can be created by calling


```
gplmout("bilo",x,t,y,h,g.b,g.bv,g.m,g.stat)
```

 `gplm02.xpl`

for the current example (cf. Figure 1). For more features of `gplmout` see Subsections 4.7 and 5.2.

Optional parameters must be given to `gplmest` in a list of optional parameters. A detailed description of what is possible can be found in Section 4, which deals with the quantlet `gplmopt`. Set:

```
opt=gplmopt("meth",1,"shf",1)
opt=gplmopt("xvars",xvars,opt)
opt=gplmopt("tg",grid(0|0,0.05|0.05,21|21),opt)
```

 `gplm03.xpl`

This will create a list `opt` of optional parameters. In the first call, `opt` is

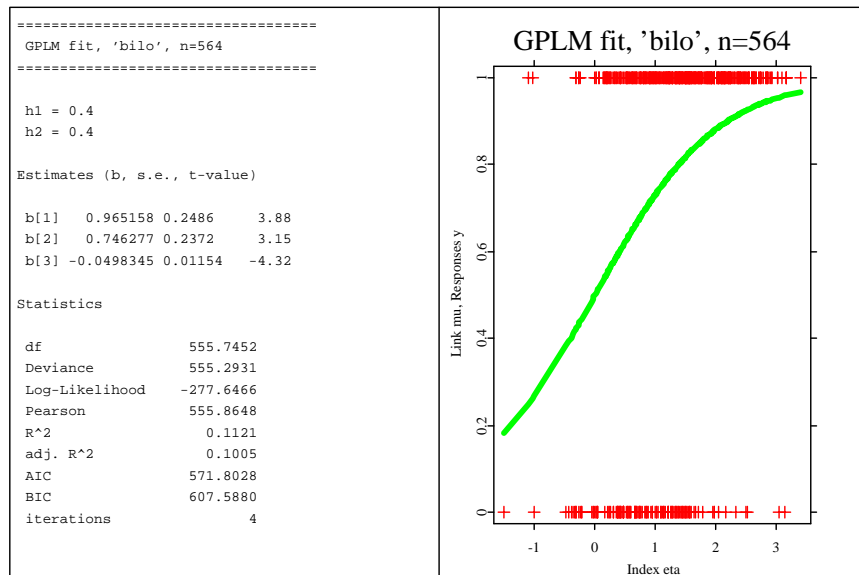



Figure 1: GPLM output display.

created with the first component `meth` (estimation method) containing the value 1 (profile likelihood algorithm) and the second component `shf` (show iteration) set to 1 ("true"). In the second call, the variable names for the linear part of the model are appended to `opt`. Finally, a grid component `tg` (for the estimation of the nonparametric part) is defined.

We repeat the estimation with these settings:

```
g=gplmest("bilo",x,t,y,h,opt)
```

 `gplm03.xpl`

This instruction now computes using profile likelihood algorithm (in contrast to the default Speckman algorithm used in example [gplm02.xpl](#)), shows the iteration in the output window and estimates the function $m(\bullet)$ on the grid `tg`. The output `g` contains one more element now:

```
g.mg
    the estimated nonparametric function on the grid
```

Since the nonparametric function $m(\bullet)$ is estimated on two-dimensional data, we can display a surface plot using the estimated function on the grid:

```
library("plot")
mg=setmask(sort(tg~g.mg),"surface")
```


 [gplm03.xpl](#)

Figure 2 shows this surface together with a scatterplot of **amount** and **age**. The scatterplot shows that the big peak of $\hat{m}(\bullet)$ is caused by only a few observations. For the complete XploRe code of this example check the file [gplm03.xpl](#).

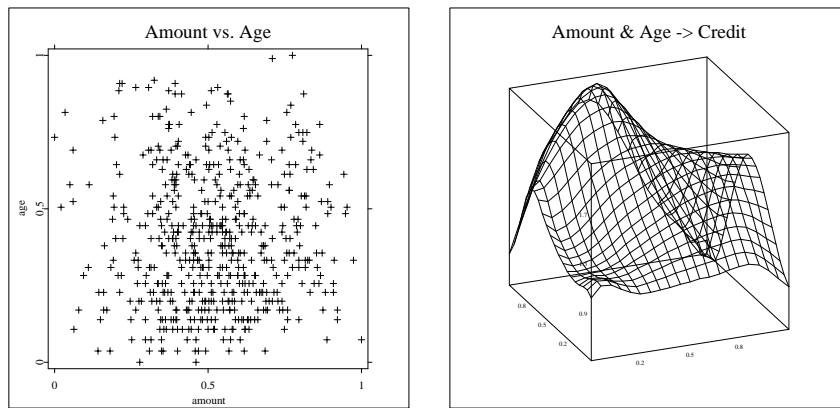


Figure 2: Scatterplot for amount and age (left). Estimate \hat{m} (right).

The estimated coefficients are slightly different here, since we used the profile likelihood instead of the Speckman algorithm in this case. Figure 3 shows the output window for the second estimation.

3.2 Estimation in Expert Mode

```
g = gplmcore(code, x, t, y, h, wx, wt, wc, b0, m0,
             off, ctrl{, upb, tg, m0g})
      estimates a GPLM in expert mode
```

The `gplmcore` quantlet is the most inner “kernel” of the GPLM estimation. It does not provide optional parameters in the usual form of an option list as described in Section 4. Also, no check is done for erroneous input. Hence, this routine can be considered to use in expert mode. It speeds up computations and might be useful in simulations, pilot estimation for other procedures or Monte Carlo methods.

The following lines show how `gplmcore` could be used in our running example. Note that all data needs to be sorted by the first column of `t`.

```
n=rows(x)
p=cols(x)
q=cols(t)

tmp=sort(t~y~x)      ; sort data by first column of t
t=tmp[, (1:q)]
y=tmp[, (q+1)]
x=tmp[, (q+2):cols(tmp)]

shf  = 1      ; show iteration (1="true")
miter = 10    ; maximal number of iterations
cnv  = 0.0001 ; convergence criterion
fscor = 0     ; Fisher scoring (1="true")
pow  = 0     ; power for power link (if useful)
nbk  = 1     ; k for neg. binomial (if useful)
meth = 0     ; algorithm ( -1 = backfitting,
                        ;           0 = Speckman
                        ;           1 = profile likelihood )
ctrl=shf|miter|cnv|fscor|pow|nbk|meth

wx  = 1      ; prior or frequency weights
wt  = 1      ; trimming weights for estimation of b
wc  = 1      ; weights for the convergence criterion
```


```

off = 0 ; offset

l=glmcore("bilo",x~t~matrix(n),y,wx,off,ctrl[1:6])
b0=l.b[1:p]
m0=l.b[p+q+1]+t*l.b[(p+1):(p+q)]

h=0.4|0.4
g=gplmcore("bilo",x,t,y,h,wx,wt,wc,b0,m0,off,ctrl)

```

 `gplm04.xpl`

Optionally, `gplmcore` can estimate the function $m(\bullet)$ on a grid, if `tg` and `m0g` are given. In addition, `gplmcore` can be also used to compute the biased parametric estimate which is needed for the specification test in Subsection 5.3. In this case the optional parameter `upb` should be set to 0 (default is 1).

4 Options

```

opt = gplmopt (string1, value1, ...{, opt})
      creates a list of options for GPLM estimation or appends options
      to an existing list

```

All options for the algorithm and optional parameters need to be collected in a list object. This allows just to set or to modify only those options which are necessary. All quantlets in the `gplm` quantlib (except for `gplmcore`) allow options. It is possible to give the same list of options to different routines. For example,

```
opt=gplmopt("miter",20,"name","MyDisplay")
```

will set the maximal number of iterations to 20 and the name of the output display to `MyDisplay`. Option lists used for the `glm` quantlib can be used as well.

With the above option settings, one can call first `gplmest` and then `gplmout`:

```

l=gplmest("bilo",x,y,opt)
gplmout("bilo",x,y,opt)

```

Both quantlets only consider those optional parameters which are intended for them. Hence `gplmest` will only care about `miter` whereas `gplmout` will only use the parameter name to present a display named `MyDisplay`.

4.1 Setting Options

As for the `glm` quantlib, it is recommended to use `gplmopt` to set the options. `gplmopt` is used in the same way as `glmopt`. Essentially, the possible options in the `gplm` quantlib are a superset of those in the `glm` quantlib. A list of options created with `glmopt` can hence be used or extended with `gplmopt`.

4.2 Grid and Starting Values

As shown in Subsection 3.1, it can be useful to estimate the nonparametric function $m(\bullet)$ not only on the observations `t`, but also on a grid `tg`. The optional parameter is:

`tg`
grid values (on the same scale as `t`)

This parameter can also be used to compute predictions for $m(\bullet)$ on other values than those given in `t`.

All presented algorithms for GPLM are iterative and require first an initialization step. Different strategies to initialize the iterative algorithm are possible:

- Start with $\tilde{\beta}$, $\tilde{m}(\bullet)$ from a parametric (GLM) fit.
- Alternatively, start with $\beta = 0$ 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 $\beta = 0$ and $m(t_j) \equiv G^{-1}(\bar{y})$.

The `gplm` quantlib uses the first method by default. If a different method is to be used, the necessary starting values can be given as optional input:

`b0`
initial values for the estimation of `b`.

`m0`
initial values for the estimation of `m`.

`m0g`
initial values for the estimation of `mg`.

4.3 Weights and Offsets

The estimation quantlet `gplmest` is able to handle special cases as weights and constraints on parameters (fix parameters). Setting weights and offsets is done in the same way as in the `glm` quantlib. Please consult the corresponding subsections of the GLM tutorial (Härdle, Klinke, and Müller 2000, Chapter 7).

Weights and offsets can always be given as a optional parameter. The corresponding components of the list of optional parameters are

`weights`
type of weights, either "`frequency`" for replication counts or "`prior`" for prior weights in weighted regression.

`wx`
weights, $n \times 1$ vector or scalar.

`wt`
trimming weights for estimation of the linear part, $n \times 1$ vector or scalar.

`wc`
weights to be used in the convergence criterion, $n \times 1$ vector or scalar.

`wr`
weights to be used in the modified LR test statistics, $n \times 1$ vector or scalar.

`off`
offset, $n \times 1$ vector or scalar.

None of these parameters should contains missing or infinity values. Defaults are `weights="prior"`, `wx=1`, `wt=1`, `wc=1`, `wr=1`, and `off=0`.

4.4 Control Parameters

There is a number of control parameters which modify the used algorithm.

meth

method to be used for GPLM estimation: -1 for backfitting, 0 for generalized Speckman estimator and 1 for profile likelihood. The default value is `meth=0` for the Speckman algorithm.

fscor

indicator for Fisher scoring (instead of Newton-Raphson optimization). `fscor=1` means that the Fisher scoring is used. Default is `fscor=0` for Newton-Raphson. This parameter is ignored for canonical link functions.

cnv

convergence criterion. The iteration stops when the relative change of the coefficients vector `b`, the estimated curve `m` and the deviance are less than `cnv`. Default is `cnv=0.0001`.

miter

maximal number of iterations. The iteration stops when this maximal number of iterations is reached. Default is `miter=10`.

nosort

`nosort=0` forces not to sort the data by the first column of `t` (and `tg`, if the optional grid `tg` is given). Default is `nosort=0`, i.e., to sort.

The following parameter switches on/off information during the computation.

shf

shows how the iteration is going on, if `shf=1` is set. Default is `shf=0`.

4.5 Model Parameters

These two parameters are only relevant for power link and negative binomial models, respectively:

pow

power for the power link function, default is `pow=0` (logarithmic link).

nbk

parameter k for the negative binomial distribution, the default is `nbk=1` (geometric distribution).

4.6 Specification Test

The modified LR test implemented in `gplmbootstraptest` (see Subsection 5.3) allows the following options:

wr

weights to be used in the modified LR test statistics, $n \times 1$ vector or scalar. The default value is `wr=1`.

tdesign

design matrix (in `t`) for the GLM hypothesis, $n \times r$ matrix. The default design is `matrix(n)~t`.

4.7 Output Modification

The `gplmout` routine which shows the output display provides some special possibilities to modify the output:

nopic

suppresses the output display if `nopic=1`. Default is `nopic=0`.

xvars

string vector, $p \times 1$, containing variable names for the columns of `x`.

name

single string, name for output and prefix for output displays from `gplmout`.

title

single string, title to be used in `gplmout`.

5 Statistical Evaluation and Presentation

5.1 Statistical Characteristics

```
stat = gplmstat (code, x, t, y, h, b, bv, m, df{, opt})
      computes statistical characteristics for an estimated GPLM
```

`gplmest` provides a number of statistical characteristics of the estimated model in the output component `stat`. The quantlet `gplmstat` can be used to create the above mentioned statistics by hand. Suppose we have input `x`, `y` and have estimated the vector of coefficients `b` (with covariance `bv`) and the nonparametric curve `m` by model "nopow". Then the list of statistics can be found from

```
stat=gplmstat("nopow",x,y,b,bv,m,df)
```

Of course, an list of options `opt` can be added at the end. If options from `opt` have been used for the estimation, these should be included for `gplmstat`, too.

The following characteristics are contained in the output `stat`. This itself is a list and covers the components

`df`

approximate degrees of freedom according to Hastie and Tibshirani (1990).

`deviance`

the deviance of the estimated model.

`pearson`

the Pearson statistic.

`loglik`

the log-likelihood of the estimated model, using the estimated dispersion parameter.

`dispersion`

an estimate for the dispersion parameter (`deviance/df`).

`aic, bic`

Akaike's AIC and Schwarz' BIC criterion, respectively.

`r2`, `adr2`
the (pseudo) coefficient of determination and its adjusted version, respectively.

`it`
the number of iterations needed.

`ret`
the return code, which is 0 if everything went without problems, 1 if the maximal number of iterations was reached, and negative if missing values have been encountered.

Sometimes, one or the other statistic may not be available, when it was not applicable. This can always be checked by searching for the components in `stat`:

```
names(stat)
```

The quantlet `names` will report all components of the list `stat`.

5.2 Output Display

```
gplmout (code, x, t, y, h, b, bv, m, stat{, opt})  
creates a nice output display for an estimated GPLM
```

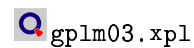
An output display containing statistical characteristics and a plot of the fitted link function can be obtained by `gplmout`.

Recall our example from Section 3:

```
opt=gplmopt("meth",1,"shf",1)  
opt=gplmopt("xvars",xvars,opt)  
opt=gplmopt("tg",grid(0|0,0.05|0.05,21|21),opt)  
g=gplmest("bilo",x,t,y,h,opt)
```

The optional component `xvars` will be used in the output display:

```
gplmout("bilo",x,t,y,h,g.b,g.bv,g.m,g.stat,opt)
```



produces the output given in Figure 3.

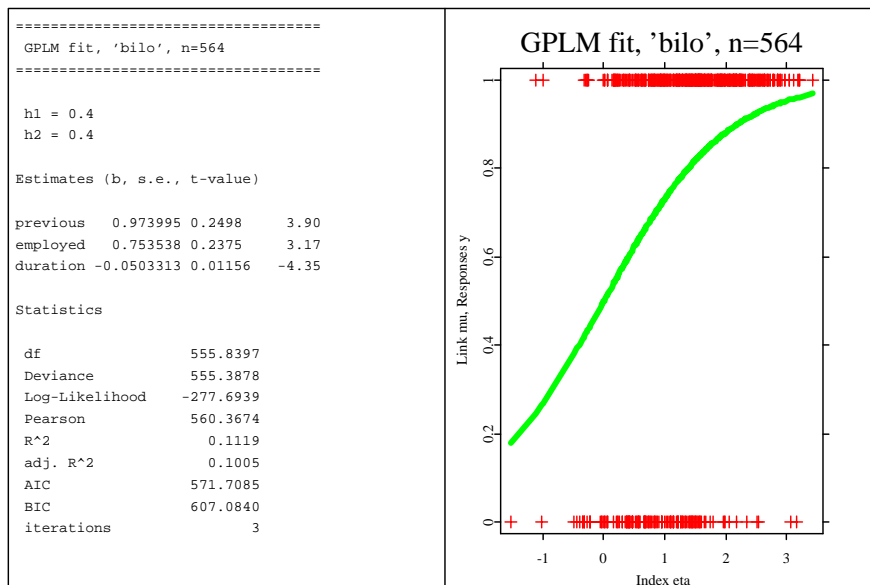


Figure 3: GPLM output display.

The optional parameters that can be used to modify the result from `gplmout` can be found in Subsection 4.7.

5.3 Model selection

```
g = gplmbootstraptest (code, x, t, y, h, nboot{, opt})
tests a GLM against the GPLM
```

To assess the estimated model it might be useful to check significance of single parameter values, or of linear combinations of parameters. To compare two different, nested models a sort of likelihood ratio (LR) test can be performed using the test statistic

$$R = 2 \sum_{i=1}^n L(\hat{\mu}_i, y_i) - L(\tilde{\mu}_i, y_i). \quad (5)$$

Here we denote the GLM fit by $\tilde{\mu}$ and the GPLM fit by $\hat{\mu}$. This approach corresponds fully to the parametric case, except that for the GPLM the approximate degrees of freedom have to be used. Please consult the corresponding subsections of the GLM tutorial (Härdle, Klinke, and Müller 2000, Chapter 7) for more information on the LR test.

A modified likelihood ratio test for testing $H_0 : G(X^T\beta + T^T\gamma + c)$ (GLM) against $H_1 : G\{X^T\beta + m(T)\}$ (GPLM) was introduced by Härdle, Mammen, and Müller (1998). They propose to use a “biased” parametric estimate $\bar{m}(t)$ instead of $t^T\tilde{\gamma} + c$ and the test statistic

$$R^\mu = 2 \sum_{i=1}^n L(\hat{\mu}_i, \hat{\mu}_i) - L(\bar{\mu}_i, \hat{\mu}_i). \quad (6)$$

Asymptotically, this test statistic is equivalent to

$$\tilde{R}^\mu = \sum_{i=1}^n w_i \left\{ x_i^T (\hat{\beta} - \tilde{\beta}) + \hat{m}(t_i) - \bar{m}(t_i) \right\}^2 \quad (7)$$

and

$$\tilde{R}_o^\mu = \sum_{i=1}^n w_i \left\{ x_i^T (\hat{\beta} - \tilde{\beta}) + \hat{m}(t_i) - \bar{m}(t_i) \right\}^2 \quad (8)$$

with


$$w_i = \frac{[G'\{x_i^T\hat{\beta} + \hat{m}(t_i)\}]^2}{V[G\{x_i^T\hat{\beta} + \hat{m}(t_i)\}]}$$

All three test statistics are asymptotically equivalent and have an asymptotic normal distribution. However, since the convergence to the limiting normal distribution is slow, it is recommended to determine the critical values of the test by bootstrap. The quantlet `gplmbootstraptest` performs this bootstrap test.

Let us continue with our credit scoring example and test whether the correct specification of the model is $G(X^T\beta + T^T\gamma + c)$ or $G\{X^T\beta + m(T)\}$. The following code computes first the GLM and applies the quantlet `gplmbootstraptest` to estimate the GPLM and perform the bootstrap test.

```
library("glm") ; GLM estimation
n=rows(x)
opt=glmopt("xvars",xvars|tvars|"constant")
l=glmest("bilo",x~t~matrix(n),y,opt)
glmout("bilo",x~t~matrix(n),y,l.b,l.bv,l.stat,opt)

library("gplm") ; GPLM estimation and test
h=0.4
nboot=10
randomize(742742)
opt=gplmopt("meth",1,"shf",1,"xvars",xvars)
opt=gplmopt("wr",prod((abs(t-0.5) < 0.40*trange),2),opt)
g=gplmbootstraptest("bilo",x,t,y,h,nboot,opt)
gplmout("bilo",x,t,y,h,g.b,g.bv,g.m,g.stat,opt)
```

 `gplm05.xpl`

Note the optional weight vector `wr` which defines weights for the test statistics. All observations outside a radius of 0.4 around the center of `t` are excluded. This is to ensure that the test result is not disturbed by outliers and boundary effects. Table 3 summarizes the coefficients from the output windows for the GLM (left column) and the GPLM (right) column.

The obtained significance levels for the test (computed for all three test statistics R^μ , \tilde{R}^μ and \tilde{R}_0^μ) can be found in the component `alpha` of the result `g`. Note that the approximations \tilde{R}^μ and \tilde{R}_0^μ (the latter in particular) may give bad results when the sample size n is small. If we run the test with random seed 742742 and `nboot=250` we get:

```
Contents of alpha
[1,] 0.035857
[2,] 0.035857
[3,] 0.043825
```

The hypothesis GLM can hence be rejected (at 5% level for R^μ , \tilde{R}^μ , \tilde{R}_0^μ).


It is also possible to test more complicated GLMs against the GPLM. For

	Coeff.	Coeff.	Coeff.
previous	0.974 (3.99)	0.954 (3.91)	0.974 (3.90)
employed	0.783 (3.34)	0.765 (3.26)	0.753 (3.17)
duration	-0.048 (-4.04)	-0.050 (-4.15)	-0.050 (-4.35)
amount	0.092 (-0.12)	1.405 (-1.09)	— —
age	0.989 (1.93)	2.785 (1.82)	— —
interaction	— —	-3.355 (-1.26)	— —
constant	0.916 (2.40)	0.275 (0.44)	— —
	GLM	GLM (interaction)	GPLM

Table 3: Coefficients from GLM (with and without interaction term) and GPLM, t -values in parentheses.

example, the nonlinear influence of **amount** and **age** could be caused by an interaction of these two variables. Consider now the GLM hypothesis $G(X^T\beta + T^T\gamma + \delta t_1 \cdot t_2 + c)$. The code for this test needs to define an optional design matrix `tdesign` which is used instead of the default `t~matrix(n)` in the previous test. The essential changes are as follows:

```
tdesign=t~prod(t,2)~matrix(n)
opt=gplmopt("tdesign",tdesign,opt)
g=gplmbootstraptest("bilo",x,t,y,h,nboot,opt)
```

 `gplm06.xpl`

The resulting coefficients for the GLM can be found in the middle column of Table 3. Performing the test with random seed 742742 and `nboot=250` yields:

```
Contents of alpha
[1,] 0.052
[2,] 0.056
[3,] 0.064
```

The hypothesis, that the correct specification is a GLM with interaction term, can hence be rejected as well (now at 10% level for R^μ , \tilde{R}^μ , \tilde{R}_o^μ).

Note that `gplmbootstraptest` also prints a warning, if missing values occurred in the bootstrap procedure. In our last example we have:

```

[1,] =====
[2,] WARNING!
[3,] =====
[4,] Missing values in bootstrap encountered!
[5,] The actually used bootstrap sample sizes are:
[6,]   nboot[1] =           249 ( 99.60%)
[7,]   nboot[2] =           249 ( 99.60%)
[8,]   nboot[3] =           249 ( 99.60%)
[9,] =====

```

Missing values are mostly due to numerical errors when the sample size is small or the dataset contains outliers.

References

- Fahrmeir, L. and Hamerle, A. (1984). *Multivariate Statistische Verfahren*, De Gruyter, Berlin.
- Fahrmeir, L. and Tutz, G. (1994). *Multivariate Statistical Modelling Based on Generalized Linear Models*, Springer.
- Härdle, W., Klinke, S., and Müller, M. (2000). *XploRe Learning Guide*, Springer.
- Härdle, W., Mammen, E., and Müller, M. (1998). Testing parametric versus semiparametric modelling in generalized linear models, *Journal of the American Statistical Association* **93**: 1461–1474.
- 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. (1997). Computer-assisted generalized partial linear models, in D. W. Scott (ed.), *Proceedings of the 29th Symposium on the Interface, Houston Texas, May 14–17, 1997*, Vol. 29/1, pp. 221–230.

- Severini, T. A. and Staniswalis, J. G. (1994). Quasi-likelihood estimation in semiparametric models, *Journal of the American Statistical Association* **89**: 501–511.
- Severini, T. A. and Wong, W. H. (1992). Generalized profile likelihood and conditionally parametric models, *Annals of Statistics* **20**: 1768–1802.
- Speckman, P. E. (1988). Regression analysis for partially linear models, *Journal of the Royal Statistical Society, Series B* **50**: 413–436.