

Müller, Marlene

Working Paper

Computer-assisted generalized partial linear models

SFB 373 Discussion Paper, No. 1997,48

Provided in Cooperation with:

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

Suggested Citation: Müller, Marlene (1997) : Computer-assisted generalized partial linear models, SFB 373 Discussion Paper, No. 1997,48, Humboldt University of Berlin, Interdisciplinary Research Project 373: Quantification and Simulation of Economic Processes, Berlin,
<https://nbn-resolving.de/urn:nbn:de:kobv:11-10064280>

This Version is available at:

<http://hdl.handle.net/10419/66270>

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.

Computer–assisted Generalized Partial Linear Models*

Marlene Müller
Humboldt University Berlin
Institute for Statistics and Econometrics
Spandauer Str. 1, D–10178 Berlin, Germany

June 12, 1997

Abstract

A particular semiparametric model of interest is the generalized partial linear model (GPLM) which allows a nonparametric modeling of the influence of the continuous co-variables.

The paper reviews different estimation procedures based on kernel methods and test procedures on the correct specification of this model (vs. a parametric generalized linear model). Simulations and an application to a data set on East–West German migration illustrate similarities and dissimilarities of the estimators and test statistics.

Semiparametric methods are highly demanding on software. Thus, the presentation is completed by indicating the practical implementation in new version of the statistical computing environment XploRe.

*To appear in: *Interface'97 Proceedings*, Houston, Texas.

The research for this paper was supported by Sonderforschungsbereich 373 at the Humboldt-University Berlin. The paper is printed using funds made available by the Deutsche Forschungsgemeinschaft.

1 Introduction

In recent years a good deal of work has been devoted to “generalize” generalized linear models (GLM). Workable alternatives to the parametric generalized linear model

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

are *single index models* (SIM), keeping the linear form of the index $\mathbf{X}^T\boldsymbol{\beta}$ but allowing G to be an arbitrary smooth function, and *generalized additive models* (GAM) that maintain G to be a known function but allow the argument inside to be a sum of unknown smooth functions. See Hastie & Tibshirani (1990), Härdle & Turlach (1992), Horowitz (1993), Powell, Stock & Stoker (1989), Weisberg & Welsh (1994) for more details on these models.

One of the reasons for the wide propagation of generalized linear models is the computational feasibility and in particular the ability to handle both discrete and continuous covariables in \mathbf{X} . A generalization should take care of these properties. The generalized linear model is a tool which covers many possible nonlinear relations between explanatory variables \mathbf{X} and the response variable Y . However, the shape of the nonlinear relation (link function G) is fixed, in most cases monotone.

A way to incorporate a nonmonotone dependence of the response on the continuous variables is given by a *generalized partial linear model* (GPLM)

$$E(Y|\mathbf{X}, \mathbf{T}) = G\{\mathbf{X}^T\boldsymbol{\beta} + m(\mathbf{T})\}, \quad (1)$$

where $\boldsymbol{\beta} = (\beta_1, \dots, \beta_p)^T$ is a finite dimensional parameter and $m(\bullet)$ is a smooth function. These models allow a nonparametric inclusion of a part of the explanatory variables. Here, we assume a decomposition of the explanatory variables into two vectors, \mathbf{X} and \mathbf{T} . \mathbf{X} denotes a p -variate random vector which usually covers discrete covariables. \mathbf{T} is a q -variate random vector of continuous covariables. In practice this can contain only those continuous variables which have most influence on the dependent variable Y .

The estimation methods for model (1) are based on the idea that an estimate $\hat{\boldsymbol{\beta}}$ can be found for known $m(\bullet)$, and an estimate $\hat{m}(\bullet)$ can be found for known $\boldsymbol{\beta}$. We will concentrate on *profile likelihood* estimation and *backfitting* in the following.

In order to estimate the GPLM by semiparametric maximum-likelihood, assume that the first two conditional moments of Y can be specified as

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

and 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

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

In the following, only the estimation of $\boldsymbol{\beta}$ and $m(\bullet)$ by means of the sample values $\{y_i, \mathbf{x}_i, \mathbf{t}_i\}$, $i = 1, \dots, n$ is discussed. The possible scale parameter σ can be obtained from

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

when $\hat{\mu}_i = G\{\mathbf{x}_i^T \hat{\boldsymbol{\beta}} + \widehat{m}(\mathbf{t}_i)\}$.

All estimation methods presented in the following are iterative and thus need starting values. Different strategies to initialize the iterative algorithm are possible:

- Start with $\tilde{\boldsymbol{\beta}}, \tilde{m}(\bullet)$ 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 $m(t_j) \equiv G^{-1}(\bar{y})$.

1.1 Profile Likelihood

The *profile likelihood* method considered in Severini & Wong (1992) aims to separate the estimation problem into a parametric and a nonparametric part. Its essential idea is to fix the parameter $\boldsymbol{\beta}$ and to estimate the nonparametric function in dependence of this fixed $\boldsymbol{\beta}$. The resulting estimate for $m_{\boldsymbol{\beta}}(\bullet)$ is then used to construct the profile likelihood for $\boldsymbol{\beta}$. As a consequence of the profile likelihood method, $\hat{\boldsymbol{\beta}}$ is asymptotically efficient. Severini & Staniswalis (1994) show that the resulting estimator $\hat{\boldsymbol{\beta}}$ is \sqrt{n} -consistent and asymptotically normal, and that estimators $\widehat{m}(\bullet) = \widehat{m}_{\hat{\boldsymbol{\beta}}}(\bullet)$ are consistent in supremum norm.

The algorithm can be motivated as follows. Consider the parametric (profile) likelihood function

$$\mathcal{L}(\boldsymbol{\beta}) = \sum_{i=1}^n L(\mu_{i,\boldsymbol{\beta}}, y_i), \quad (3)$$

$\mu_{i,\boldsymbol{\beta}} = G\{\mathbf{x}_i^T \boldsymbol{\beta} + m_{\boldsymbol{\beta}}(\mathbf{t}_i)\}$. This function is optimized to obtain an estimate for $\boldsymbol{\beta}$. The *smoothed* or *local* likelihood

$$\mathcal{L}^{\mathbf{H}}(m_{\boldsymbol{\beta}}(t)) = \sum_{i=1}^n \mathcal{K}_{\mathbf{H}}(\mathbf{t} - \mathbf{t}_i) L(\mu_{i,m_{\boldsymbol{\beta}}(\mathbf{t})}, y_i), \quad (4)$$

$\mu_{i,m_{\boldsymbol{\beta}}(\mathbf{t})} = G\{\mathbf{x}_i^T \boldsymbol{\beta} + m_{\boldsymbol{\beta}}(\mathbf{t})\}$ is optimized to estimate the smooth function $m_{\boldsymbol{\beta}}(\mathbf{t})$ at point \mathbf{t} . The local weights $\mathcal{K}_{\mathbf{H}}(\mathbf{t} - \mathbf{t}_i)$ here denote kernel weights with \mathcal{K} denoting a (multidimensional) kernel function and \mathbf{H} a bandwidth matrix.

Abbreviate now $m_j = m_{\boldsymbol{\beta}}(\mathbf{t}_j)$ and 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 . The maximization of the local likelihood (4) requires to solve

$$0 = \sum_{i=1}^n \ell'_i(\mathbf{x}_i^T \boldsymbol{\beta} + m_j) \mathcal{K}_{\mathbf{H}}(\mathbf{t}_i - \mathbf{t}_j). \quad (5)$$

For $\boldsymbol{\beta}$ we have from (3) to solve

$$0 = \sum_{i=1}^n \ell'_i(\mathbf{x}_i^T \boldsymbol{\beta} + m_i) \{\mathbf{x}_i + \mathbf{m}'_i\}. \quad (6)$$

A further differentiation of (5) leads to an expression for the derivative \mathbf{m}'_j of m_j with respect to $\boldsymbol{\beta}$

$$\mathbf{m}'_j = - \frac{\sum_{i=1}^n \ell''_i(\mathbf{x}_i^T \boldsymbol{\beta} + m_j) \mathcal{K}_{\mathbf{H}}(\mathbf{t}_i - \mathbf{t}_j) \mathbf{x}_i}{\sum_{i=1}^n \ell''_i(\mathbf{x}_i^T \boldsymbol{\beta} + m_j) \mathcal{K}_{\mathbf{H}}(\mathbf{t}_i - \mathbf{t}_j)}. \quad (7)$$

Equations (5)–(6) imply the following iterative Newton–Raphson type algorithm.

Algorithm (P)	
<ul style="list-style-type: none"> • <i>updating step for $\boldsymbol{\beta}$</i> 	$\boldsymbol{\beta}^{new} = \boldsymbol{\beta} - \mathcal{B}^{-1} \sum_{i=1}^n \ell'_i(\mathbf{x}_i^T \boldsymbol{\beta} + m_i) \tilde{\mathbf{x}}_i$ <p>with a Hessian type matrix</p> $\mathcal{B} = \sum_{i=1}^n \ell''_i(\mathbf{x}_i^T \boldsymbol{\beta} + m_i) \tilde{\mathbf{x}}_i \tilde{\mathbf{x}}_i^T$ <p>and</p> $\tilde{\mathbf{x}}_j = \mathbf{x}_j - \frac{\sum_{i=1}^n \ell''_i(\mathbf{x}_i^T \boldsymbol{\beta} + m_j) \mathcal{K}_{\mathbf{H}}(\mathbf{t}_i - \mathbf{t}_j) \mathbf{x}_i}{\sum_{i=1}^n \ell''_i(\mathbf{x}_i^T \boldsymbol{\beta} + m_j) \mathcal{K}_{\mathbf{H}}(\mathbf{t}_i - \mathbf{t}_j)}.$
<ul style="list-style-type: none"> • <i>updating step for m_j</i> 	$m_j^{new} = m_j - \frac{\sum_{i=1}^n \ell'_i(\mathbf{x}_i^T \boldsymbol{\beta} + m_j) \mathcal{K}_{\mathbf{H}}(\mathbf{t}_i - \mathbf{t}_j)}{\sum_{i=1}^n \ell''_i(\mathbf{x}_i^T \boldsymbol{\beta} + m_j) \mathcal{K}_{\mathbf{H}}(\mathbf{t}_i - \mathbf{t}_j)}.$

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

The updating step for m_j is of quite complex structure. In some models (in particular for identity and exponential link functions G) equation (5) can be solved explicitly for m_j . However, it is possible to rewrite the updating step for $\boldsymbol{\beta}$ in a form which is comparable with the estimators introduced later. Define \mathcal{S}^P the smoother matrix with elements

$$\mathcal{S}_{ij}^P = \frac{\ell''_i(\mathbf{x}_i^T \boldsymbol{\beta} + m_j) \mathcal{K}_{\mathbf{H}}(\mathbf{t}_i - \mathbf{t}_j)}{\sum_{i=1}^n \ell''_i(\mathbf{x}_i^T \boldsymbol{\beta} + m_j) \mathcal{K}_{\mathbf{H}}(\mathbf{t}_i - \mathbf{t}_j)} \quad (8)$$

and let \mathcal{X} be the design matrix with rows \mathbf{x}_i^T . We can rewrite the updating step for $\boldsymbol{\beta}$.

Algorithm (P)

- *updating step for β*

$$\beta^{new} = (\widetilde{\mathcal{X}}^T \mathcal{W} \widetilde{\mathcal{X}})^{-1} \widetilde{\mathcal{X}}^T \mathcal{W} \widetilde{\mathbf{z}}$$

with

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

\mathcal{I} denotes the identity matrix, \mathbf{v} is a vector and \mathcal{W} is a diagonal matrix containing the first and second derivatives of $\ell_i(\mathbf{x}_i^T \beta + m_i)$, respectively. The variable $\widetilde{\mathbf{z}}$ 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 $\widetilde{\mathcal{X}}$).

1.2 Simpler Variant of Profile Likelihood

The profile likelihood estimator is particularly easy to derive in case of a model with identity link G and normally distributed y_i . Here, $\ell'_i = y_i - \mathbf{x}_i^T \beta - m_j$ and $\ell''_i \equiv -1$. The latter yields the smoother matrix \mathcal{S} with elements

$$\mathcal{S}_{ij} = \frac{\mathcal{K}_{\mathbf{H}}(\mathbf{t}_i - \mathbf{t}_j)}{\sum_{i=1}^n \mathcal{K}_{\mathbf{H}}(\mathbf{t}_i - \mathbf{t}_j)}. \quad (9)$$

Moreover, the update for m_j simplifies to

$$m_j^{new} = \frac{\sum_{i=1}^n (y_i - \mathbf{x}_i^T \beta) \mathcal{K}_{\mathbf{H}}(\mathbf{t}_i - \mathbf{t}_j)}{\sum_{i=1}^n \mathcal{K}_{\mathbf{H}}(\mathbf{t}_i - \mathbf{t}_j)}$$

or simply

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

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

$$\beta^{new} = (\widetilde{\mathcal{X}}^T \widetilde{\mathcal{X}})^{-1} \widetilde{\mathcal{X}}^T \widetilde{\mathbf{y}}$$

with $\widetilde{\mathcal{X}} = (\mathcal{I} - \mathcal{S})\mathcal{X}$ and $\widetilde{\mathbf{y}} = (\mathcal{I} - \mathcal{S})\mathbf{y}$. These are the essentially the estimators for the partial linear model proposed by Speckman (1988).

Recall that each iteration step of a GLM can be obtained by a weighted least squares regression on an adjusted dependent variable. Hence, in the partial linear model the weighted least squares regression could be replaced by an partial linear fit on the adjusted dependent variable

$$\mathbf{z} = \mathcal{X}\beta + \mathbf{m} - \mathcal{W}^{-1}\mathbf{v}. \quad (10)$$

Now, \mathbf{v} is a vector and \mathcal{W} a diagonal matrix containing the first and second derivatives of $\ell_i(\mathbf{x}_i^T \boldsymbol{\beta} + m_i)$, respectively. See Hastie & Tibshirani (1990, p. 154) for a reference for this generalization of the Speckman estimator.

The basic simplification of this approach is to use the smoothing matrix \mathcal{S}

$$\mathcal{S}_{ij} = \frac{\ell_i''(\mathbf{x}_i^T \boldsymbol{\beta} + m_i) \mathcal{K}_{\mathbf{H}}(\mathbf{t}_i - \mathbf{t}_j)}{\sum_{i=1}^n \ell_i''(\mathbf{x}_i^T \boldsymbol{\beta} + m_i) \mathcal{K}_{\mathbf{H}}(\mathbf{t}_i - \mathbf{t}_j)}. \quad (11)$$

Note the difference in ℓ_i'' compared to the smoother matrix \mathcal{S}^P in the general case. We will come back to the computational simplification in Section 5.

An expression for each iteration step in matrix notation is possible here:

Algorithm (S)	
<ul style="list-style-type: none"> • <i>updating step for $\boldsymbol{\beta}$</i> 	$\boldsymbol{\beta}^{new} = (\widetilde{\mathcal{X}}^T \mathcal{W} \widetilde{\mathcal{X}})^{-1} \widetilde{\mathcal{X}}^T \mathcal{W} \widetilde{\mathbf{z}},$
<ul style="list-style-type: none"> • <i>updating step for \mathbf{m}</i> 	$\mathbf{m}^{new} = \mathcal{S}(\mathbf{z} - \mathcal{X} \boldsymbol{\beta})$
<p>using the notations</p> $\begin{aligned} \widetilde{\mathcal{X}} &= (\mathcal{I} - \mathcal{S}) \mathcal{X}, \\ \widetilde{\mathbf{z}} &= (\mathcal{I} - \mathcal{S}) \mathbf{z} = \widetilde{\mathcal{X}} \boldsymbol{\beta} - \mathcal{W}^{-1} \mathbf{v}. \end{aligned}$	

The update of the index $\mathcal{X} \boldsymbol{\beta} + \mathbf{m}$ in each iteration step can be expressed by a linear estimation matrix

$$\mathcal{X} \boldsymbol{\beta}^{new} + \mathbf{m}^{new} = \mathcal{R}^S \mathbf{z}$$

where

$$\mathcal{R}^S = \widetilde{\mathcal{X}} \{ \widetilde{\mathcal{X}}^T \mathcal{W} \widetilde{\mathcal{X}} \}^{-1} \widetilde{\mathcal{X}}^T \mathcal{W} (\mathcal{I} - \mathcal{S}) + \mathcal{S}. \quad (12)$$

This matrix \mathcal{R}^S will allow an (approximate) generalization of the likelihood ratio test in this case.

Recall, that the essential difference of algorithms (S) and (P) lies in the fact, that the latter always uses $\ell_i'(\mathbf{x}_i^T \boldsymbol{\beta} + m_i)$ and $\ell_i''(\mathbf{x}_i^T \boldsymbol{\beta} + m_i)$ instead of $\ell_i'(\mathbf{x}_i^T \boldsymbol{\beta} + m_j)$ and $\ell_i''(\mathbf{x}_i^T \boldsymbol{\beta} + m_j)$. Thus, there are some cases when both algorithms obviously should produce very similar results. This is first, when the bandwidth \mathbf{H} is small. Second, algorithms (S) and (P) will produce estimators that are close, when $m(\bullet)$ is relatively constant or small with respect to the parametric part. We point out in Sections 3 and 4 that both estimators (P) and (S) very often resemble each other.

1.3 Backfitting

The backfitting method was suggested as an iterative algorithm to fit an additive model (Hastie & Tibshirani 1990). The key idea is to regress the additive components separately

on partial residuals. The ordinary partial linear model (with identity link function)

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

is a special case, consisting of only two additive functions. Denote \mathcal{P} the projection matrix $\mathcal{P} = \mathcal{X}(\mathcal{X}^T \mathcal{X})^{-1} \mathcal{X}^T$ and \mathcal{S} a smoother matrix. Abbreviate $\mathbf{m} = (m_1, \dots, m_n)^T = (m(\mathbf{t}_1), \dots, m(\mathbf{t}_n))^T$ and $\mathbf{y} = (y_1, \dots, y_n)^T$. Then backfitting means to solve

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

In this case no iteration is necessary (Hastie & Tibshirani 1990, p. 118) and the explicit solution is given by

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

For a GPLM, backfitting means to perform an additive fit on the adjusted dependent variable which was defined in (10). We use again the kernel smoother matrix \mathcal{S} from (11). As in algorithm (S), an explicit expression for each iteration step is possible. Note the difference in the updating step for $\boldsymbol{\beta}$. Also, as the algorithm (S), backfitting shares the property of being linear on the variable \mathbf{z} :

$$\mathcal{X}\boldsymbol{\beta}^{new} + \mathbf{m}^{new} = \mathcal{R}^B \mathbf{z}$$

with

$$\mathcal{R}^B = \widetilde{\mathcal{X}}\{\mathcal{X}^T \mathcal{W} \widetilde{\mathcal{X}}\}^{-1} \mathcal{X}^T \mathcal{W}(\mathcal{I} - \mathcal{S}) + \mathcal{S}, \quad (13)$$

see Hastie & Tibshirani (1990, p. 173).

Algorithm (B)	
• <i>updating step for $\boldsymbol{\beta}$</i>	
	$\boldsymbol{\beta}^{new} = (\mathcal{X}^T \mathcal{W} \widetilde{\mathcal{X}})^{-1} \mathcal{X}^T \mathcal{W} \tilde{\mathbf{z}},$
• <i>updating step for \mathbf{m}</i>	
	$\mathbf{m}^{new} = \mathcal{S}(\mathbf{z} - \mathcal{X}\boldsymbol{\beta}),$
using the notations	
	$\begin{aligned} \widetilde{\mathcal{X}} &= (\mathcal{I} - \mathcal{S})\mathcal{X}, \\ \tilde{\mathbf{z}} &= (\mathcal{I} - \mathcal{S})\mathbf{z} = \widetilde{\mathcal{X}}\boldsymbol{\beta} - \mathcal{W}^{-1}\mathbf{v}. \end{aligned}$

In practice, often some of the predictors are correlated. Hastie & Tibshirani (1990) therefore propose also a modified backfitting algorithm, which first searches for a (parametric) solution and only fits the remaining parts nonparametrically. For the simulations (Section 3), we considered additionally modified backfitting and refer to it as algorithm (MB).

2 Testing the GPLM

Having estimated the influence $m(\bullet)$ of the covariables T , it is naturally to ask, whether the estimate $\widehat{m}(\bullet)$ is significantly different from a linear function obtained by a parametric GLM fit. Typically, a test procedure for

$$\begin{aligned} H_0 &: m(t) = \mathbf{t}^T \boldsymbol{\gamma} + \gamma_0 \\ H_1 &: m(\bullet) \text{ is an arbitrary smooth function,} \end{aligned}$$

is based on semiparametric generalizations of the parametric likelihood ratio test. Hastie & Tibshirani (1990) propose to use the difference of the deviances of the linear and the semiparametric model, respectively, and to approximate the degrees of freedom in the semiparametric case. The asymptotic behavior of this method is unknown, though. Härdle, Mammen & Müller (1996) derive an asymptotic normal distribution for a slightly modified test statistic. We will concentrate on these two approaches here.

2.1 Likelihood ratio test and approximate degrees of freedom

Denote the the semiparametric estimates $\hat{\mu}_i = G\{\mathbf{x}_i^T \hat{\boldsymbol{\beta}} + \widehat{m}(\mathbf{t}_i)\}$ and the parametric estimates $\tilde{\mu}_i = G\{\mathbf{x}_i^T \tilde{\boldsymbol{\beta}} + \mathbf{t}_i^T \tilde{\boldsymbol{\gamma}} + \tilde{\gamma}_0\}$. A natural approach is to compare both estimates by a likelihood ratio test statistic

$$R = 2 \sum_{i=1}^n L(\hat{\mu}_i, y_i) - L(\tilde{\mu}_i, y_i). \quad (14)$$

which would have an asymptotic χ^2 distribution if the estimates $\hat{\mu}_i$ were from an nesting parametric fit.

In the semiparametric case, this test statistic seems to work well in practical situations. However, an approximate number of degrees of freedom needs to be defined for the GPLM. The basic idea of Hastie & Tibshirani (1990, p. 155–158) is the following. Denote by

$$D(\mathbf{y}, \hat{\boldsymbol{\mu}}) = 2 \sum_{i=1}^n L(\mu_i^{max}, y_i) - L(\hat{\mu}_i, y_i), \quad (15)$$

the *deviance* in observations y_i and fitted values $\hat{\mu}_i$. μ_i^{max} are the parameter values that maximize the log-likelihood $\sum_i L(\bullet, y_i)$. Abbreviate the estimated index $\hat{\boldsymbol{\eta}} = \mathcal{X} \hat{\boldsymbol{\beta}} + \widehat{m}$ and suppose an adjusted dependent variable $\mathbf{z} = \hat{\boldsymbol{\eta}} - \mathcal{W}^{-1} \mathbf{v}$. If at convergence of the iterative estimation

$$\hat{\boldsymbol{\eta}} = \mathcal{R} \mathbf{z} = \mathcal{R}(\hat{\boldsymbol{\eta}} - \mathcal{W}^{-1} \mathbf{v}) \quad (16)$$

with a linear operator \mathcal{R} , then

$$D(\mathbf{y}, \hat{\boldsymbol{\mu}}) \approx (\mathbf{z} - \hat{\boldsymbol{\eta}})^T \mathcal{W}^{-1} (\mathbf{z} - \hat{\boldsymbol{\eta}}) \quad (17)$$

which has approximately

$$df^{err}(\hat{\boldsymbol{\mu}}) = n - \text{tr} \left(2\mathcal{R} - \mathcal{R}^T \mathcal{W} \mathcal{R} \mathcal{W}^{-1} \right) \quad (18)$$

degrees of freedom. In practice, the computation of the trace $\text{tr} \left(\mathcal{R}^T \mathcal{W} \mathcal{R} \mathcal{W}^{-1} \right)$ can be rather difficult. A simpler approximation is

$$df^{err}(\hat{\boldsymbol{\mu}}) = n - \text{tr}(\mathcal{R}) \quad (19)$$

which were correct if \mathcal{R} were a projection operator.

Now, for the comparison of the semiparametric $\hat{\mu}$ and the parametric $\tilde{\mu}$ the test statistic (14) can be expressed by

$$R = D(\mathbf{y}, \tilde{\mu}) - D(\mathbf{y}, \hat{\mu})$$

and should follow approximately a χ^2 distribution with $df^{err}(\tilde{\mu}) - df^{err}(\hat{\mu})$ degrees of freedom.

Property (16) holds for the algorithms (B) and (S) with matrices \mathcal{R}^B and \mathcal{R}^S , respectively. A direct application to the profile likelihood algorithm (P) is not possible because of the more involved estimation of the nonparametric function $m(\bullet)$. However a workable approximation can be obtained by using the degrees of freedom from

$$\mathcal{R}^P = \tilde{\mathcal{X}}\{\tilde{\mathcal{X}}^T \mathcal{W} \tilde{\mathcal{X}}\}^{-1} \tilde{\mathcal{X}}^T \mathcal{W} (\mathcal{I} - \mathcal{S}^P) + \mathcal{S}, \quad (20)$$

where $\tilde{\mathcal{X}}$ denotes $(\mathcal{I} - \mathcal{S}^P)\mathcal{X}$.

2.2 Modified likelihood ratio test

A direct comparison of the semiparametric estimates $\hat{\mu}_i$ and the parametric estimates $\tilde{\mu}_i$ can be misleading because $\widehat{m}(\bullet)$ has a non-negligible smoothing bias, even under linearity hypothesis. Hence, Härdle et al. (1996) propose to use a bias-corrected parametric estimate $\overline{m}(\mathbf{t}_j)$ instead of $\mathbf{t}_j^T \tilde{\gamma} + \tilde{\gamma}_0$. This estimate can be obtained from the updating procedure for m_j on the parametric estimate. Note that here the second argument of $L(\bullet, \bullet)$ should be the parametric estimate of $E(y_i | \mathbf{x}_i, \mathbf{t}_i)$ instead of y_i which means to apply the smoothing step according to (4) to the artificial data set $\{G(\mathbf{x}_i^T \tilde{\beta} + \mathbf{t}_i^T \tilde{\gamma} + \tilde{\gamma}_0), \mathbf{x}_i, \mathbf{t}_i\}, i = 1, \dots, n$.

Using this bias-corrected parametric estimate $\overline{m}(\bullet)$, Härdle et al. (1996) propose the test statistic

$$R^\mu = 2 \sum_{i=1}^n L(\hat{\mu}_i, \hat{\mu}_i) - L(\bar{\mu}_i, \hat{\mu}_i), \quad (21)$$

where $\bar{\mu}_i = G\{\mathbf{x}_i^T \tilde{\beta} + \overline{m}(\mathbf{t}_i)\}$ is the bias corrected GLM fit and $\hat{\mu}_i$ is the semiparametric GPLM fit to the observations. Asymptotically, this test statistic is equivalent to

$$\tilde{R}^\mu = \sum_{i=1}^n w_i \left\{ \mathbf{x}_i^T (\hat{\beta} - \tilde{\beta}) + \widehat{m}(\mathbf{t}_i) - \overline{m}(\mathbf{t}_i) \right\}^2 \quad (22)$$

with

$$w_i = \frac{[G'\{\mathbf{x}_i^T \hat{\beta} + \widehat{m}(\mathbf{t}_i)\}]^2}{V[G\{\mathbf{x}_i^T \hat{\beta} + \widehat{m}(\mathbf{t}_i)\}]}.$$

Hence the resulting test statistic can be interpreted as a weighted quadratic difference of the (bias corrected) parametric predictor $\mathbf{x}_i^T \tilde{\beta} + \overline{m}(\mathbf{t}_i)$ and the semiparametric predictor $\mathbf{x}_i^T \hat{\beta} + \widehat{m}(\mathbf{t}_i)$.

Both test statistics R^μ and \tilde{R}^μ have the same asymptotic normal distribution if the profile likelihood algorithm (P) is used. (A χ^2 approximation does not hold in this case since kernel smoother matrices are not projection operators.) It has been pointed out in Härdle et al. (1996) that the normal approximation does not work well. Therefore, for the calculation of quantiles, it is recommended to use a bootstrap approximation of the quantiles of the test statistic:

1. Generate samples y_1^*, \dots, y_n^* with

$$\begin{aligned} E^*(y_i^*) &= G(\mathbf{x}_i^T \tilde{\boldsymbol{\beta}} + \mathbf{t}_i^T \tilde{\boldsymbol{\gamma}} + \gamma_0) \\ \text{Var}^*(y_i^*) &= \hat{\sigma}^2 V\{G(\mathbf{x}_i^T \tilde{\boldsymbol{\beta}} + \mathbf{t}_i^T \tilde{\boldsymbol{\gamma}} + \gamma_0)\}. \end{aligned}$$

2. Calculate estimates based on the bootstrap samples and finally the test statistics R^* . The quantiles of the distribution of R are estimated by the quantiles of the conditional distributions of R^* .

There are several possibilities for the choice of the conditional distribution of the y_i^* 's. In a binary response model, the distribution of y_i is completely specified by $\mu_i = G(\mathbf{x}_i^T \boldsymbol{\beta} + \mathbf{t}_i^T \boldsymbol{\gamma} + \gamma_0)$. If the distribution of y_i cannot be specified (apart from the first two moments) one should use wild bootstrap, see Härdle & Mammen (1993).

3 Simulations

To compare the competing estimators and test statistics some simulations have been performed. A logit model was used to simulate data:

$$E(Y|\mathbf{X}, T) = P(Y = 1|\mathbf{X}, T) = F\{\mathbf{X}^T \boldsymbol{\beta} + m(T)\}$$

with $F(\bullet)$ denoting the cumulative distribution function of the (standard) logistic distribution. The simulations were run

$$\begin{aligned} \text{under the hypothesis:} \quad m(t) &= \frac{3}{2}t, \\ \text{under the alternative:} \quad m(t) &= \frac{3}{2} \cos(\pi t), \end{aligned}$$

with $\boldsymbol{\beta} = (\beta_1, \beta_2)^T = (1, -1)^T$ in both cases. For the design of the explanatory variables two patterns are used. In both design patterns T and X_1 are independent and uniform on $[-1, 1]$. The variable X_2 is defined by an discretization of $\cos\{\pi(\rho T + (1 - \rho)U)\}$ where U is independent from T and X_1 and uniform on $[-1, 1]$ as well. We use

$$\begin{aligned} \text{independent design:} \quad \rho &= 0, \\ \text{dependent design:} \quad \rho &= 0.7. \end{aligned}$$

Throughout all computations in the Quartic kernel $K(u) = \frac{15}{16}(1 - u^2)^2 I(|u| \leq 1)$ was used for the kernel weights.

First, we compare the fits obtained by the different algorithms (P), (S), (B) and (MB) (modified backfitting). Tables 1 and 2 show the mean *average squared errors* (ASE's) as well as mean deviances and degrees of freedom in the independence and dependence design, respectively. By ASE for μ we mean

$$\frac{1}{n} \sum_{i=1}^n (\hat{\mu}_i - \mu_i)^2.$$

ASE for m is defined accordingly. ASE for $\boldsymbol{\beta} = (\beta_1, \beta_2)^T$ is the sum of the ASE's for both components β_1 and β_2 . The mean ASE's in Tables 1 and 2 are averaged over all simulations.

	$\hat{\mu}$	\hat{m}	$\hat{\beta}$	D	df^{err}
GPLM (P)	1.323	0.255	0.187	101.726	94.47
GPLM (S)	1.271	0.243	0.164	101.947	94.52
GPLM (B)	1.229	0.242	0.143	102.122	94.53
GPLM (MB)	1.229	0.242	0.143	102.124	94.50
$n = 100, h = 0.6$					
GPLM (P)	0.582	0.115	0.064	261.143	243.83
GPLM (S)	0.577	0.117	0.060	261.468	243.88
GPLM (B)	0.559	0.121	0.056	261.698	243.90
GPLM (MB)	0.559	0.121	0.056	261.699	243.87
$n = 250, h = 0.5$					
GPLM (P)	0.316	0.064	0.035	523.840	492.89
GPLM (S)	0.318	0.065	0.033	524.233	492.93
GPLM (B)	0.316	0.068	0.032	524.473	493.95
GPLM (MB)	0.316	0.068	0.032	524.473	493.93
$n = 500, h = 0.4$					

Table 1: Mean ASE's for μ ($\times 100$), m and β , mean deviances and mean degrees of freedom. Model under alternative, independent design, 500 Monte-Carlo replications.

	$\hat{\mu}$	\hat{m}	$\hat{\beta}$	D	df^{err}
GPLM (P)	1.416	0.280	0.277	115.986	94.44
GPLM (S)	1.351	0.273	0.273	116.190	94.55
GPLM (B)	1.403	0.344	0.345	116.823	94.66
GPLM (MB)	1.403	0.344	0.345	116.823	94.48
$n = 100, h = 0.6$					
GPLM (P)	0.654	0.128	0.096	295.133	243.77
GPLM (S)	0.641	0.131	0.091	295.442	243.88
GPLM (B)	0.697	0.193	0.168	296.618	243.99
GPLM (MB)	0.697	0.193	0.167	296.607	243.84
$n = 250, h = 0.5$					
GPLM (P)	0.342	0.068	0.050	594.150	492.82
GPLM (S)	0.340	0.070	0.049	594.497	492.92
GPLM (B)	0.382	0.106	0.094	595.873	493.02
GPLM (MB)	0.382	0.106	0.093	595.864	492.89
$n = 500, h = 0.4$					

Table 2: Mean ASE's for μ ($\times 100$), m and β , mean deviances and mean degrees of freedom. Model under alternative, dependent design, 500 Monte-Carlo replications.

α	0.01	0.05	0.10	0.20
R parametric	0.016	0.056	0.108	0.224
R (P)	0.016	0.084	0.180	0.348
R (S)	0.016	0.080	0.168	0.348
R (B)	0.016	0.076	0.184	0.376
R (MB)	0.008	0.072	0.156	0.300
R^μ (P) bootstrap	0.032	0.056	0.108	0.212
$n = 100, h = 0.6$				
R parametric	0.020	0.052	0.104	0.188
R (P)	0.020	0.056	0.144	0.316
R (S)	0.016	0.060	0.144	0.312
R (B)	0.016	0.060	0.144	0.324
R (MB)	0.012	0.052	0.140	0.292
R^μ (P) bootstrap	0.028	0.044	0.100	0.196
$n = 250, h = 0.5$				

Table 3: Percentage of rejections. Model under hypothesis, dependent design, 250 Monte–Carlo replications.

α	0.01	0.05	0.10	0.20
R parametric	0.836	0.944	0.972	0.988
R (P)	0.764	0.916	0.960	0.984
R (S)	0.768	0.920	0.964	0.992
R (B)	0.760	0.928	0.964	0.996
R (MB)	0.728	0.904	0.952	0.984
R^μ (P) bootstrap	0.832	0.936	0.952	0.976
$n = 100, h = 0.6$				
R parametric	1.000	1.000	1.000	1.000
R (P)	0.996	1.000	1.000	1.000
R (S)	0.996	1.000	1.000	1.000
R (B)	0.996	1.000	1.000	1.000
R (MB)	0.996	1.000	1.000	1.000
R^μ (P) bootstrap	0.996	1.000	1.000	1.000
$n = 250, h = 0.5$				

Table 4: Percentage of rejections. Model under alternative, dependent design, 250 Monte–Carlo replications.

The mean values can of course only give a rough impression. A closer inspection of the simulation results reveals, that indeed the deviance is almost always minimized by the profile likelihood method (P). Looking at the averaged squared errors, backfitting algorithms usually perform better for independent design and smaller sample size. This changes significantly, when dependences among the explanatory variables are present and

the sample size grows. Here, the averaged squared errors become typically lowest for algorithm (S). As Tables 1 and 2 indicate, both profile likelihood estimates (P) and (S) and both backfitting procedures (B) and (MB) get close when n increases and h decreases.

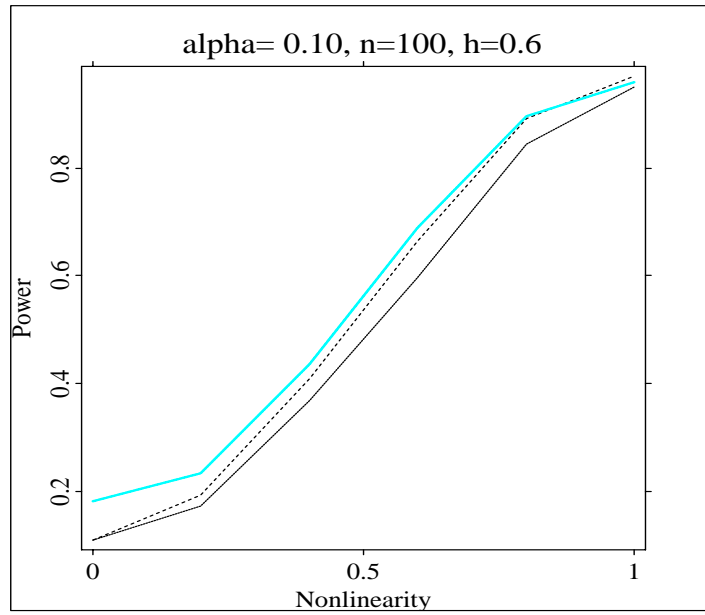


Figure 1: Power of likelihood ratio statistics $R(P)$, $R^\mu(P)$ and R parametric (grey, black and dashed). $n = 100$, dependent design, 250 Monte-Carlo replications.

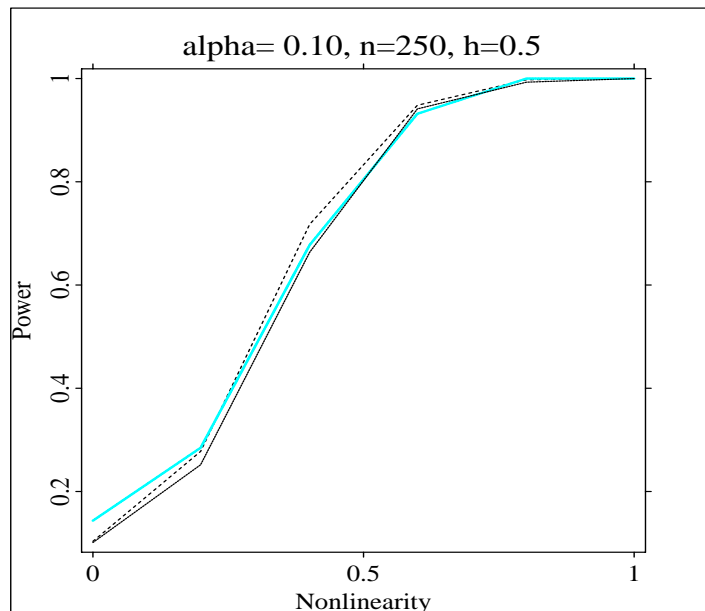


Figure 2: Power of likelihood ratio statistics $R(P)$, $R^\mu(P)$ and R parametric (grey, black and dashed). $n = 250$, dependent design, 250 Monte-Carlo replications.

Next, we compare all algorithms with respect to testing. As a parametric reference test we use the likelihood ratio test that tests the hypothesis $m(t) = \gamma_1 t + \gamma_0$ against the alternative $m(t) = \gamma_2 \cos(\pi t) + \gamma_1 t + \gamma_0$. Tables 3 and 4 summarize the results for the true hypothesis $m(t) = \frac{3}{2}t$ (Table 3) and the true alternative $m(t) = \frac{3}{2}\cos(\pi t)$ (Table 4) for different nominal significance levels α and different sample sizes n . It is clear to recognize, that all tests based on the approximate degrees of freedom have very similar power but are too rejective on the hypothesis. The bootstrapped R^μ for algorithm (P) follows the given nominal level under the hypothesis quite well and does not loose power under the alternative compared to the other test procedures.

This can also be seen from Figures 1 and 2, where power functions for R (P), bootstrapped R^μ (P) in comparison with the fully parametric likelihood ratio test are shown. The significance level α is fixed to 0.10 here. The true underlying function $m(\bullet)$ is a convex combination of $\frac{3}{2}t$ and $\frac{3}{2}\cos(\pi t)$ such that increasing nonlinearity means more weight is given to the cosine term.

4 Example: Migration

We illustrate the semiparametric estimation and the test procedure with an real data example on East–West German migration. The sample consists of East Germans, which have been surveyed in 1991 in the German Socio-Economic Panel, see GSOEP (1991). Among other questions the East German participants have been asked, if they can imagine to move to the western part of Germany or West Berlin. We give the value $y_i = 1$ to those who responded positive and 0 if not.

	Yes	No	(in %)	
Y migration	39.9	60.1		
X_1 family/friends	88.8	11.2		
X_2 unemployed	21.1	78.9		
X_3 city size	35.8	64.2		
X_4 female	50.2	49.8		
	Min	Max	Mean	S.D.
X_5 age (years)	18	65	39.93	12.89
T income (DM)	400	4000	2262.22	769.82

Table 5: Descriptive statistics for migration data. Sample from Mecklenburg–Vorpommern, $n = 402$.

In the following, we present the estimation results for Mecklenburg–Vorpommern (a state in the very North of Eastern Germany, $n = 402$). Some descriptive statistics is summarized in Table 5. Table 6 shows on the left the results of the parametric logit fit for Mecklenburg–Vorpommern. For simplicity both continuous variables (age, household income) have been linearly transformed to $[0, 1]$.

The migration intention is definitely determined by age (X_5). However, also the unemployment (X_2), city size (X_3) and household income (T) variables are significant. A further analysis of this data set by a generalized additive model showed that age has a

	Logit (t value)	(P)	(S)	(B)	(MB)
const.	-0.358 (-0.68)	—	—	—	—
X_1	0.589 (1.54)	0.600	0.599	0.395	0.595
X_2	0.780 (2.81)	0.800	0.794	0.765	0.779
X_3	0.822 (3.39)	0.842	0.836	0.784	0.815
X_4	-0.388 (-1.68)	-0.402	-0.400	-0.438	-0.394
X_5	-3.364 (-6.92)	-3.329	-3.313	-3.468	3.334
T	1.084 (1.90)	—	—	—	—
	Linear (GLM)	Part. Linear (GPLM)			

Table 6: Logit coefficients and coefficients in GPLM for migration data. Sample from Mecklenburg–Vorpommern. $n = 402$, $h = 0.3$ for the GPLM.

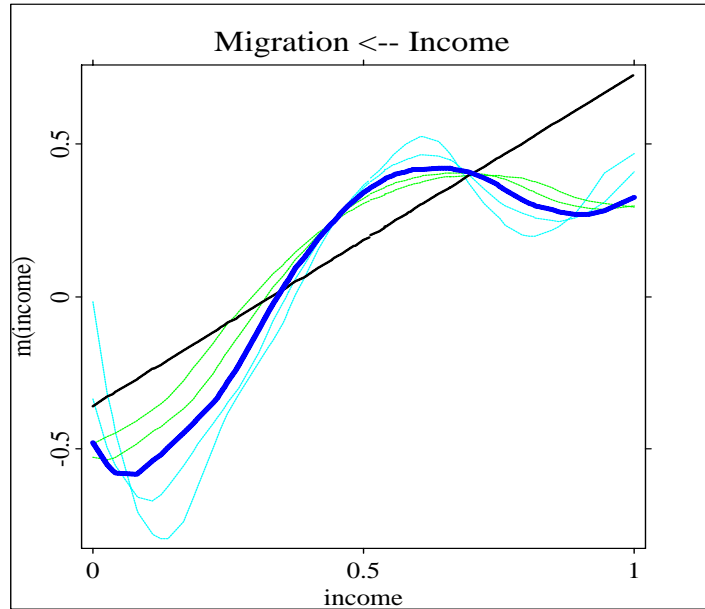


Figure 3: GPLM logit fit for migration data. Profile likelihood estimator (P) for m , with $h = 0.3$ (thick curve), $h = 0.2$, $h = 0.25$, $h = 0.35$, $h = 0.4$ (thin curves) and parametric logit fit (medium line).

nearly perfect linear influence. Because of this relation, we use a generalized partial linear model with a logistic link function and only the influence of household income modeled as a nonparametric function. The coefficients for the parametric covariates are given on the right side of Table 6.

The nonparametric estimate $\widehat{m}(\bullet)$ in this example seems to be an obvious nonlinear function, see Figure 3. As already observed in the simulations, both profile likelihood methods (P) and (S) give very similar results (Figure 4). Also the estimation from modified backfitting (MB) does not differ much. The backfitting estimator (B) however differs, depending on the size of h . For smaller h , the estimates from all algorithms (P), (S), (B) and (MB) are very near. This is caused by the kernel weights, which mainly use the

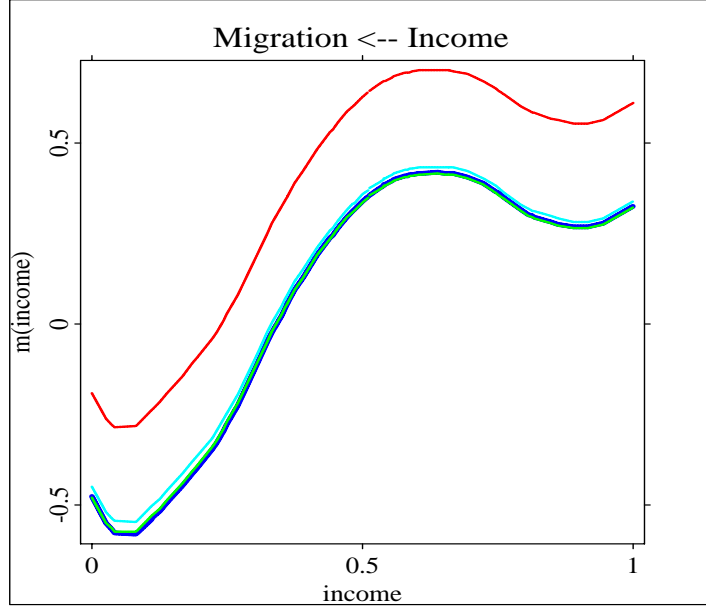


Figure 4: GPLM logit fit for migration data. Profile likelihood (P) and (S), modified backfitting (lower curves), backfitting (upper curve), $h = 0.3$.

observation i itself to estimate $m(t_i)$.

Going back to Table 6, one sees a difference in the coefficients for the parametric part as well. Both “negative influence” variables X_4 , X_5 get more weight by algorithm (B), the other variables get less which implies a positive vertical shift in the estimate for $m(\bullet)$ see upper curve in Figure 4. The shape of the backfitted $\widehat{m}(\bullet)$ is very similar though, which is to be explained by the identical smoothing step for \mathbf{m} in algorithms (B) and (S).

The obvious difference between backfitting (B) and the other procedures is an interesting observation. It is most likely due to the multivariate dependence structure within of the explanatory variables, an effect which is not easily reflected in simulations. The profile likelihood methods (P) and (S) have by construction a similar correction ability for concavity as the modified backfitting (MB) has.

Finally consider the testing problem for this example. By Figures 3 and 4 it is difficult to judge significance of the nonlinearity. For this real data example, it cannot be excluded that the difference between the nonparametric and the linear fit may be caused by boundary and bias problems of $\widehat{m}(\bullet)$. Additionally, the covariable “age” (included in a linear way) has dominant influence in the model.

Table 7 shows the results of the application of the different test statistics for different choices of the bandwidth h . As we have seen in the simulations, the likelihood ratio test statistic R and the modified test statistic R^μ in combination with bootstrap give very similar results. The number of bootstrap simulations has been chosen as $n^* = 400$. Linearity is clearly rejected (at 10% level) for all bandwidths from 0.2 to 0.4.

The different behavior of the tests for different h give some indication on possible deviance of $m(\bullet)$ from linear functions. The appearance of small wiggles of small length seems not significant for the bootstrap ($h = 0.2$). Also, the bootstrapped R^μ still rejects for $h = 0.5$. This is due to the comparison of the semiparametric estimator with a bias

h	0.20	0.25	0.30	0.35	0.40
R (P)	0.066	0.054	0.048	0.045	0.035
R (S)	0.068	0.055	0.047	0.044	0.033
R (B)	0.073	0.064	0.062	0.069	0.068
R (MB)	0.068	0.056	0.048	0.045	0.035
R^μ (P) bootstrap	0.065	0.054	0.042	0.042	0.045

Table 7: Observed significance levels for linearity test for migration data, $n = 402$. 400 bootstrap replications.

corrected parametric one, yielding more independence of the bandwidth.

5 Computational Issues

Generalizing the generalized linear model causes increasing complexity and thus demands for an efficient computational implementation. XploRe is a flexible and extensible environment that has been designed for a large scale of statistical tasks ranging from data analysis to complex smoothing algorithms. We want to stress that user-extensible functions also provide transparency of the implemented procedures.

Generalized partial linear models in XploRe are available from the library `gplm`, see Müller, Rönz & Härdle (1997). The algorithm for GPLM requires first an initialization step, this is done by default by a parametric GLM fit with the same link function. Next, the smoothing step for the nonparametric function $m(\bullet)$ has to be carried out. Consider the profile likelihood estimator first. The updating step for $m_j = m_{\beta}(\mathbf{t}_j)$ requires a ratio with numerator and denominator of convolution type

$$\sum_{i=1}^n \delta_{ij} K_{\mathbf{H}}(\mathbf{t}_i - \mathbf{t}_j), \quad (23)$$

where δ_{ij} is a derivative of the log-likelihood. Note, that this has to be done at least for all \mathbf{t}_j ($j = 1, \dots, n$) since the updated values of $m(\bullet)$ at all observation points are required in the updating step for β . Thus $O(n^2)$ operations are necessary and in each of this operations, the kernel function $K_{\mathbf{H}}(\mathbf{t}_i - \mathbf{t}_j)$ and both likelihood derivatives need to be evaluated.

As a consequence, this GPLM estimator in XploRe is implemented in a hybrid fashion. To estimate a logit GPLM, the user calls a macro written in XploRe. This macro itself calls two compiled functions which perform the update of $m(\mathbf{t})$ and calculate $\widetilde{\mathcal{X}}$ in an efficient way. Both functions are written in C and available from a shared library, which is dynamically linked to XploRe at runtime. The speed of operations in such compiled functions is comparable to that of XploRe internal commands.

In contrast to internal commands, however, experienced users can modify the supplied C source code or add own extensions. This allows the required flexibility, extensibility and transparency for the implementation.

Note that the evaluation of (23) is a standard procedure if δ_{ij} would be only dependent on i (e.g. in Nadaraya–Watson kernel regression). This is the case for algorithms (S)

and (B). Here, XploRe’s predefined kernel convolution can be used. In difference to the profile likelihood estimation (P), the derivatives δ_i can be stored and need to be computed only once in each iteration step. Still, $O(n^2)$ operations are necessary, the practical computation of algorithms (S) and (B) is three to four times faster though. Also, algorithms (S) and (B) often need less iterations and seems to be more stable for outliers in the design. Hence, algorithm (S), which seems to be a nice compromise between accuracy and computational efficiency is used as the default estimation method for a GPLM in XploRe.

6 Summary and Conclusions

The paper compares three estimation methods for generalized partial linear models (GPLM). The estimators are investigated for their small sample properties by means of simulated and real data.

- Backfitting outperforms profile likelihood estimation for a GPLM under independent design. However, if the explanatory variables are dependent, a variant of the profile likelihood method (simple profile likelihood method) seems to work best.
- When testing the parametric versus the semiparametric specification is concerned, the investigated profile likelihood variants perform similar. A likelihood ratio test with an approximate number of degrees of freedom seems to work reasonably well. A bootstrap modification of the likelihood ratio statistic enhance accuracy of the test results and makes results more independent of the bandwidth.
- The simple profile likelihood method can be considered as a good compromise between accuracy and computational efficiency in estimation and specification testing.

Acknowledgments

I thank Joan G. Staniswalis for her discussion at The Interface’97 conference and Berwin A. Turlach for helpful remarks concerning the modified backfitting estimator.

References

- GSOEP (1991). *Das Sozio-ökonomische Panel (SOEP) im Jahre 1990/91*, Projektgruppe “Das Sozio-ökonomische Panel”, Deutsches Institut für Wirtschaftsforschung. Vierteljahreshefte zur Wirtschaftsforschung, pp. 146–155.
- Härdle, W. & Mammen, E. (1993). Testing parametric versus nonparametric regression, *Annals of Statistics* **21**: 1926–1947.
- Härdle, W., Mammen, E. & Müller, M. (1996). Testing parametric versus semiparametric modelling in generalized linear models, *SFB 373 Discussion Paper 960028*, Institut für Statistik und Ökonometrie, Humboldt-Universität zu Berlin.

- Härdle, W. & Turlach, B. A. (1992). Nonparametric approaches to generalized linear models, in L. Fahrmeier, B. Francis, R. Gilchrist & G. Tutz (eds), *Advances in GLIM and Statistical Modelling*, Vol. 78 of *Lecture Notes in Statistics*, Springer-Verlag, New York, pp. 213–225.
- Hastie, T. J. & Tibshirani, R. J. (1990). *Generalized Additive Models*, Vol. 43 of *Monographs on Statistics and Applied Probability*, Chapman and Hall, London.
- Horowitz, J. L. (1993). Semiparametric and nonparametric estimation of quantal response models, in G. S. Madala, C. R. Rao & H. D. Vinod (eds), *Handbook of Statistics*, Elsevier Science Publishers, pp. 45–72.
- Müller, M., Rönz, B. & Härdle, W. (1997). Computerassisted semiparametric generalized linear models, *Computational Statistics, Special Issue: 10 Years AG GLM* **12**: 153–172.
- Powell, J. L., Stock, J. H. & Stoker, T. M. (1989). Semiparametric estimation of index coefficients, *Econometrica* **57**(6): 1403–1430.
- Severini, T. A. & Staniswalis, J. G. (1994). Quasi-likelihood estimation in semiparametric models, *Journal of the American Statistical Association* **89**: 501–511.
- Severini, T. A. & 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.
- Weisberg, S. & Welsh, A. H. (1994). Adapting for the missing link, *Annals of Statistics* **22**: 1674–1700.
- XploRe (1997). *XploRe – the interactive statistical computing environment*, WWW: <http://wotan.wiwi.hu-berlin.de/xplore>.