A Local Estimation Approach in Multi-Categorical Varying-Coefficient Models

Silke Edlich
Technische Universität Berlin

March 23, 1998

Overview

1. Example
2. Varying-Coefficient Model
3. Model Fitting
4. Simulation
5. Estimation with S-Plus

Cervix Carcinom Data

We have $n\!=\!356$ observations of patients after a radical cervix carcinom operation (recorded at the University of Graz since 1971).

Primary goal: Investigation of interactions between a relapse and possible risk-factors.

- y:
  relapse (yes $=1$, no $=0$)
- LYM:
  lymph nodes infected (yes $=1$, no $=0$)
- SPR:
  spread of tumor over neighbourhood tissue (yes $=1$, no $=0$)
- AGE:
  age of the patient (years)

Generalized Linear Model

$$P(y\!=\!1|\,\mathrm{LYM{,}SPR{,}AGE}) = h(\eta) \quad \text{with}$$

$$\eta = \mathrm{LYM}*(\mathrm{AGE}+\mathrm{AGE}^2+\ldots) + \mathrm{SPR}*(\mathrm{AGE}+\mathrm{AGE}^2+\ldots)$$

Multi-Categorical Varying-Coefficient Model

$$\mu = E(y|x, u) = h(Z(x)\beta(u)) \quad \text{where}$$

- $y' = (y_1, \ldots, y_q)$ is the response vector,
- $Z(x)$ is a design matrix composed of the covariates $x' = (x_1, \ldots, x_p)$,
- $\beta(u) = \beta_u$ is the unknown parameter vector $\in \mathbb{R}^r$ which may vary with the effect modifiers $u \in \mathbb{R}^p$ and
- $h : \mathbb{R}^q \to \mathbb{R}^q$ is a common response function.

Example: Cervix Carzinom $(q = 1)$

$$P(y = 1|\underbrace{\mathrm{LYM}, \mathrm{SPR}}_{x}, \underbrace{\mathrm{AGE}}_{u})$$

$$= F(\beta_0(\mathrm{AGE}) + \underbrace{x_{\mathrm{LYM}}\beta_{\mathrm{LYM}}(\mathrm{AGE})}_{\downarrow} + \underbrace{x_{\mathrm{SPR}}\beta_{\mathrm{SPR}}(\mathrm{AGE})}_{\downarrow})$$

$$\begin{array}{ll} x_{\mathrm{LYM}}x_{\mathrm{AGE}}\beta_{\mathrm{LYM}\,*\,\mathrm{AGE}} & x_{\mathrm{SPR}}x_{\mathrm{AGE}}\beta_{\mathrm{SPR}\,*\,\mathrm{AGE}} \\ +x_{\mathrm{LYM}}x^2_{\mathrm{AGE}}\beta_{\mathrm{LYM}\,*\,\mathrm{AGE}^2} & +x_{\mathrm{SPR}}x^2_{\mathrm{AGE}}\beta_{\mathrm{SPR}\,*\,\mathrm{AGE}^2} \end{array}$$

Varying-Coefficients as General Model

We consider the simpler case of univariate y

$$E(y|x, u) = h(\beta_0(u_0) + x_1\beta_1(u_1) + \ldots + x_p\beta_p(u_p)).$$

Thus the model includes following special cases if

- $u_0 \equiv \ldots \equiv u_p \equiv 1$
  Generalized linear model,
  (McCullagh & Nelder, 1989)

- $u_1 \equiv \ldots \equiv u_p \equiv 1$
  semiparametric model,
  (Hunsberger, 1994)

- $x_1 \equiv \ldots \equiv x_p \equiv 1$
  Generalized additive model,
  (Hastie & Tibshirani, 1990)

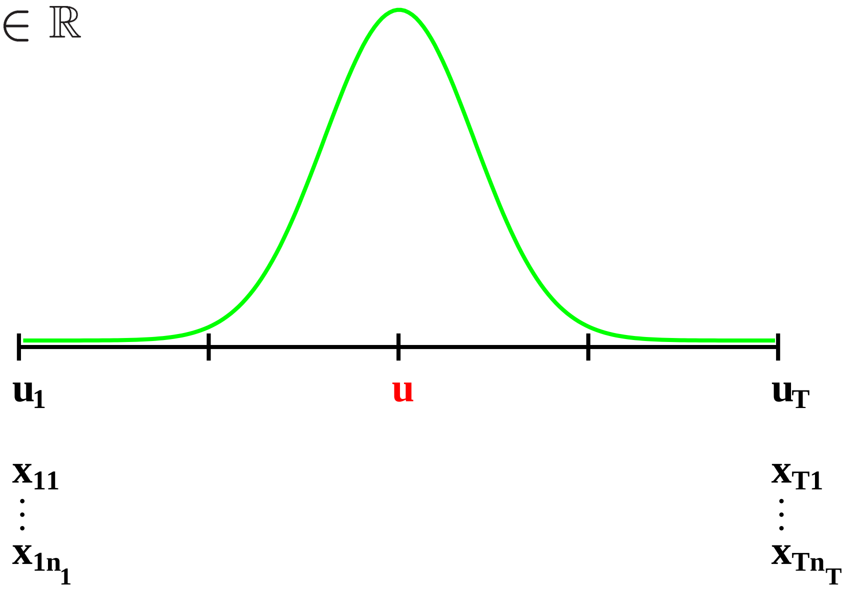
- $u \equiv u_0 \equiv u_1 \equiv \ldots \equiv u_p$
  model with one modifying variable.
  (Fan & Gijbels, 1996)

Local Likelihood Estimation

Model: $\mu_i = h(\eta_i)$, Data: (y_i, x_i, u_i), $i = 1, \ldots, N$

$$\eta_i = Z_{i0}\beta_0(u_i) + Z_{i1}\beta_1(u_i) + \ldots + Z_{ip}\beta_p(u_i)$$

Idea: $u \in \mathbb{R}$



Definition of the weights:

$$w_\gamma(u, u_i) = K\left(\frac{u - u_i}{\gamma}\right) / K(0),$$

where K is an unimodal, symmetric kernel function with the smoothing parameter $\gamma > 0$.

Local Log-Likelihood-Function:

$$l_\gamma(\beta_u) \;=\; \sum_{i=1}^{N} w_\gamma(u, u_i)\{y_i'\theta_i(\beta_u) - b(\theta_i(\beta_u))\}/\phi$$

$\theta_i(\beta_u)$	natural parameter
$b(\theta_i(\beta_u))$	cumulant function
ϕ	dispersion parameter

Local Score-Function:

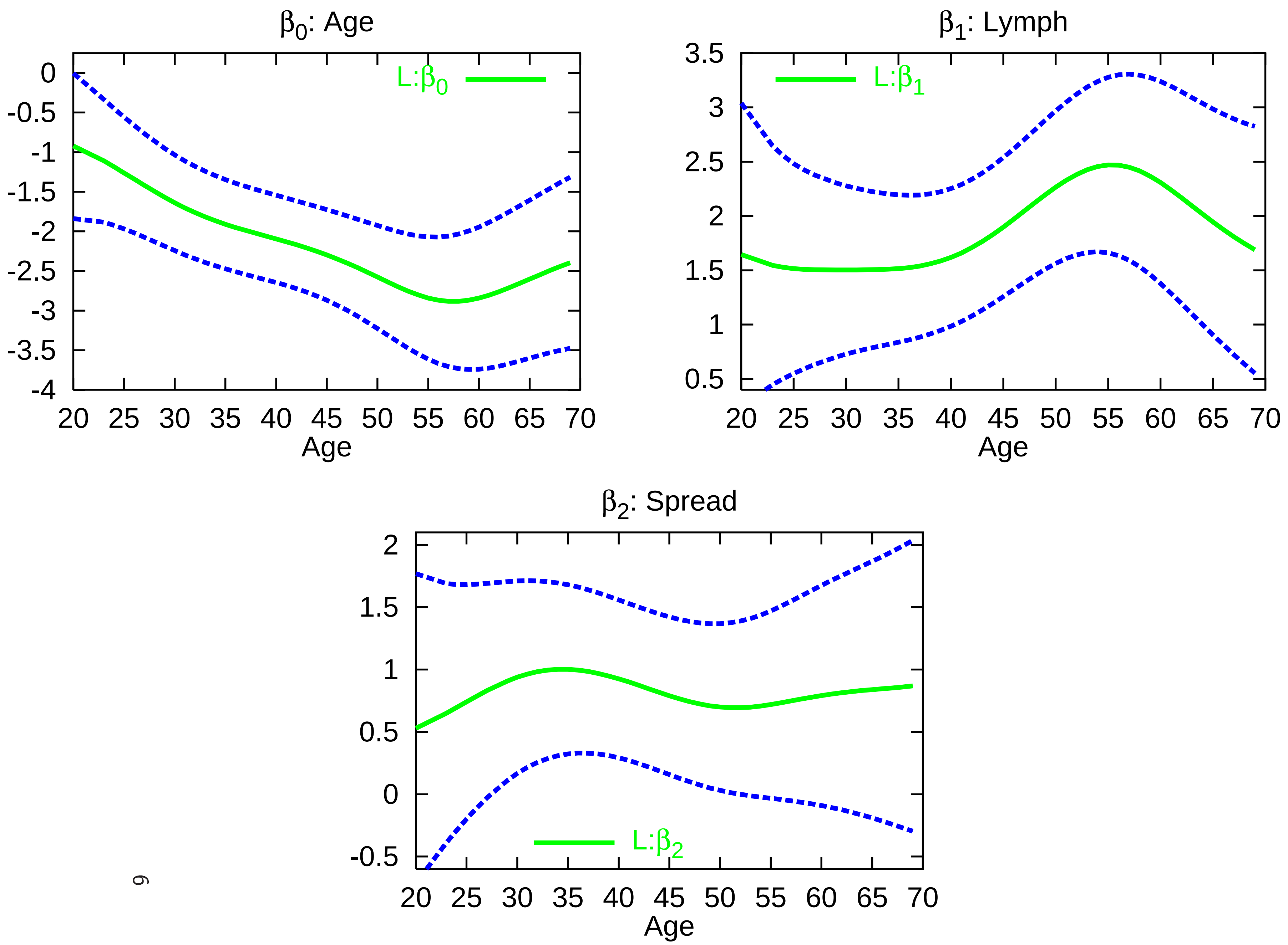
$$s_\gamma(\beta_u) \;=\; \sum_{i=1}^{N} w_\gamma(u, u_i) Z_i' V_i(\beta_u)\{y_i - h(\eta_i)\} \stackrel{!}{=} 0$$

with $\eta_i = Z_i\beta_u$ and $V_i(\beta_u) = \dfrac{\partial h(\eta_i)}{\partial \eta}\Sigma_i^{-1}(\beta_u)$

Iterative Fisher-Scoring-Algorithm yields Numerical Solution:

$$\hat\beta_u^{k+1} = \hat\beta_u^{k} + F_\gamma^{-1}(\hat\beta_u^{k}) s_\gamma(\hat\beta_u^{k}) \qquad \text{with}$$

$$F_\gamma(\beta_u) = \sum_{i=1}^{N} w_\gamma(u, u_i) Z_i' V_i(\beta_u)\Sigma_i(\beta_u)V_i'(\beta_u) Z_i$$



β0: Age
L:β0
Age
β1: Lymph
L:β1
Age
β2: Spread
L:β2
Age

Local Likelihood-Estimation with Backfitting

Model: $\mu_i = h(\eta_i), \quad i = 1, \ldots, N$ with

$$\eta_i = \underbrace{Z_{i0}\beta_0(u_{i0})}_{\gamma_0} + \ldots + \underbrace{Z_{ir}\beta_r(u_{ir})}_{\gamma_r} + \ldots + \underbrace{Z_{ip}\beta_p(u_{ip})}_{\gamma_p}$$

Algorithm:

1. Initiation

   Compute the estimates by fitting the fixed effect model

   $$\mu_i = h(Z_{i0}\beta_0 + \ldots + Z_{ip}\beta_p)$$

   and start with $\hat{\beta}^{(0)}(u) = (\hat{\beta}_0, \ldots, \hat{\beta}_p)'$, $k = 0$

2. Loop

   Compute $r = 0, \ldots, p$ estimates $\hat{\beta}_r^{(k+1)}(u_r)$ for the unknown parameters. The local equation for the rth component is

   $$\sum_{i=1}^{N} w_{\gamma_r}(u_{0r}, u_{ir}) Z'_{ir} V_i(\eta_{ir})(y_i - h(\eta_{ir})) \stackrel{!}{=} 0$$

with the predictor

$$\hat{\eta}_{ir}^{(k+1)} = \underbrace{Z_{i0}\hat{\beta}_0^{(k+1)}(u_{i0}) + \dots}_{\text{fixed}} + Z_{ir}\hat{\beta}_r{}^{(k+1)}(u_{ir}) + \underbrace{\dots + Z_{ip}\hat{\beta}_p^{(k)}(u_{ip})}_{\text{fixed}}$$

3. Continue (2) until all parameters converge.

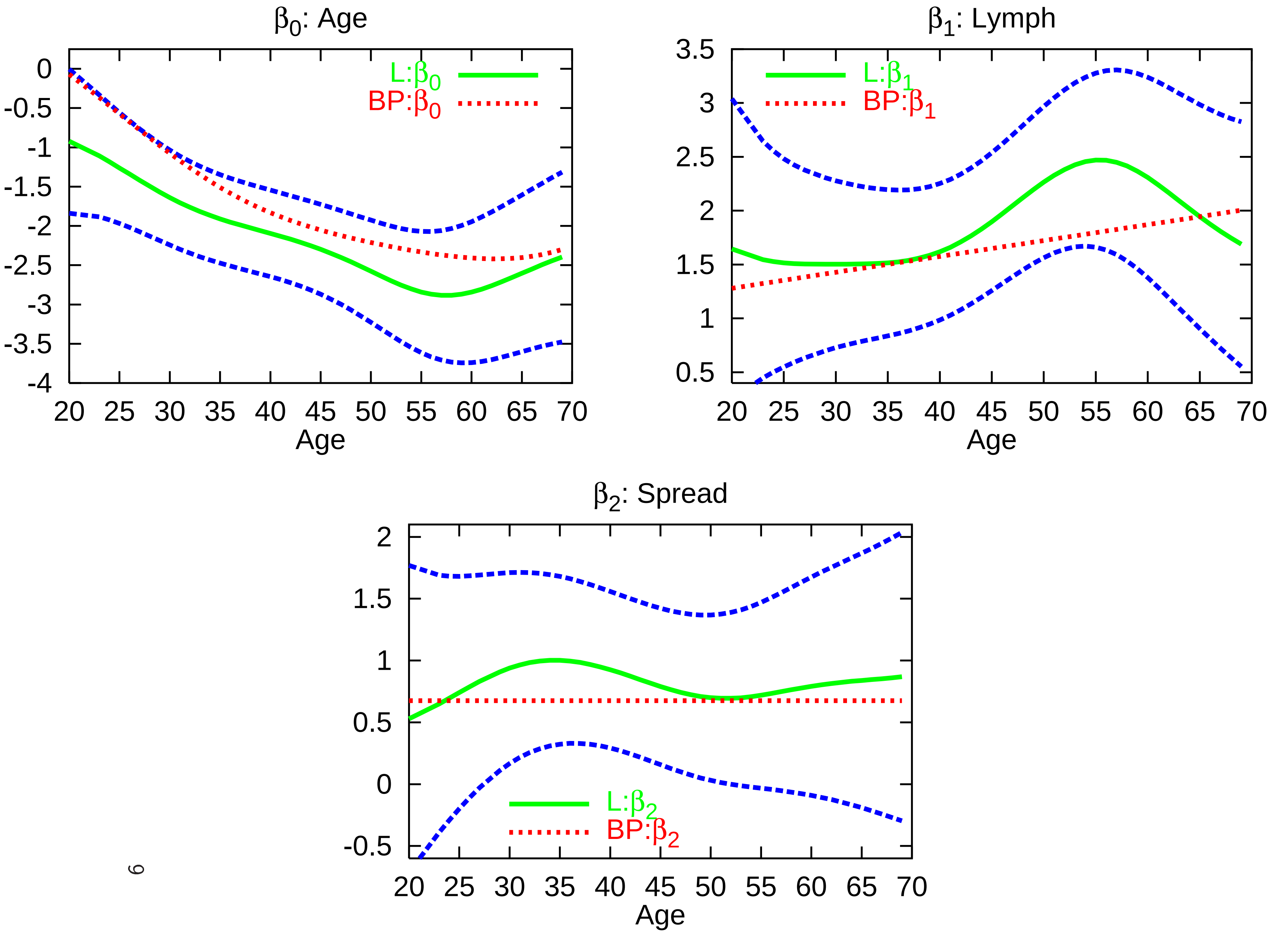
Local Polynomial Fitting

$\beta_r(u)$ smooth: Taylor approximation

$$\beta_r(u) \approx \sum_{j=0}^{g_r} \beta_{rj}^*(u-u^*)^j \text{ with } \beta_{rj}^* = \frac{\beta_r^{(j)}(u^*)}{j!}$$

Estimate β_r^* at target point $u = u^*$ with $g_r = 2$:

$$\tilde{\eta}(Z_{ir}, u^*) = \underbrace{\dots}_{\text{fixed}} + Z_{ir} * \left[\beta_{r0}^* + \beta_{r1}^*(u_i - u^*)^1 + \beta_{r2}^*(u_i - u^*)^2\right] + \underbrace{\dots}_{\text{fixed}}$$



β0: Age
L:β0
BP:β0
Age
β1: Lymph
L:β1
BP:β1
Age
β2: Spread
L:β2
BP:β2
Age

Simulation Studies

Model: $y_{ti} \sim B(n_{ti}, \pi_{ti})$ with $t = 0, \dots, 20, \quad i = 1, 2$

$$\pi_{ti} = \frac{\exp(\beta_0(t) + x_i\beta_1(t))}{1 + \exp(\beta_0(t) + x_i\beta_1(t))}, \quad x_i \in \{0, 1\}$$

For each $n_{ti}=2$, $n_{ti}=5$, $n_{ti}=10$ and $n_{ti}=20$ we simulate 100 data sets.

1. Simulation:

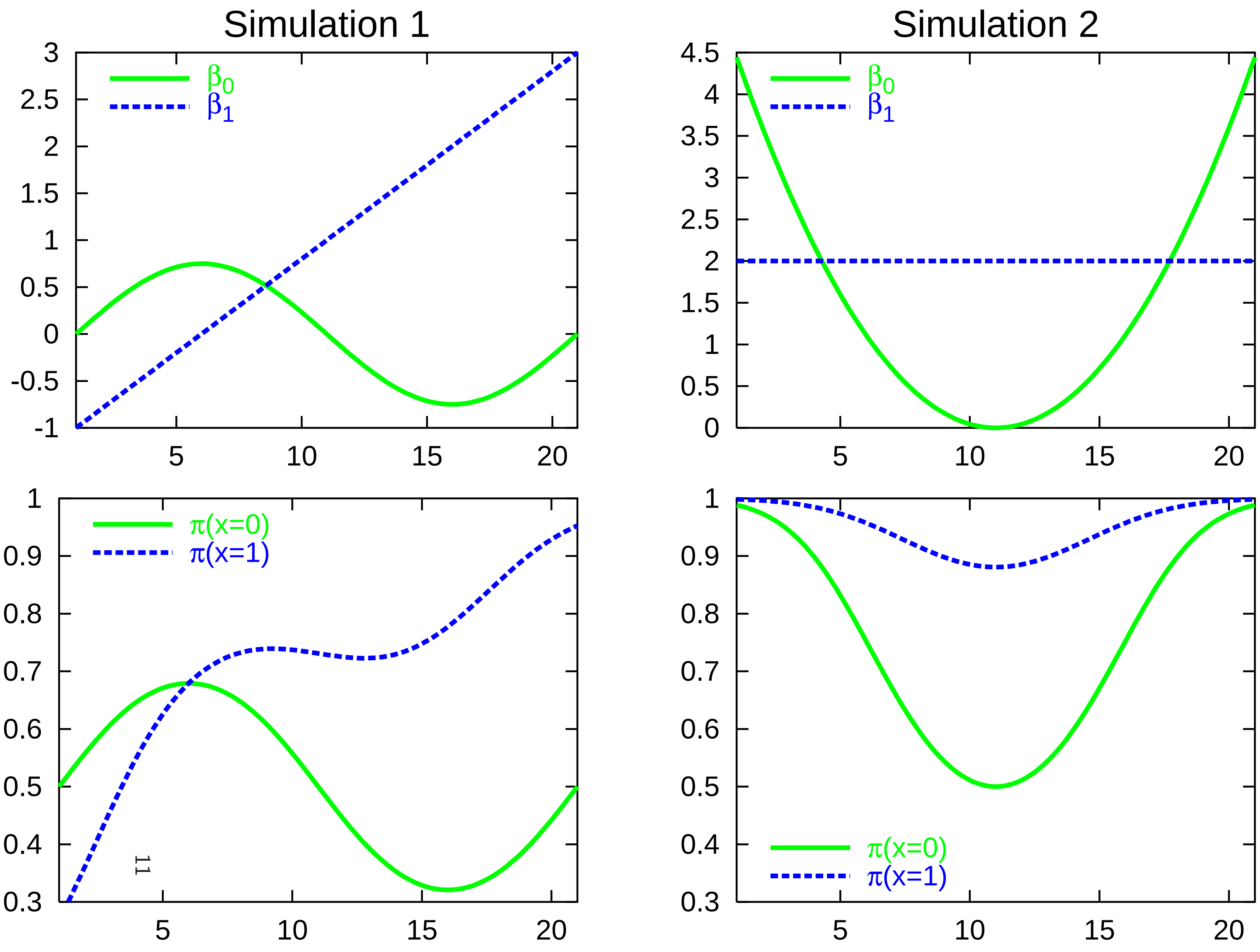
$$\beta_0(t) = 0.75 \sin(2\pi * t/20)$$
$$\beta_1(t) = -1 + 0.2 * t$$

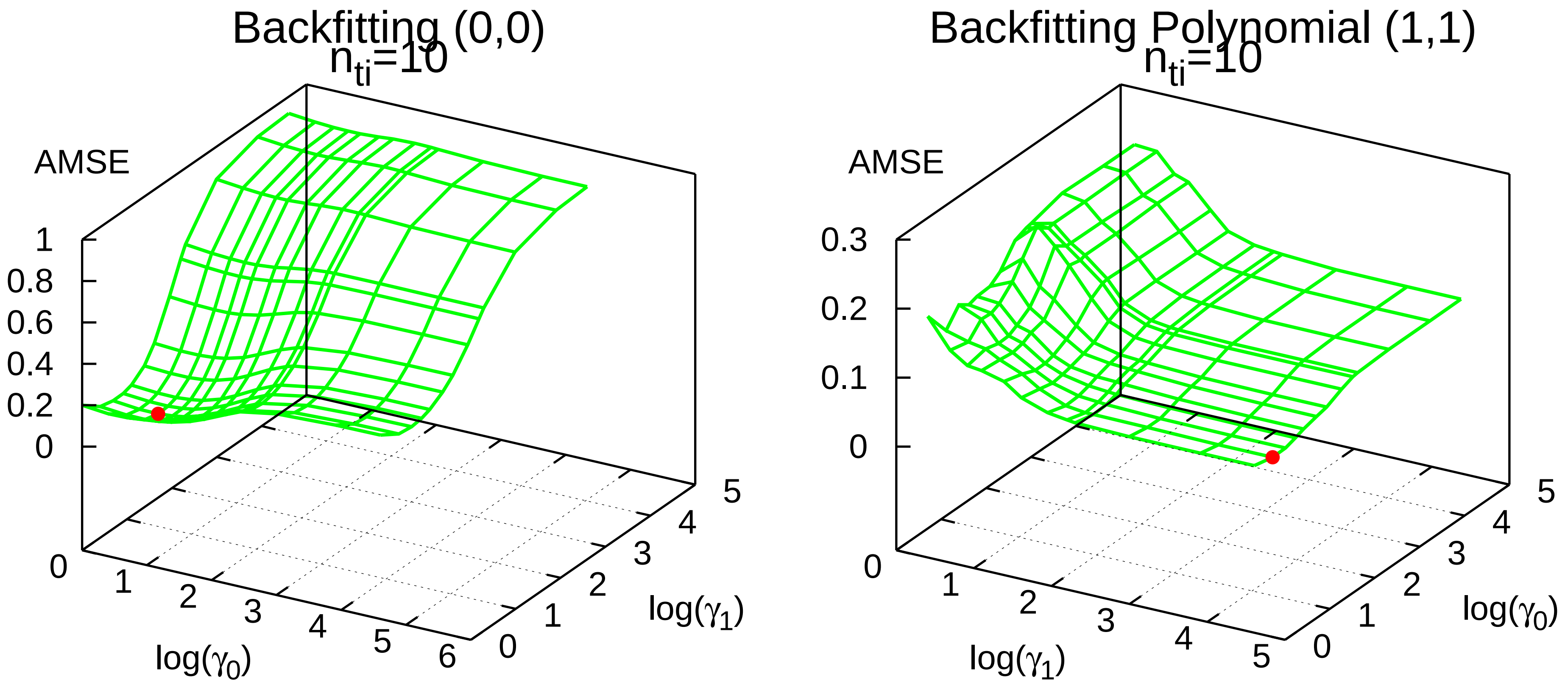
2. Simulation:

$$\beta_0(t) = 0.4((t-10)/3)^2$$
$$\beta_1(t) = 2$$

Comparison between the Estimators

$$\mathrm{AMSE}(\pi) = \frac{1}{100}\sum_{s=1}^{100} \frac{1}{42 n_{ti}} \sum_{t=0}^{20}\sum_{i=1}^{2}\sum_{j=1}^{2}(\pi_{tij} - \hat{\pi}_{tij})^2$$


Simulation 1
Simulation 2
β_0
β_1
$\pi(x=0)$
$\pi(x=1)$


Backfitting (0,0)
n_ti=10
AMSE
1
0.8
0.6
0.4
0.2
0
0
1
2
3
4
5
6
log(γ0)
0
1
2
3
4
5
log(γ1)
Backfitting Polynomial (1,1)
n_ti=10
AMSE
0.3
0.2
0.1
0
0
1
2
3
4
5
log(γ1)
0
1
2
3
4
5
log(γ0)

Results of the 1. Simulation

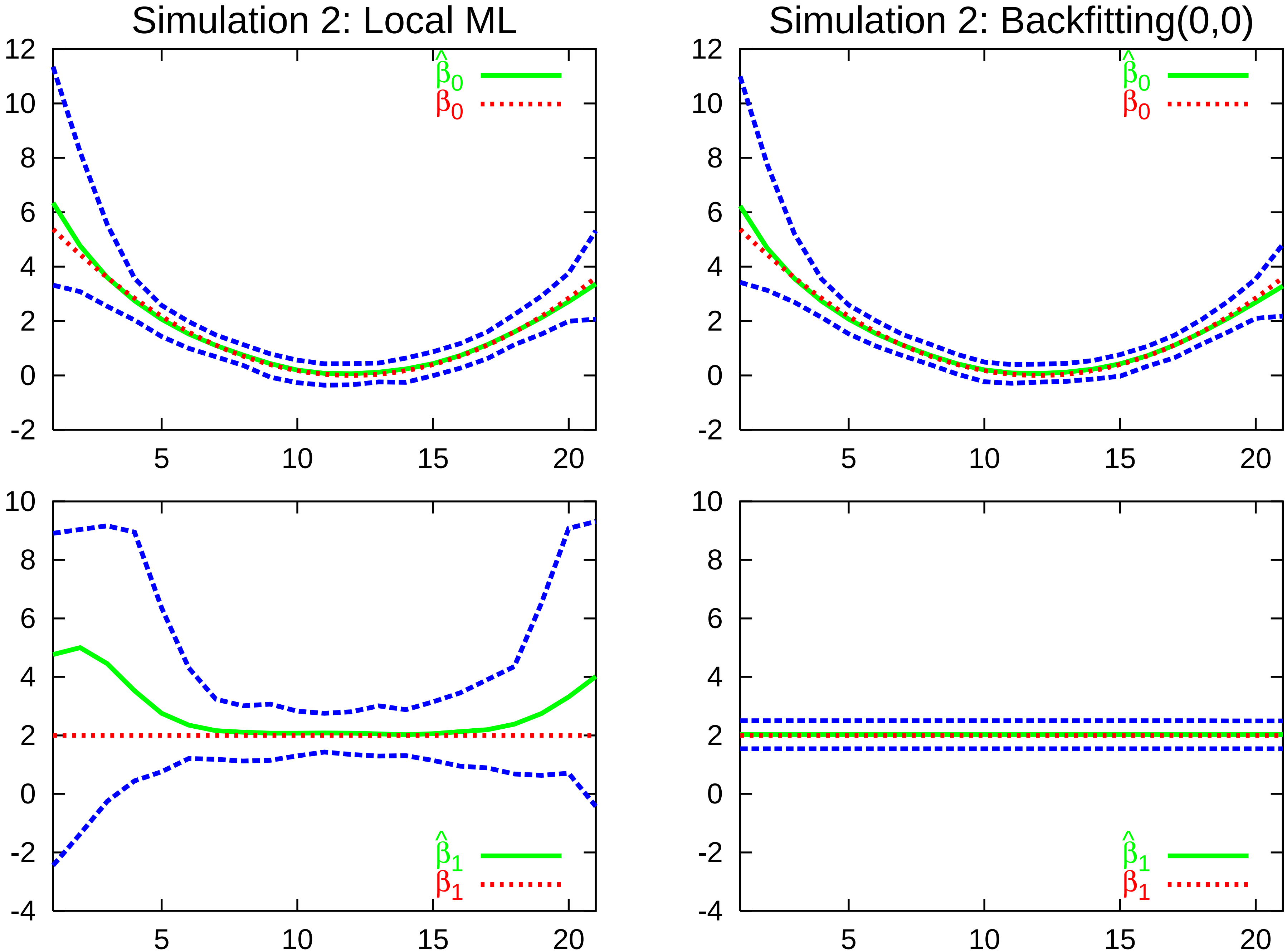
Minimal $\mathrm{AMSE}(\pi, \gamma)$-values				
n_{ti}	LML	LMLPoly(1,1)	Back(0,0)	BackPoly(1,1)
$n_{ti} = 2$	0.0275551	0.0301419	0.0263286	0.0231029
$n_{ti} = 5$	0.0140908	0.0152139	0.0129157	0.0125222
$n_{ti} = 10$	0.0082204	0.0083472	0.0073922	0.0068916

Smoothing parameter γ_0, γ_1 corresponding to $\mathrm{AMSE}(\pi, \gamma)$-value				
n_{ti}	LML	LMLPoly(1,1)	Back(0,0)	BackPoly(1,1)
$n_{ti} = 2$	4.0, 4.0	7.0, 7.0	3.0, 3.0	3.0, 50.0
$n_{ti} = 5$	2.5, 2.5	4.0, 4.0	2.5, 2.5	2.5, 100.0
$n_{ti} = 10$	2.0, 2.0	2.5, 2.5	2.0, 2.0	2.0, 100.0

Results of the 2. Simulation

Minimal $\mathrm{AMSE}(\pi, \gamma)$-values					
n_{ti}	LML	BackPoly(0,0)	BackPoly(1,1)	BackPoly(2,2)	BackPoly(2,0)
$n_{ti} = 20$	0.0161787	0.00976659	0.011182	0.00741261	0.00468223

Smoothing parameter γ_0, γ_1 corresponding to $\mathrm{AMSE}(\pi, \gamma)$-value					
n_{ti}	LML	BackPoly(0,0)	BackPoly(1,1)	BackPoly(2,2)	BackPoly(2,0)
$n_{ti} = 20$	1.0, 1.0	1.0, 50.0	2.0, 1.0	100.0, 4.0	10.0, 20.0


Simulation 2: Local ML
Simulation 2: Backfitting(0,0)
$\hat{\beta}_0$
β_0
$\hat{\beta}_1$
β_1
12
10
8
6
4
2
0
-2
-4
5
10
15
20

Data Situation: Cervix

y1	y2	LYM	SPR	AGE
0	1	0	0	20
0	1	0	0	23
1	0	0	0	24
1	0	1	0	24
1	0	0	0	25
.				
.				
.				
0	2	0	0	67
1	0	1	1	67
0	1	0	0	68
1	0	0	1	68
0	1	0	1	69

$\underbrace{\text{y1 y2}}_{y} \quad \underbrace{\text{LYM SPR}}_{x} \quad \underbrace{\text{AGE}}_{u}$

Example estimated with S-Plus

Calling for the function vcm

```
erg <- vcm( x, y, n, u, link="logit",
          error="binomial", kernel="normal",
          bandwidth=20, degree=c(1,1,0))
```

Important Options:

- `link:`
  `"logit", "cummulative.logit", ...`
- `u.points:`
  grid points used for estimation
- `degree:`
  degree of polynom
- `offset:`
  "Backfitting"
- `resid:`
  residuals
- `weight.matrix:`
  define your own weight matrix

Detailed explanation of the options in S-Plus: `help("vcm")`

Output

Output: estimated coefficients

erg$coef

```
      Intercept1        X1        X2    I(u1)^1  X1(u1)^1

  [1,]  -0.401114  0.855681  0.900106  -0.079283  0.044414
  [2,]  -0.659916  0.959836  0.892117  -0.077566  0.045517
  [3,]  -0.744716  0.997622  0.889355  -0.076943  0.045777
   .
   .
   .
[163,]  -2.902211  2.183661  0.801697  -0.028444  0.012665
[164,]  -2.902211  2.183661  0.801697  -0.028444  0.012665
[165,]  -2.903133  2.168310  0.801841  -0.027007  0.011205
```

Output

Output: estimated variance

erg$var

```
, , 1
               [,1]          [,2]           [,3]          [,4]           [,5]
[1,]  0.34087822 -0.3341507049 -0.02459538414 -0.0145728792  0.01482898661
[2,] -0.33415070  0.7091492792  0.00020425379  0.0148012891 -0.02800390929
[3,] -0.02459538  0.0002042538  0.09626891432 -0.0010206474 -0.00006904577
[4,] -0.01457288  0.0148012891 -0.00102064740  0.0007727832 -0.00075987316
[5,]  0.01482899 -0.0280039093 -0.00006904577 -0.0007598732  0.00127293667

, , 2
 .
 .
 .
```

Plotting the result

```
diag.var <- matrix(0, nrow(erg$coef), ncol(erg$coef))
for(i in 1:nrow(erg$coef))
   diag.var[i,] <- diag(1.96*sqrt(erg$var[,,i]))

lower <- cbind(erg$coef - diag.var)
upper <- cbind(erg$coef + diag.var)
pidx <- order(u)


par(mfrow=c(2,ceiling(ncol(erg$coef)/2)-1))
title <- c("Intercept", "SPR", "GRZ")
for(i in 1:(ncol(erg$coef)-2))
{
  plot(u[pidx,], erg$coef[pidx,i], type="n" ,
  ylim=c(min(lower[,i]),max(upper[,i])), xlab="age",
  ylab="", main=title[i])
  lines(u[pidx,], erg$coef[pidx,i], type="l", col=8,
  lwd=5)
  lines(u[pidx,], lower[pidx,i], type="l", col=2, lwd=5)
  lines(u[pidx,], upper[pidx,i], type="l", col=2, lwd=5)
}
```

Datasituation: Preference for voting a party

CDU	SPD	AGE
2	6	18
3	4	19
3	9	20
1	7	21
2	13	22
4	16	23
.	.	.
.	.	.
.	.	.
y		u

Reading data

```
pfem <- read.table("party.dat", header=TRUE)
x <- NULL
u <- as.matrix(pfem[,3])
y <- as.matrix(pfem[,1])
n <- as.matrix(pfem[,1] + pfem[,2])
```

Functioncalls

```
erg3 <- vcm( x, y, n, u, link="logit", error="binomial",
          kernel="normal", bandwidth=3, resid="pearson")

erg8 <- vcm( x, y, n, u, link="logit", error="binomial",
          kernel="normal",bandwidth=8, resid="pearson")

erg1000 <- vcm( x, y, n, u, link="logit", error="binomial",
          kernel="normal",bandwidth=1000, resid="pearson")
```

Output

erg3$coef

```
      Intercept1
 [1,]  -1.18576872
 [2,]  -1.25227491
 [3,]  -1.31323211
  .
  .
  .
[73,]  -0.29974033
[74,]  -0.27002903
[75,]  -0.78510044
```

erg3$var

```
, , 1

           [,1]
[1,] 0.07922771

, , 2
           [,1]
[1,] 0.06552503

, , 3
.
.
.
```

Plotting the result

```
par(mfrow=c(2,2))
plot(c(20, 90),c(-4,4), type="n", ylab="", main="Residualplot (1000)")
lines(u[1:73], erg1000&resid[1:73,1], type="p", col=3, lwd=5)
abline(0,0)
plot(c(20, 90),c(-4,4), type="n", ylab="", main="Residualplot (8)")
lines(u[1:73], erg8&resid[1:73,1], type="p", col=3, lwd=5)
abline(0,0)
plot(c(20, 90),c(-4,4), type="n", ylab="", main="Residualplot (3)")
lines(u[1:73], erg3&resid[1:73,1], type="p", col=3, lwd=5)
abline(0,0)
plot(c(20, 90),c(-2,1), type="n", ylab="", main="Preference for voting a
party (3)")
lines(u[1:73], erg3&coef[1:73,1], type="l", col=3, lwd=5)
lines(u[1:73], erg3&coef[1:73,1] - (1.96*sqrt(erg3&var[,,1])), type="l")
lines(u[1:73], erg3&coef[1:73,1] + (1.96*sqrt(erg3&var[,,1])), type="l")
```

Option: u.points

```
u12 <- cbind(c(20, 20.5, 21))


ergup <- vcm( x, y, n, u, kernel="normal",
       error="binomial", link="logit",
       u.points=u12, bandwidth = 3 )


print(cbind(erg3$u[1:5,1], erg3$coef[1:5,1]))

         [,1]        [,2]
  [1,]     18   -1.185769
  [2,]     19   -1.252275
  [3,]     20   -1.313232
  [4,]     21   -1.360463
  [5,]     22   -1.387907

print(cbind(u12, ergup$coef))

                Intercept1
 [1,]   20.0     -1.313232
 [2,]   20.5     -1.339021
 [3,]   21.0     -1.360463
```

Option: weight.matrix

```
ergall <- vcm( x, y, n, error="binomial",
       link="logit", all.obs=T )

u12 <- cbind(c(20, 20.5, 21))
weights <- matrix(rep(1,nrow(y)*lenghts(u12)),nrow=3)
ergg <- vcm( x, y, n, u, error="binomial",
       link="logit", u.points=u12,
       weight.matrix=weights)
```

```
ergall$coef

        Intercept1
  [1,]  -0.2904871

ergg$coef

       Intercept1
 [1,]  -0.2904871
 [2,]  -0.2904871
 [3,]  -0.2904871
```

Integration into S-Plus

- Implemented in `C++`
- `C`-Interface to S-Plus
- S-Plus Version: 2.1 or higher
- Available for Solaris and Windows 95 / NT 4.0
- `http://stat.cs.tu-berlin.de/~monster`
- Questions: `Silke.Edlich@cs.tu-berlin.de`

Choice of Smoothing Parameter

Intention: Minimizing the Mean Squared Error

Hastie & Tibshirani (1990) suggest a leaving one out cross validation.

Cross validated likelihood function:

$$\text{cvl} = \sum_i l_i(\hat{\beta}^{(-i)}(u))$$

where $\hat{\beta}^{(-i)}(u)$ is estimated by neglecting all observations of the ith individual.

It can be shown:

maximizing cvl(γ) $\Leftrightarrow$ minimizing the M.S.E.

Backfitting in GAMs

Convergence and uniqueness

Buja, Hastie & Tibshirani, 1989:

$$\|S_1 S_2\| < 1 \qquad S_i \mathrel{\hat{=}} \text{smoother matrix}$$

Restrictions: bivariate, symmetric projection smoothers

Opsomer & Rupert 1997 (bivariate), Opsomer 1995 (general):

$$\max_{i=2,\ldots,p} \left\| \sum_{j=1}^{p-1} S_i S_j \right\| < 1$$

- general smoothers
- not dependent on the order of fitting
- pairwise condition, more strict with increasing p

Backfitting in GAMs

Local polynomial smoothers

$$\sup_{x_i, x_j} \left| \frac{f_{ij}(x_i, x_j)}{f_i(x_i) f_j(x_j)} - 1 \right| < \frac{1}{p-1} \rightarrow \text{Existence of direct solution}$$

with f marginal (univariate,bivariate) density

- bivariate dependence between covariates
- condition in the data-generating process, sufficent