Generalized Partially Linear Models

Roberto G. Gutierrez

Stata Corporation

OUTLINE

I. Introduction

  A. Basic concepts behind glm

  B. Iterative reweighted least squares

II. The partially linear model

  A. Definition

  B. New algorithm

III. Example

  A. Kyphosis data

  B. Computational issues

IV. Further work

INTRODUCTION

Basic Concepts of GLM

Canonical exponential family

$$f(y|\theta,\phi) = \exp\left\{\frac{y\theta - b(\theta)}{a(\phi)} + c(y,\phi)\right\}$$

where

$$\begin{aligned}\theta &= \text{canonical parameter} \\ &= \textit{somefunction}(\mathbf{x}\boldsymbol{\beta}) \\ \phi &= \text{scale parameter}\end{aligned}$$

Log–likelihood

$$\mathcal{L}(\theta,\phi|\mathbf{y}) = \sum_{i=1}^{n}\left\{\frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + c(y_i,\phi)\right\}$$

For $\eta = \mathbf{x}\boldsymbol{\beta}$ = linear predictor

$$\mu \equiv E(y) = b'(\theta) \equiv g^{-1}(\eta)$$
$$\text{Var}(y) = b''(\theta)a(\phi) \equiv V(\mu)a(\phi)$$

and thus $g(\mu) = \mathbf{x}\boldsymbol{\beta}$ is called a link function since it links $E(y)$ to the linear predictor.

MLE estimate of $\boldsymbol{\beta}$ solves

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\beta}} = \sum_{i=1}^{n} \frac{y_i - \mu_i}{a(\phi)V(\mu_i)} \left(\frac{\partial \mu}{\partial \eta}\right) \mathbf{x}_i^t = 0$$

Newton-Raphson solution (i.e. using the observed Hessian) requires evaluation of

$$\frac{\partial \mu}{\partial \eta}, \qquad \frac{\partial^2 \mu}{\partial \eta}, \qquad \frac{\partial V(\mu)}{\partial \mu}$$

Fisher scoring (which uses the expected Hessian) requires only the evaluation of $\partial\mu/\partial\eta$ (or $\partial\eta/\partial\mu$), and fits into the algorithm of iterated reweighted least–squares (IRLS).

Can be generalized into maximum quasi–likelihood estimation for which only the components μ and $V(\mu)$ need to be specified.

Examples

Poisson model

$$\begin{aligned} f(y|\mu) &= \exp\{y\ln(\mu) - \mu - \ln\Gamma(y+1)\} \\ \theta &= \ln(\mu) \\ b(\theta) &= \exp(\theta) = \mu \\ b''(\theta) &= \mu = V(\mu) \\ a(\phi) &= 1 \end{aligned}$$

Canonical link is

$$g(\mu) = \ln(\mu) = \eta = \mathbf{x}\boldsymbol{\beta}$$

Gamma model

$$\begin{aligned} f(y|\mu,\phi) &= \exp\left\{\frac{y/\mu + \ln(\mu)}{-\phi} + \frac{\phi+1}{\phi}\ln(y) - \frac{\ln(\phi)}{\phi} - \ln\Gamma(\phi^{-1})\right\} \\ \theta &= 1/\mu \\ b(\theta) &= \ln(\theta) = -\ln(\mu) \\ b''(\theta) &= -1/\theta^2 = -\mu^2 \\ a(\phi) &= -\phi \end{aligned}$$

Thus, we can take $V(\mu) = \mu^2$ and the canonical link is the reciprocal link

$$g(\mu) = 1/\mu = \eta = \mathbf{x}\boldsymbol{\beta}$$

Iterated reweighted least squares

1. Construct an adjusted dependent variable (pseudo–response)

$$z_i = \eta_i + (y_i - \mu_i)\left(\frac{\partial \eta_i}{\partial \mu_i}\right)$$

2. Construct weights

$$w_i = \left(\frac{\partial \mu_i}{\partial \eta_i}\right)^2 \{V(\mu_i)\}^{-1}$$

3. Perform a weighted linear regression of z_i on $\mathbf{x}_i$ and calculate new $\eta_i = \mathbf{x}_i\boldsymbol{\beta}$ and $\mu_i = g^{-1}(\eta_i)$.

4. Iterate

The point is that GLM/IRLS can be done using `regress` with `iweights`.

The Partially Linear Model

Rather than the standard linear predictor

$$\eta_i = \mathbf{x}_i \boldsymbol{\beta}$$

the partially linear model allows

$$\eta_i = f(x_i) + \mathbf{v}_i \boldsymbol{\beta}$$

that is, one predictor allowed to be nonlinear.

Stata command `fracpoly` will treat $f(x_i)$ as a fractional polynomial.

My approach is nonparametric, and instead uses a local-linear smooth to estimate $f(x_i)$.

That is, step 3. in the previous is replaced by a weighted partially linear (gaussian errors) model.

New Algorithm

Old Step 3: Perform a weighted linear regression of z_i on $\mathbf{x}_i$ and calculate new $\eta_i = \mathbf{x}_i\boldsymbol{\beta}$ and $\mu_i = g^{-1}(\eta_i)$.

New Step 3:

3a. Perform a weighted linear regression of z_i on x_i and $\mathbf{v}_i$.

3b. Form residuals $e_i = z_i - \mathbf{v}_i\widehat{\boldsymbol{\beta}}$.

3c. Perform a weighted local linear smooth of e_i on x_i. This can be done using `locpoly` with `iweights`.

3d. Form residuals $e_i^* = z_i - \widehat{f}(x_i)$. Regress (with weights) e_i^* on $\mathbf{v}_i$.

3e. Iterate 3b. – 3d. until convergence.

3f. Form $\eta_i = \widehat{f}(x_i) + \mathbf{v}_i\widehat{\boldsymbol{\beta}}$ and $\mu_i = g^{-1}(\eta_i)$.

The above algorithm is known as backfitting and is done using my `partlin` command.

Example

Consider the data used by Bell et al. (1989). Data on 83 children who undergo corrective spine surgery.

Response is the presence of **kyphosis**, a forward flexion of the spine.

Covariates are **age** (months), the starting vertebrae level of the surgery (**startvert**), and the number of levels involved in the surgery (**numvert**).

```
. list in 1/10

     +-------------------------------------+
     | age   startv~t   numvert   kyphosis |
     |-------------------------------------|
  1. |  71          5         3          0 |
  2. | 158         14         3          0 |
  3. | 128          5         4          1 |
  4. |   2          1         5          0 |
  5. |   1         15         4          0 |
     |-------------------------------------|
  6. |   1         16         2          0 |
  7. |  61         17         2          0 |
  8. |  37         16         3          0 |
  9. | 113         16         2          0 |
 10. |  59         12         6          1 |
     +-------------------------------------+
```

Fitting the binomial partially linear model.

```
. gplm kyphosis age startvert numvert, fam(bin) reps(200) nolog

Generalized partially linear models              No. of obs      =        83
Deviance         =  63.993901

Variance function: V(u) = u*(1-u)                [Bernoulli]
Link function    : g(u) = ln(u/(1-u))            [Logit]

------------------------------------------------------------------------------
    kyphosis |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
   startvert | -.1128096   .1084214    -1.04   0.298    -.3253116    .0996924
     numvert | 1.1742779   .3547490     3.31   0.001     .4789826   1.8695732
------------------------------------------------------------------------------
Deviance test of f(age) = 0:       chi2(3.58) =      2.14 Prob > chi2 = 0.6462
```

Compared to fitting a standard GLIM:

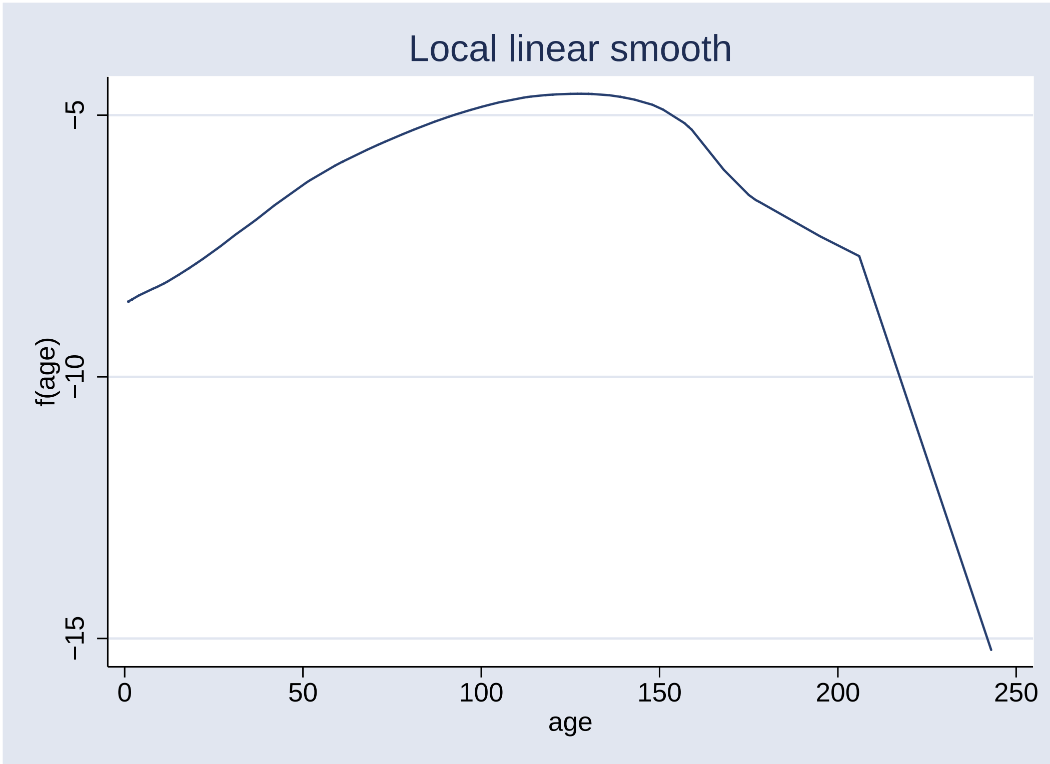
```
. glm kyphosis age startvert numvert , fam(binom) link(logit) nolog

Generalized linear models                        No. of obs      =        83
Optimization     : ML: Newton-Raphson            Residual df     =        79
                                                 Scale parameter =         1
Deviance         =  64.87296734                  (1/df) Deviance =  .8211768
Pearson          =  68.36550054                  (1/df) Pearson  =  .8653861

Variance function: V(u) = u*(1-u)                [Bernoulli]
Link function    : g(u) = ln(u/(1-u))            [Logit]
Standard errors  : OIM

Log likelihood   = -32.43648367                  AIC             =  .8779876
BIC              = -284.2154407

------------------------------------------------------------------------------
    kyphosis |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         age |    .006094   .0055402     1.10   0.271    -.0047645    .0169525
   startvert | -.1972165   .0657152    -3.00   0.003     -.326016   -.0684171
     numvert |   .3031238   .1789986     1.69   0.090    -.0477071    .6539546
       _cons | -1.249726   1.242394    -1.01   0.314    -3.684773    1.185321
------------------------------------------------------------------------------
```


Local linear smooth
f(age)
−5
−10
−15
0
50
100
150
200
250
age

Computational Issues

Bandwidth Selection: Given the iterated local linear smooth, automated bandwidth selection is essential. I use the Rule-of-Thumb method by Sheather and Wand (1995 JASA). There is the issue of proper order, however.

Standard errors: bootstrap, although it is possible to use matrix calculations and the smoothing matrix to get standard errors of the linear predictor.

Degrees of freedom for the smooth: can be calculated as the trace of the smoothing matrix, since kernel smoothing is a linear operation.

The above, especially bootstrapping, makes current implementation of `gplm` rather slow, but plugins have made things doable, at least.

`partlin`, bandwidth selection, and degrees of freedom calculation are all implemented as plugins.

Further Work

Faster standard errors (directly calculated).

Error bars for plot of the estimated smooth.

Generalization to single index models

Generalization to generalized additive model, where the user can specify either kernel methods or splines.