Profiling the parameters of models with linear predictors
The profileModel R package

Ioannis Kosmidis

Research Fellow
Department of Statistics

useR! 2008, Dortmund
Outline

1. Why develop a package for profiling?
2. The profileModel R package
3. Examples
4. More on profileModel
A variety of estimation methods

- **Deviations from maximum likelihood:**
  - Firth (1993) for penalized likelihoods and adjusted scores.
  - Lindsay (1988) for composite likelihoods.

- **Estimating equations:**
Appropriate objectives (inference functions) can be profiled:

- Heinze & Schemper (2002); Bull et al. (2007) for profiles of penalized likelihoods.
- Lindsay & Qu (2003) for profiles of appropriate quadratic score functions.
The \texttt{profileModel} R package has been developed to

- calculate,
- plot, and
- construct confidence intervals from

the profiles of \texttt{user-defined} objectives (via “plug-in” functions) for \texttt{arbitrary} \texttt{glm}-like fitted objects (\texttt{object}) with linear predictor.
Supported fitted objects

Fitted objects constructed according to Chambers & Hastie (1991, Chapter 2):

- The fitting procedure which results in object accepts offset in formula.
- object$call is the call that resulted in object.
- object$terms exists with the same meaning as for lm/glm objects.
- coef(object) returns a vector of coefficients with each component corresponding to a column of model.matrix(object)
The profileModel objective functions

- **Restricted fit**: Fix a parameter at a value and estimate the remaining parameters (using offset).
- The profiles of the objective are obtained/extracted from restricted fits.
The profileModel objective functions (cont.)

For example,

- object is the result of a glm call.
- Interest on the profiles of the log-likelihood (use deviance).

→ An appropriate profileModel objective is

```r
profObj <- function(restrFit, dispersion)
  restrFit$deviance/dispersion
```

- Within the profileModel function:

  → the restricted fits for a grid of parameter values are obtained, and
  → for each restricted fit the difference

    ```r
    profObj(restrFit) - profObj(object)
    ```

    is calculated.
Profiling some standard deviations away from the estimate

e.g.

```r
> library(MASS)
> m1 <- glm(Claims ~ District + Group + Age +
+    offset(log(Holders)), data = Insurance,
+    family = poisson)
> prof1 <- profileModel(m1, objective = profObj,
+    dispersion = 1)
> plot(prof1)
```
Why develop a package for profiling?
The profileModel R package
Examples
More on profileModel

Objective functions
The profileModel class and function
Confidence intervals

Kosmidis, I.  Profiling the parameters of models with linear predictors
Profiling over a grid of values.

e.g.

```r
> prof2 <- update(prof1,
+    which = paste("District", 2:4, sep=""),
+    grid.bounds = c(-0.5, 0, -0.1, 0.5, 0, 1))
> plot(prof2)
```
Why develop a package for profiling?
The profileModel R package
Examples
More on profileModel

Objective functions
The profileModel class and function
Confidence intervals

District2

District3

District4

Kosmidis, I. Profiling the parameters of models with linear predictors
Profiling until the profiles reach a certain value

- Construction of asymptotic confidence intervals.
- This procedure, currently, depends on the **convexity** of the objective.

```r
> prof3 <- update(prof2,
+   grid.bounds = NULL,
+   quantile = qchisq(0.95, 1))
> plot(prof3)
```
Why develop a package for profiling?

The profileModel R package

Examples

More on profileModel

Objective functions

The profileModel class and function

Confidence intervals

District2

Profiling the parameters of models with linear predictors

District3

District4

Kosmidis, I.
Asymptotic confidence intervals based on the profiles

- Using spline smoothing.
  - It is fast.
  - Useful for routine use.
- Using a binary search.
  - It is slower than smoothing but it returns accurate (up to a tolerance) endpoints.
  - Useful when the spline does not approximate well the profile (large departures from quadratic behaviour or asymptotes) and for empirical coverage studies.
Profile likelihood for `survreg` objects

An appropriate objective for `survreg` objects is

```r
> profLogLik <- function(restrFit) {
+   -2*restrFit$loglik[2]
+ }
```

Then,

```r
> library(survival)
> m3 <- survreg(
+   Surv(futime, fustat) ~ ecog.ps + rx,
+   ovarian, dist= "weibull", scale = 1)
> prof.m3 <- profileModel(m3,
+   quantile=qchisq(0.95,1),
+   objective = profLogLik,
+   stdErrors = summary(m3)$table[,2])
```
Profile likelihood for survreg objects (cont.)

- The 95% asymptotic profile confidence intervals are

```r
> ci.m3 <- profConfint(prof.m3)
> ci.m3

<table>
<thead>
<tr>
<th></th>
<th>Lower</th>
<th>Upper</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intercept)</td>
<td>4.504032</td>
<td>9.7478129</td>
</tr>
<tr>
<td>ecog.ps</td>
<td>-1.653003</td>
<td>0.711507</td>
</tr>
<tr>
<td>rx</td>
<td>-0.563139</td>
<td>1.801371</td>
</tr>
</tbody>
</table>
```

- The 95% Wald asymptotic confidence intervals are

```r
> confint(m3)

<table>
<thead>
<tr>
<th></th>
<th>2.5 %</th>
<th>97.5 %</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intercept)</td>
<td>4.371006</td>
<td>9.552669</td>
</tr>
<tr>
<td>ecog.ps</td>
<td>-1.583621</td>
<td>0.717352</td>
</tr>
<tr>
<td>rx</td>
<td>-0.568984</td>
<td>1.731989</td>
</tr>
</tbody>
</table>
```

- The confidence intervals are similar because the profiles are almost quadratic.

```r
> plot(prof.m3, signed = TRUE, cis = ci.m3)
```
Why develop a package for profiling?
The profileModel R package

Examples
More on profileModel

Profile likelihood for survreg objects
Infinite maximum likelihood estimates.
Infinite maximum likelihood estimates

Data:

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>13</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>12</td>
<td>20</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>18</td>
</tr>
</tbody>
</table>

```r
> x1 <- c(0, 0, 1, 1)
> x2 <- c(0, 1, 0, 1)
> y <- c(16, 1, 12, 0)
> tots <- c(16, 13, 20, 18)
> m2 <- glm(y/tots ~ x1 + x2,
+ weights = tots,
+ family=binomial(probit))
> coef(m2)
(Intercept) x1       x2
  6.649437 -6.396090 -8.075514
> coef(summary(m2))[,"Std. Error"]
(Intercept) x1       x2
  5914.617  5914.617  5914.617
```
Why develop a package for profiling?

The profileModel R package

Examples

More on profileModel

Profile likelihood for survreg objects

Infinite maximum likelihood estimates.

Kosmidis, I. Profiling the parameters of models with linear predictors
Infinite maximum likelihood estimates (cont.)

- Default profile method

```
> confint(m2)
Waiting for profiling to be done...
   2.5 %   97.5 %
(Intercept)   -511.1173     NA
  x1              NA  506.1762
  x2  -2561.4923    382.1928
```

- The `profileModel`'s method for confidence intervals.

```
> confintModel(m2, quantile = qchisq(0.95, 1),
+    stepsize = 0.2, objective = profObj,
+    dispersion = 1, method = "zoom")

                  Lower    Upper
(Intercept)  1.245953     Inf
  x1         -Inf  -0.8845107
  x2         -Inf -2.4205613
```
Documentation and conclusions

- Package and complementary material
  - Package available on CRAN (http://cran.r-project.org).
  - For more examples and further features see \texttt{?profileModel} and \texttt{?confintModel}, and
  - complementary material for \texttt{profileModel} on http://go.warwick.ac.uk/kosmidis/software.

- Key features
  - It allows developers to have access to profiling capabilities by merely authoring a function for the objective to be profiled
    \[ \rightarrow \] see \texttt{?RaoScoreStatistic} for an implementation of the quadratic score statistic for \texttt{glm}-like objects.
  - It provides an alternative to already implemented methods for profiling.
  - In its current version (0.5-4), it has been tested and it is known to work with objects resulting from \texttt{lm}, \texttt{glm}, \texttt{polr}, \texttt{gee}, \texttt{geeglm}, \texttt{brglm}, \texttt{BTm} and \texttt{survreg}.
Future development

- Profiling objectives for pairs of parameters and a method for plotting the contours of the profile.
- Quantile-based profiling and confidence intervals for non-convex objectives.
- Implementation using parallel computing.


