Teratology - Alternative Binary Models

August 21, 2024

First the teratology data are loaded. The data set exists in two different versions, "teratology" shows the aggregated count data whereas "teratology2" includes the original data.

```
> library(catdata)
> data(teratology)
> data(teratology2)
```

For the first two models, the simple independence model and the quasi-likelihood model, the aggregated data are needed.

```
> attach(teratology)
```

The simple and naive independence model is fitted by the following command.

```
> mLogit <- glm(cbind(D,L) ~ as.factor(Grp), family=binomial())
> summary(mLogit)
```

Now the quasi–likelihood model is fitted. The coefficients are the same as in the independence model before, only the standard errors have to be multiplied by $\sqrt{\hat{\phi}}$.

```
> mQuasi <- glm(cbind(D,L) ~ as.factor(Grp), family=quasibinomial(link="logit"))
> summary(mQuasi)
```

The next model to be fitted is a GEE with independence correlation structure. For that purpose the library "gee" is loaded.

```
> library(gee)
```

Now we use the original data set "teratology2".

```
> detach(teratology)
> attach(teratology2)
```

The GEE is fitted by the following command. The coefficients are again equal to those from the independence model, the standard errors for the independence models can be found in the column "Naive S.E.". The new standard errors from the GEE are those in the column "Robust S.E.".

```
> mGee <- gee(y ~ as.factor(Grp), id=Rat, family=binomial)
> summary(mGee)
```

For the following beta–binomial model the library "VGAM" with its function "vglm" and the data set "teratology" is needed.

```
> library(VGAM)

> detach(teratology2)
> attach(teratology)
```

Furthermore we construct the Variable N as sum of all fetuses in one litter. We will use N to make a subset with $N > 1$ for the beta–binomial model.

```
> N <- D + L
```

Now the beta–binomial model is fitted.

```
> mBetaBin <- vglm(cbind(D,L) ~ as.factor(Grp), family=betabinomial, subset=N>1)
> summary(mBetaBin)
```

For the following two mixed models again the original data are required.

```
> detach(teratology)
> attach(teratology2)
```

With the function "glmmPQL" from the "MASS"–library a mixed model is fitted by penalized quasi–likelihood, the mixed model contains random intercepts but no random slopes.

```
> mMixPql<- glmmPQL(y ~ as.factor(Grp), random=~1 | Rat, family=binomial)
> summary(mMixPql)
```

In order to fit a mixed model by maximum likelihood we load the library "glmmML".

```
> library(glmmML)
```

For a mixed model to be fitted by Gauss–Hermite quadrature we need the function "glmmML" with the option "method='ghq'", the favoured number of quadrature points is determined by the option "n.points".

```
> mGaussH <- glmmML(y ~ as.factor(Grp), cluster=Rat, method = "ghq", n.points = 14,
+                   boot = 0)
> summary(mGaussH)
```

Again we change the data set, for the discrete mixture model "teratology" is required.

```
> detach(teratology2)
> attach(teratology)
```

For discrete mixture models the library "flexmix" with its functions "flexmix" and "stepFlexmix" can be used.

```
> library(flexmix)
```

In "stepFlexmix" the procedure is run several times, the maximum likelihood solution is returned. The favoured number of iterations can be specified by the option "nrep". The number of components is determined by the option "k".

Due to random processes the results of different runs of "stepFlexmix" will differ slightly.

```
> detach(package:nlme)
> detach(package:VGAM)
> library(stats4)

> mDiscmix <-stepFlexmix(cbind(D,L) ~ 1, k = 2, nrep=5,
+                   model = FLXMRglmfix(family = "binomial",fixed =~as.factor(Grp)))
> summary(mDiscmix)
> parameters(mDiscmix)
```