Robust Inference in Generalized Linear Models

Claudio Agostinelli
claudio@unive.it

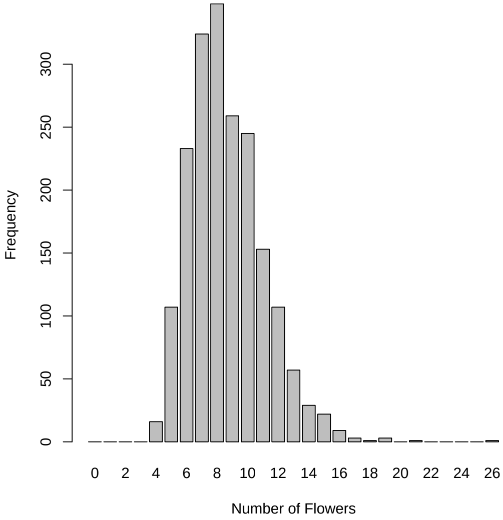
Dipartimento di Statistica
Università Ca' Foscari di Venezia
San Giobbe, Cannaregio 873, Venezia
Tel. 041 2347446, Fax. 041 2347444
http://www.dst.unive.it/˜claudio

12 August 2008

Outline

1. Introduction
    - Weighted Likelihood Estimators
    - How to construct the Weights
2. Robust estimation for GLM
    - Standard weights
    - Weights based on the Anscombe residuals
    - What is available
3. TODO list
4. Conclusions

Cyclamen dataset



This data (www.statsci.org/data/general/cyclamen.html) comes from an experiment on induction of flowering of cyclamen. ▸ R code

Weighted Likelihood Estimating Equations

The estimating equations of WLEE is a modified version of the MLE equations where at each score is associated a weight defined as follows

$$w(x; \theta, f^*) = \frac{A(\delta(x; \theta, f^*)) + 1}{\delta(x; \theta, f^*) + 1}$$

where $A(\delta)$ is the Residual Adjustment Function.
Hence the WLEE estimator is the solution of

$$\sum_{i=1}^{n} w(x_i; \theta, f^*) u(x_i; \theta) = 0$$

where $u(x; \theta)$ is the score function for the model.

Pearson Residuals

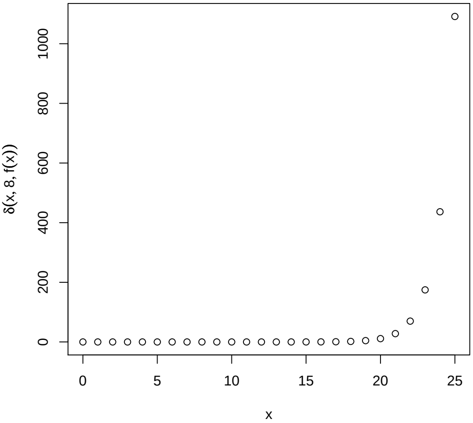
In our approach, outliers are observations that are highly unlikely to occur under the assumed model [see Markatou et al., 1995, 1998].

This definition is well adapted in many context since it is based on a **"probabilistic" distance**. One way to measure this discrepancy is to use the Pearson Residuals [Lindsay, 1994] defined as follows

$$\delta(x, \theta, f^*) = \frac{f^*(x)}{m^*(x; \theta)} - 1$$

where $f^*(x)$ is a non parametric density estimator based on the data and $m^*(x; \theta)$ is a smoothed version of the density of the model. Note that in the discrete case the smoothing is needed.

Pearson Residuals



Pearson Residuals ($\delta(x; \lambda = 8, f(x))$) for $f(x) = 0.98m(x; 8) + 0.02m(x; 20)$ where $m(x; \lambda)$ is the probability distribution of a Poisson distribution. ▸ R code

Residual Adjustment Function

In order to construct a weight function attached to the score function we need to choose a Residual Adjustment Function (RAF) [see Lindsay, 1994]. Here we will choose it inside two different families

- Power Divergence Measure [Cressie and Read, 1984, 1988];
- Generalized Kullback–Leibler Disparity [Park and Basu, 2003].

Power Divergence Measure RAF

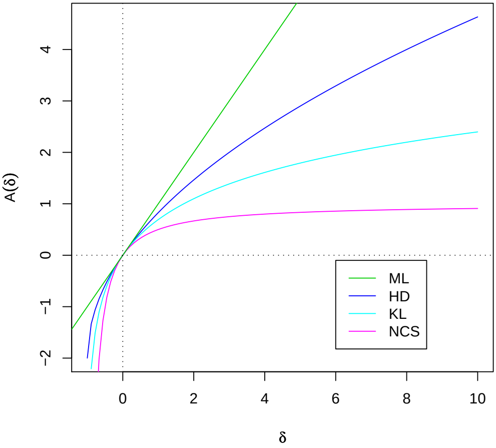
Residual Adjustment Function based on the Power Divergence Measure was introduced in Lindsay [1994]

$$A_{pdm}(\delta, \tau) = \begin{cases} \tau \left((\delta + 1)^{1/\tau} - 1 \right) & \tau < \infty \\ \log(\delta + 1) & tau = \infty \end{cases}$$

Special cases:

- $\tau = 1$: Maximum likelihood;
- $\tau = 2$: Hellinger distance;
- $\tau \to \infty$: Kullback–Leibler divergence;
- $\tau = -1$: Neyman's Chi–Square.

Power Divergence Measure



Residual Adjustment Function: Hellinger (HD), Kullback–Leibler (KL), Neyman's Chi–square (NCS) ▸ R code

Generalized Kullback–Leibler RAF

Residual Adjustment Function based on the Generalized Kullback–Leibler divergence was introduced in Park and Basu [2003]

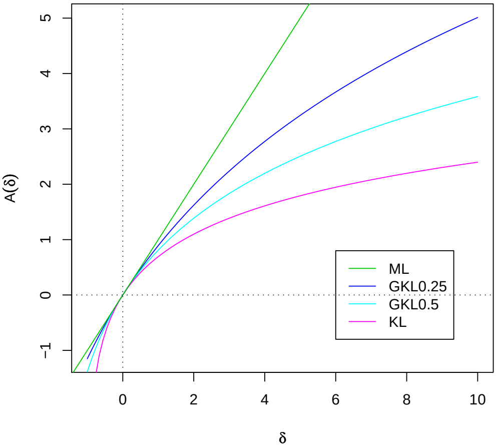
$$A_{gkl}(\delta, \tau) = \frac{\log(\tau\delta + 1)}{\tau} \qquad 0 \leq \tau \leq 1$$

Special cases:

- $\tau \rightarrow 0$: Maximum likelihood;
- $\tau = 1$: Kullback–Leibler divergence.

It could be interpreted as a linear combination between the Likelihood divergence and the Kullback–Leibler divergence.

Generalized Kullback–Leibler



Residual Adjustment Function: Generalized Kullback–Leibler ▸ R code

Example: Poisson distribution

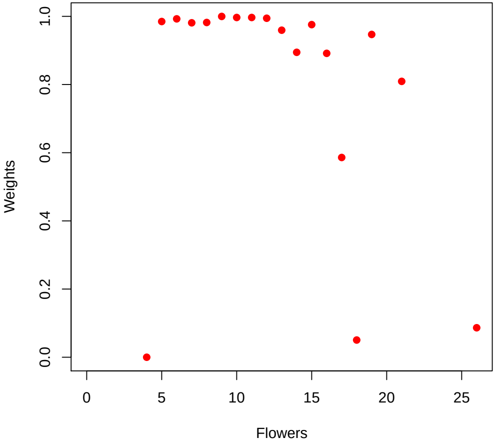
The WLEE for the Poisson distribution is the solution(s) of the following fixed point equation

$$\lambda = \frac{\sum_{i=1}^{n} w(x_i; \lambda) x_i}{\sum_{i=1}^{n} w(x_i; \lambda)}$$

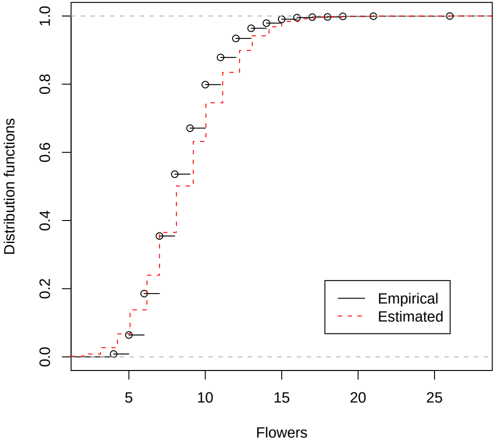
this is implemented in function `wle.poisson` in the package `wle`. For the `cyclamen` dataset we have

```
> wlepois <- wle.poisson(Flowers)
> wlepois
```

```
Call: wle.poisson(x = Flowers)
lambda: [1] 8.66
Number of solutions 1
```



Weights for the Cyclamen dataset ▸ R code



Cyclamen dataset: Comparison between the empirical distribution function and the estimated model distribution ▸ R code

Literature review

A. Bianco and VJ. Yohai. Robust estimation in the logistic regression model, in robust statistics. In H. Rieder, editor, *Data Analysis and Computer Intensive Methods, Proceedings of the workshop in honor of Peter J. Huber*, 1996.

E. Cantoni and E. Ronchetti. Robust inference for generalized linear models. *Journal of the American Statistical Association*, 2001.

M. Markatou, A. Basu, and BG. Lindsay. Weighted likelihood estimating equations: The discrete case with applications to logistic regression. *Journal of Statistical Planning and Inference*, 1997.

CH. Muller and N. Neykov. Breakdown points of trimmed likelihood estimators and related estimators in generalized linear models. *Journal of Statistical Planning and Inference*, 2003.

PJ. Rousseeuw and A. Christmann. Robustness against separation and outliers in logistic regression. *Computational Statistics & Data Analysis*, 2003.

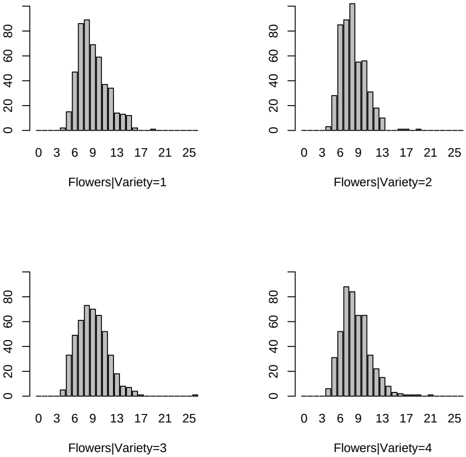
LA. Stefanski, RJ. Carroll, and D. Ruppert. Optimally bounded score functions for generalized linear models with applications to logistic regression. *Biometrika*, 1986.

WLEE for GLM

We need to consider two situations:

1. levels of the predictors with a “sufficient” number of replications;
   In this case we can define the weights as ususal by consider the conditional distribution of the response variable with respect to the corresponding levels.
2. levels of the predictors with one or “too few” number of replications;

Example: Flowers∼Variety



Conditional distribution of 'Flowers' given 'Variety' ▸ R code

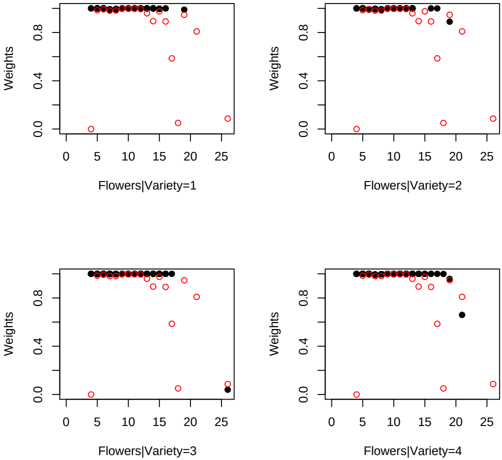
Example: Flowers∼Variety

```
> outvar <- glm(Flowers ~ Variety,
+     family = poisson)
> outvar
```

```
Call: glm(formula = Flowers  Variety, family = poisson)
Coefficients: (Intercept) Variety2 Variety3 2.1925842 -0.0989213
-0.0009307 Variety4 -0.0434527
Degrees of Freedom: 1917 Total (i.e. Null); 1914 Residual Null
Deviance: 1256 Residual Deviance: 1230 AIC: 8868
```

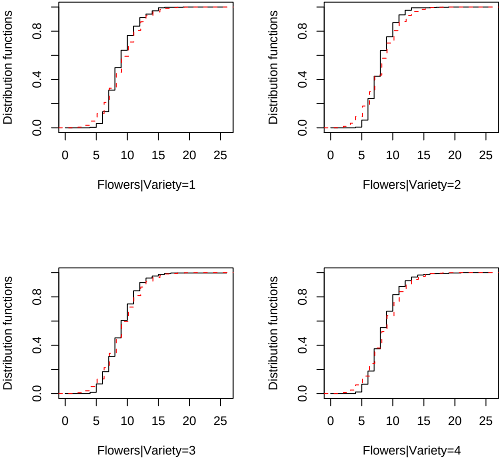
```
> wleoutvar <- wle.glm(Flowers ~
+     Variety, family = poisson)
> wleoutvar
```

```
Call: wle.glm(formula = Flowers  Variety, family = poisson)
Root: 1 Coefficients: (Intercept) Variety2 Variety3 2.192945
-0.099111 -0.005113 Variety4 -0.044739
Degrees of Freedom: 1917 Total (i.e. Null); 1914 Residual Null
Deviance: 1227 Residual Deviance: 1202 AIC: 8815
Number of solutions 1
```



Weights for the Cyclamen dataset in the model `Flowers∼Variety`

▸ R code



Conditional distributions for the Cyclamen dataset in the model `Flowers∼Variety` ▸ R code

WLEE for GLM

Case 2

- use observations in a neighborhood of the level predictors in order to evaluate the weights of the response in that level (This is implemented using the parameter `window.size` in the `wle.glm.control` function);
- use weights based on the asymptotic distribution of the Anscombe residuals (`use.asymptotic` in `wle.glm.control`).

Anscombe residuals

They are introduced in

D. R. Cox and E. J. Snell. A general definition of residuals. *Journal of the Royal Statistical Society. Series B (Methodological)*, 30(2):248–275, 1968.

R. M. Loynes. On cox and snell's general definition of residuals. *Journal of the Royal Statistical Society. Series B (Methodological)*, 31(1):103–106, 1969.

Donald A. Pierce and Daniel W. Schafer. Residuals in generalized linear models. *Journal of the American Statistical Association*, 81(396):977–986, 1986.

Rollin Brant. Residual components in generalized linear models. *The Canadian Journal of Statistics*, 15(2):115–126, 1987.

Anscombe residuals

The Anscombe residuals are obtain by the following transformation in both the Y and μ

$$A(y) = \int \mathrm{Var}(\mu)^{-1/3} \; d\mu$$

where $\mathrm{Var}(\mu)$ is the variance function expressed in terms of μ. The Anscombe residual adjusts for the scale of the variance by dividing by

$$\frac{\partial}{\partial \mu} A(y) \sqrt{\tau^2}$$

where $\tau^2 = \frac{\partial^2}{\partial \theta^2} b(\theta)$.

Anscombe residuals

- Poisson [Cox and Snell, 1968]

$$\frac{\frac{3}{2}(Y^{2/3} - (\mu - 1/6)^{2/3})}{\mu^{1/6}}$$

- Binomial (this function includes a bias correction term) [Cox and Snell, 1968]

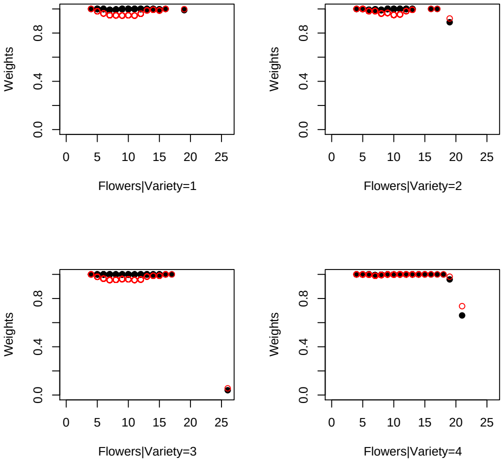
$$\frac{\phi(Y/m) - \phi(\theta - \frac{1}{6}(1 - 2\theta)/m)}{\theta^{1/6}(1 - \theta)^{1/6}/\sqrt{m}}$$

where $\phi(u) = \int_0^u t^{-1/3}(1 - t)^{-1/3}\, dt$ $(0 \leq u \leq 1)$ could be computed using the Incomplete Beta function.

Example: Flowers∼Variety

```
> wleoutvarasy <- wle.glm(Flowers ~
+     Variety, family = poisson,
+     control = list(glm = glm.control(),
+         wle = wle.glm.control(use.asymptotic = length(Flo
> wleoutvarasy
```

```
Call: wle.glm(formula = Flowers   Variety, family = poisson,
control = list(glm = glm.control(), wle =
wle.glm.control(use.asymptotic = length(Flowers))))
Root: 1 Coefficients: (Intercept) Variety2 Variety3 2.194016
-0.102452 -0.005794 Variety4 -0.045259
Degrees of Freedom: 1917 Total (i.e. Null); 1914 Residual Null
Deviance: 1209 Residual Deviance: 1183 AIC: 8594
Number of solutions 1
```



Comparison of Weights based on the Poisson and on the Anscombe residuals for the Cyclamen dataset in the model `Flowers∼Variety`

▸ R code

What is implemented

- **Family**: Poisson, Binomial, QuasiPoisson, QuasiBinomial;
- Estimation process;
- Print method.

How it is implemented

- `wle.glm` the main function. It accepts all the arguments of `glm`. The `control` argument has two components: `glm` which is the usual argument given by `glm.control()` and the `wle` argument given by `wle.glm.control()`.
- `wle.glm.fit` the function that performs the fit. It calls `glm.fit` and `wle.glm.weights`;
- `wle.glm.weights` the function that evaluates the weights for a given set of parameters;
- `wle.glm.control` controls the arguments for the WLEE part;
- `residuals.anscombe` which evaluates the Anscombe residuals for several family, since now, Poisson, Binomial, Gamma, InverseGamma.

TODO

- **Short term**
  - Documentation;
  - Check print method (S3);
  - Summary method (S3);
  - Plot method (S3) similar to the one available for `wle.lm`.
- **Long term**
  - Anova function (`anova.wle.glm`, S3), deviance (`deviance.wle.glm`, S3) and tests by the extension of the results in Agostinelli and Markatou [2001];
  - Add more families: Gamma, InverseGamma, Normal.
  - Model selection: the actual weights AIC (in print method) is based on actual model, this is not the best way to do, we will implement a method `extractAIC.wle.glm` with weights based on the full model;
- **Very Long term**
  - build function `wle.dglm` for over and under dispersed models;
  - build function `wle.bigglm` for big datasets.

Conclusions

- We introduced Robust estimators for Generalized Linear Models based on Weighted Likelihood Estimating Equations;
- We use the package `wle`;
- The `Sweave` of the presentation would be available at: `www.dst.unive.it/~claudio/R/index.html`;
- Functions will be available in the next release of the `wle` package, version 0.9-4.

```
> cyclamen <- na.omit(read.table("./cyclamen.txt",
+     header = TRUE))
> cyclamen$Variety <- factor(cyclamen$Variety)
> cyclamen$Regimem <- factor(cyclamen$Regimem)
> cyclamen$Fertilizer <- factor(cyclamen$Fertilizer)
> Flowers <- cyclamen$Flowers
> Variety <- cyclamen$Variety
> par(mai = c(1, 1, 0, 0))
> barplot(table(factor(Flowers, levels = 0:26)),
+     xlab = "Number of Flowers",
+     ylab = "Frequency")
```

◂ Return

```
> delta <- function(x, eps = 0.02) {
+     res <- ((1 - eps) * dpois(x,
+         8) + eps * dpois(x, 20))/dpois(x,
+         8) - 1
+     return(res)
+ }
> plot(0:25, delta(0:25), xlab = "x",
+     ylab = expression(delta(x,
+         8, f(x))))
```

◂ Return

```
> Apdm <- function(x, tau) {
+     if (tau == Inf)
+         a <- log(x + 1)
+     else a <- tau * ((x + 1)^(1/tau) -
+         1)
+     return(a)
+ }
> plot(function(x) Apdm(x, tau = 2),
+     from = -1, to = 10, xlab = expression(delta),
+     ylab = expression(A(delta)),
+     col = 4)
> plot(function(x) Apdm(x, tau = Inf),
+     from = -1, to = 10, col = 5,
+     add = TRUE)
> plot(function(x) Apdm(x, tau = -1),
+     from = -1, to = 10, col = 6,
+     add = TRUE)
```

```
> abline(0, 1, col = 3)
> abline(h = 0, lty = 3)
> abline(v = 0, lty = 3)
> legend(6, -0.1, legend = c("ML",
+     "HD", "KL", "NCS"), col = 3:6,
+     lty = rep(1, 4))
```

◂ Return

```
> Agkl <- function(x, tau) {
+     if (tau == 0)
+         a <- x
+     else a <- log(tau * x + 1)/tau
+     return(a)
+ }
> plot(function(x) Agkl(x, tau = 0.25),
+     from = -1, to = 10, xlab = expression(delta),
+     ylab = expression(A(delta)),
+     col = 4)
> plot(function(x) Agkl(x, tau = 0.5),
+     from = -1, to = 10, col = 5,
+     add = TRUE)
> plot(function(x) Agkl(x, tau = 1),
+     from = -1, to = 10, col = 6,
+     add = TRUE)
> abline(0, 1, col = 3)
> abline(h = 0, lty = 3)
> abline(v = 0, lty = 3)
> legend(6, 0.8, legend = c("ML",
+      "GKL0.25", "GKL0.5", "KL"),
+      col = 3:6, lty = rep(1, 4))
```

◂ Return

```
> nodup <- !duplicated(Flowers)
> plot(Flowers[nodup], wlepois$weights[nodup],
+     xlim = c(0, 26), xlab = "Flowers",
+     ylab = "Weights", pch = 19,
+     col = 2)
```

◂ Return

```
> plot(ecdf(Flowers), main = "",
+     xlab = "Flowers", ylab = "Distribution functions")
> plot(function(x) ppois(x, lambda = wlepois$lambda),
+     type = "s", add = TRUE, col = 2,
+     lty = 2)
> legend(x = "bottomright", legend = c("Empirical",
+     "Estimated"), col = 1:2, lty = 1:2,
+     inset = 0.1)
```

◂ Return

```
> flowersvariety <- sapply(split(factor(Flowers,
+     0:26), Variety), table)
> layout(matrix(1:4, nrow = 2, byrow = TRUE))
> barplot(flowersvariety[, 1], xlab = "Flowers|Variety=1",
+     ylim = c(0, 105))
> barplot(flowersvariety[, 2], xlab = "Flowers|Variety=2",
+     ylim = c(0, 105))
> barplot(flowersvariety[, 3], xlab = "Flowers|Variety=3",
+     ylim = c(0, 105))
> barplot(flowersvariety[, 4], xlab = "Flowers|Variety=4",
+     ylim = c(0, 105))
```

◂ Return

```
> fv <- split(1:length(Flowers),
+     Variety)
> nodup <- !duplicated(Flowers)
> layout(matrix(1:4, nrow = 2, byrow = TRUE))
> plot(Flowers[fv[[1]]], wleoutvar$root1$wle.weights[fv[[1
+     xlim = c(0, 26), xlab = "Flowers|Variety=1",
+     ylab = "Weights", pch = 19,
+     col = 1, ylim = c(0, 1))
> points(Flowers[nodup], wlepois$weights[nodup],
+     xlim = c(0, 26), col = 2)
> plot(Flowers[fv[[2]]], wleoutvar$root1$wle.weights[fv[[2
+     xlim = c(0, 26), xlab = "Flowers|Variety=2",
+     ylab = "Weights", pch = 19,
+     col = 1, ylim = c(0, 1))
> points(Flowers[nodup], wlepois$weights[nodup],
+     xlim = c(0, 26), col = 2)
> plot(Flowers[fv[[3]]], wleoutvar$root1$wle.weights[fv[[3
+     xlim = c(0, 26), xlab = "Flowers|Variety=3",
+     ylab = "Weights", pch = 19,
+     col = 1, ylim = c(0, 1))
> points(Flowers[nodup], wlepois$weights[nodup],
+     xlim = c(0, 26), col = 2)
> plot(Flowers[fv[[4]]], wleoutvar$root1$wle.weights[fv[[4
+     xlim = c(0, 26), xlab = "Flowers|Variety=4",
+     ylab = "Weights", pch = 19,
+     col = 1, ylim = c(0, 1))
> points(Flowers[nodup], wlepois$weights[nodup],
+     xlim = c(0, 26), col = 2)
```

◂ Return

```
> fit.val <- sapply(split(wleoutvar$root1$fitted.values,
+     Variety), function(x) x[1])
> layout(matrix(1:4, nrow = 2, byrow = TRUE))
> plot(0:26, cumsum(flowersvariety[,
+     1]/sum(flowersvariety[, 1])),
+     main = "", xlab = "Flowers|Variety=1",
+     ylab = "Distribution functions",
+     type = "s")
> plot(function(x) ppois(x, lambda = fit.val[1]),
+     type = "s", add = TRUE, col = 2,
+     lty = 2)
> plot(0:26, cumsum(flowersvariety[,
+     2]/sum(flowersvariety[, 2])),
+     main = "", xlab = "Flowers|Variety=2",
+     ylab = "Distribution functions",
+     type = "s")
> plot(function(x) ppois(x, lambda = fit.val[2]),
+     type = "s", add = TRUE, col = 2,
+     lty = 2)
> plot(0:26, cumsum(flowersvariety[,
+     3]/sum(flowersvariety[, 3])),
+     main = "", xlab = "Flowers|Variety=3",
+     ylab = "Distribution functions",
+     type = "s")
> plot(function(x) ppois(x, lambda = fit.val[3]),
+     type = "s", add = TRUE, col = 2,
+     lty = 2)
> plot(0:26, cumsum(flowersvariety[,
+     4]/sum(flowersvariety[, 4])),
+     main = "", xlab = "Flowers|Variety=4",
+     ylab = "Distribution functions",
+     type = "s")
> plot(function(x) ppois(x, lambda = fit.val[4]),
+     type = "s", add = TRUE, col = 2,
+     lty = 2)
```

◂ Return

```
> fv <- split(1:length(Flowers),
+     Variety)
> layout(matrix(1:4, nrow = 2, byrow = TRUE))
> plot(Flowers[fv[[1]]], wleoutvar$root1$wle.weights[fv[[1
+     xlim = c(0, 26), xlab = "Flowers|Variety=1",
+     ylab = "Weights", pch = 19,
+     col = 1, ylim = c(0, 1))
> points(Flowers[fv[[1]]], wleoutvarasy$root1$wle.weights[f
+     col = 2)
> plot(Flowers[fv[[2]]], wleoutvar$root1$wle.weights[fv[[2
+     xlim = c(0, 26), xlab = "Flowers|Variety=2",
+     ylab = "Weights", pch = 19,
+     col = 1, ylim = c(0, 1))
> points(Flowers[fv[[2]]], wleoutvarasy$root1$wle.weights[f
+     col = 2)
> plot(Flowers[fv[[3]]], wleoutvar$root1$wle.weights[fv[[3
+     xlim = c(0, 26), xlab = "Flowers|Variety=3",
+     ylab = "Weights", pch = 19,
+     col = 1, ylim = c(0, 1))
> points(Flowers[fv[[3]]], wleoutvarasy$root1$wle.weights[f
+     col = 2)
> plot(Flowers[fv[[4]]], wleoutvar$root1$wle.weights[fv[[4]
+     xlim = c(0, 26), xlab = "Flowers|Variety=4",
+     ylab = "Weights", pch = 19,
+     col = 1, ylim = c(0, 1))
> points(Flowers[fv[[4]]], wleoutvarasy$root1$wle.weights[f
+     col = 2)
```

◂ Return

C. Agostinelli and M. Markatou. Test of hypotheses based on the weighted likelihood methodology. *Statistica Sinica*, 11(2): 499–514, 2001.

D. R. Cox and E. J. Snell. A general definition of residuals. *Journal of the Royal Statistical Society. Series B (Methodological)*, 30 (2):248–275, 1968. ISSN 00359246. URL `http://www.jstor.org/stable/2984505`.

N. Cressie and T.R.C. Read. Multinomial goodness–of–fit tests. *Journal of the Royal Statistical Society, Series B*, 46:440–464, 1984.

N. Cressie and T.R.C. Read. *Cressie–Read Statistic*, pages 37–39. Wiley, 1988. In: Encyclopedia of Statistical Sciences, Supplementary Volume, edited by S. Kotz and N.L. Johnson.

B.G. Lindsay. Efficiency versus robustness: The case for minimum hellinger distance and related methods. *Annals of Statistics*, 22: 1018–1114, 1994.

M. Markatou, A. Basu, and B.G. Lindsay. Weighted likelihood estimating equations: The continuous case. Technical report, Department of Statistics, Columbia University, New York, 1995.

M. Markatou, A. Basu, and B.G. Lindsay. Weighted likelihood estimating equations with a bootstrap root search. *Journal of the American Statistical Association*, 93:740–750, 1998.

C. Park and A. Basu. The generalized kullback-leibler divergence and robust inference. *Journal of Statistical Computation and Simulation*, 73(5):311–332, 2003.

These slides are prepared using LaTeX, beamer class and Sweave package in R . They are compiled with R ver. 2.6.0 running under OS darwin8.10.1 and package `wle` ver. 0.9-3.