

# AIC from quasi-possion model

Meilin Yan

July 18, 2016

## AIC formula

```
# AIC = -2*loglikelihood + k*(Number of parameters in the fitted model)
# Number of parameters is the equivalent degree of freedom.
```

## Function to compute the Q-AIC in quasi-possion models from Antonio's paper

```
fqaic <- function(model) {
  loglik <- sum(dpois(model$y, model$fitted.values, log=TRUE))
  phi <- summary(model)$dispersion
  qaic <- -2*loglik + 2*summary(model)$df[3]*phi
  return(qaic)
}
```

## Verification of the Q-AIC function

Fit the model twice with possion (regular likelihood model) and quasi-possion Possion model. Use AIC to choose df per year.

```
library(dlnm)
```

```
## Warning: package 'dlnm' was built under R version 3.2.4
```

```
## This is dlnm 2.2.3. For details: help(dlnm) and vignette('dlnmOverview').
## Important changes: see file.show(system.file('Changesince220',package='dlnm'))
```

```
library(splines)
data("chicagoNMMAPS")
data <- chicagoNMMAPS

cb.pm <- crossbasis(data$pm10, lag=15, argvar=list(fun="lin"),
  arglag=list(fun="poly", degree=4))

cb.temp <- crossbasis(data$temp, lag=3, argvar=list(df=5),
  arglag=list(fun="strata", breaks=1))
```

```

# Poisson model
mod.posi.6 <- glm(death ~ cb.pm + cb.temp + ns(time, 6*14) + dow,
  family = poisson(), data)
mod.posi.7 <- glm(death ~ cb.pm + cb.temp + ns(time, 7*14) + dow,
  family = poisson(), data)
mod.posi.8 <- glm(death ~ cb.pm + cb.temp + ns(time, 8*14) + dow,
  family = poisson(), data)

# Quasi-Poisson model
mod.quasi.6 <- glm(death ~ cb.pm + cb.temp + ns(time, 6*14) + dow,
  family = quasipoisson(), data)
mod.quasi.7 <- glm(death ~ cb.pm + cb.temp + ns(time, 7*14) + dow,
  family = quasipoisson(), data)
mod.quasi.8 <- glm(death ~ cb.pm + cb.temp + ns(time, 8*14) + dow,
  family = quasipoisson(), data)

```

## Get log-likelihoods from poisson and quasi-poisson models

```

# Poisson model
loglik.posi <- logLik(mod.posi.6)

# Quasi-Poisson model
loglik.quasi <- sum(dpois(mod.quasi.6$y, mod.quasi.6$fitted.values, log=TRUE))

loglik.posi

```

```
## 'log Lik.' -13653.47 (df=105)
```

```
loglik.quasi
```

```
## [1] -13653.47
```

```
# They are the same
```

## Get number of parameters from the two models

The dispersion parameter, which was forced to be 1 in a poisson model, is allowed to be estimated in a quasi-poisson model. And the dispersion parameter tells us how many times larger the variance is than the mean.

- In a poisson model,
  - number of parameters = number of coefficients.
- In a quasi-poisson model,
  - number of parameters = (number of coefficients)\*(dispersion parameter).

```
# Poission model
# number of coefficients
summary(mod.posi.6)$df[3]
```

```
## [1] 106
```

```
# dispersion parameters
summary(mod.posi.6)$dispersion
```

```
## [1] 1
```

```
# Quasi-Poission model
# number of coefficients
n.coef.quasi <- summary(mod.quasi.6)$df[3]
n.coef.quasi
```

```
## [1] 106
```

```
# dispersion parameters
n.disper.quasi <- summary(mod.quasi.6)$dispersion
n.disper.quasi
```

```
## [1] 1.298815
```

```
# number of parameters
n.coef.quasi*n.disper.quasi
```

```
## [1] 137.6744
```

## Compare model based on AIC

```
# Poission model
# AIC calculated in R
AIC(mod.posi.6, mod.posi.7, mod.posi.8) # model with 7 df per year has the smallest AIC
```

```
##           df      AIC
## mod.posi.6 105 27516.94
## mod.posi.7 118 27504.14
## mod.posi.8 130 27426.50
```

```
# AIC calculated with fqaic function
c(fqaic(mod.posi.6), fqaic(mod.posi.7), fqaic(mod.posi.8)) # model with 7 df per year has the smallest .
```

```
## [1] 27518.94 27508.14 27434.50
```

Same results but the values of AIC vary a little. I guess R calculates AIC by using a more accurate or complex formula. I am going to figure it out.

```

# Quasi-possion model
AIC(mod.quasi.6, mod.quasi.7, mod.quasi.8) # Not available from R

##           df AIC
## mod.quasi.6 105  NA
## mod.quasi.7 118  NA
## mod.quasi.8 130  NA

c(fqaic(mod.quasi.6), fqaic(mod.quasi.7), fqaic(mod.quasi.8)) # model with 8 df has the smallest AIC

## [1] 27582.28 27578.68 27505.00

```

So I think the Q-AIC function is OK for us to extract AIC from a quasi-possion model.