Generalized linear model validation

Bert van der Veen

Department of Mathematical Sciences, NTNU

The model

Writing the linear model:

$$y_i = \alpha + \mathbf{x}_i \boldsymbol{\beta} + \epsilon_i \sim \mathcal{N}(0, \sigma^2) \tag{1}$$

Is the same as:

$$\mathbb{E}(y_i | \mathbf{x}_i) = \alpha + \mathbf{x}_i \boldsymbol{\beta} \tag{2}$$

as long as $\mathbb{E}(\epsilon_i) = 0$.

Generalised linear model

$$\begin{aligned} g\{\mathbb{E}(y_i|x_i)\} &= \eta_i = \alpha + x_i\beta \\ \mathbb{E}(y_i|x_i) &= g^{-1}(\eta_i) = g^{-1}(\alpha + x_i\beta) \end{aligned} \tag{3}$$

GLMs do not have a residual term

GLM Assumptions

- No outliers
- Independence
- Correct distribution
- Correct link function
- Correct variance function (implied by previous two)

Outline

- Residual plots for checking GLM assumptions
- Issues with residuals in GLMs
- Types of residuals
- Residual diagnostics
- Dharma/quantile residuals
- Prediction error

Modeling checking

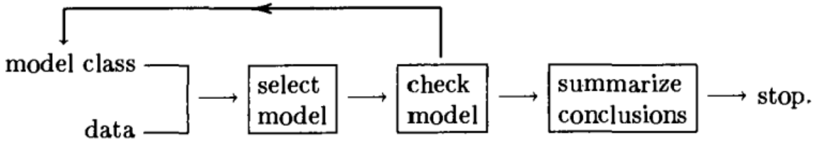


Figure 1: McCullagh and Nelder (1989) workflow

Methods for checking GLM assumptions

- ~~Tests~~
- Outliers: Cook's distance, Residual vs. fitted
- Independence: Partial residual plots, Residuals vs. fitted, Rresiduals vs. lagged residuals
- Correct distribution: QQ-plot
- Correct link function: residuals vs. fitted, include η_i in the model (Hinkley 1985) , linear predictor against transformed response
- Correct variance function: Abs(residuals) vs. fitted

Response residuals

We could use the same residual as in linear regression:

$$\epsilon_i = y_i - \widehat{\mu}_i$$

But we do not expect these to look nice in GLMs.

- Because the variance depends on the mean
- We want nice looking residuals when a model is good
- We want bad looking residuals otherwise

GLM residuals: Pearson's

Even though we lack the ϵ_i term in GLMs, we still calculate the residuals similarly

GLM residuals: Pearson's

Even though we lack the ϵ_i term in GLMs, we still calculate the residuals similarly

$$\widehat{\epsilon_{i,pearson}} = \frac{y_i - \hat{\mu}_i}{\sqrt{\mathsf{var}(y_i; \hat{\mu}_i, \hat{\phi})}} \tag{4}$$

as the scaled difference between data and mean

GLM residuals: Pearson's

Even though we lack the ϵ_i term in GLMs, we still calculate the residuals similarly

$$\widehat{\epsilon_{i,pearson}} = \frac{y_i - \hat{\mu}_i}{\sqrt{\mathsf{var}(y_i; \hat{\mu}_i, \hat{\phi})}} \tag{4}$$

as the scaled difference between data and mean

Approximately normally distributed in large samples

Recall deviance?

We do not have a ϵ_i term in GLMs, but we do have the deviance function

Recall deviance?

We do not have a ϵ_i term in GLMs, but we do have the deviance function

$$D(\mathbf{y}; \hat{\boldsymbol{\mu}}) = \sum_{i=1}^{n} 2y_i\{g(y_i) - g(\hat{\mu}_i)\} - y_i + \hat{\mu}_i \tag{5}$$

Which represents distance to the saturated model

GLM residuals: deviance

$$D(\mathbf{y}; \hat{\boldsymbol{\mu}}) = \sum_{i=1}^{n} 2y_i\{g(y_i) - g(\hat{\mu}_i)\} - y_i + \hat{\mu}_i \tag{6}$$

GLM residuals: deviance

$$D(\mathbf{y}; \hat{\boldsymbol{\mu}}) = \sum_{i=1}^{n} 2y_i\{g(y_i) - g(\hat{\mu}_i)\} - y_i + \hat{\mu}_i \tag{6}$$

- Note the summation over the observations
- Each observation has a defined contribution to the deviance
- $d_i = 2y_i\{g(y_i) - g(\hat{\mu}_i)\} - y_i + \hat{\mu}_i$
- We can use this to define a residual

GLM residuals: deviance

$$D(\mathbf{y}; \hat{\boldsymbol{\mu}}) = \sum_{i=1}^{n} 2y_i\{g(y_i) - g(\hat{\mu}_i)\} - y_i + \hat{\mu}_i \tag{6}$$

- Note the summation over the observations
- Each observation has a defined contribution to the deviance
- $d_i = 2y_i\{g(y_i) - g(\hat{\mu}_i)\} - y_i + \hat{\mu}_i$
- We can use this to define a residual

Deviance residuals

$$\epsilon_{i,\widehat{deviance}} = \text{sign}(y_i - \hat{\mu}_i)\sqrt{d_i}, \qquad \text{so that } \sum_{i=1}^{n} \epsilon_{\widehat{deviance},i}{}^2 \tag{7}$$

Approximately normally distributed in large samples

Why deviance residuals

- Converges faster to approximate normality

  (Dunn and Smyth 2018, Cox and Snell 1968)
- Can be adjusted for small samples (Pierce and Schafer 1986)
  - e.g. Poisson $\mu_i^{-0.5}/6$

Why deviance residuals

- Converges faster to approximate normality

  (Dunn and Smyth 2018, Cox and Snell 1968)
- Can be adjusted for small samples (Pierce and Schafer 1986)
  - e.g. Poisson $\mu_i^{-0.5}/6$

Still inappropriate for discrete data and small samples

In practice

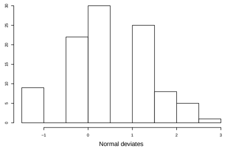
In practice, both Pearson's and Deviance residuals are often non-normally distributed.

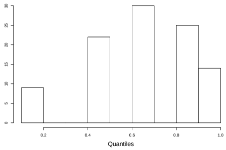
Randomized Quantile residual (Dunn and Smyth 1996)

- Gold standard residual
- Better suited for small samples and discrete data types
- Exactly normally distributed
- Suitable for all kinds of models

Continuous

$$r_Q = \Phi^{-1}\left\{\mathcal{F}\left(y_i; \hat{\mu}_i, \hat{\phi}\right)\right\} \tag{8}$$

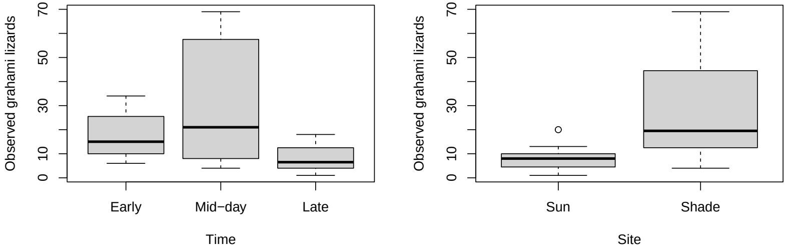




Example

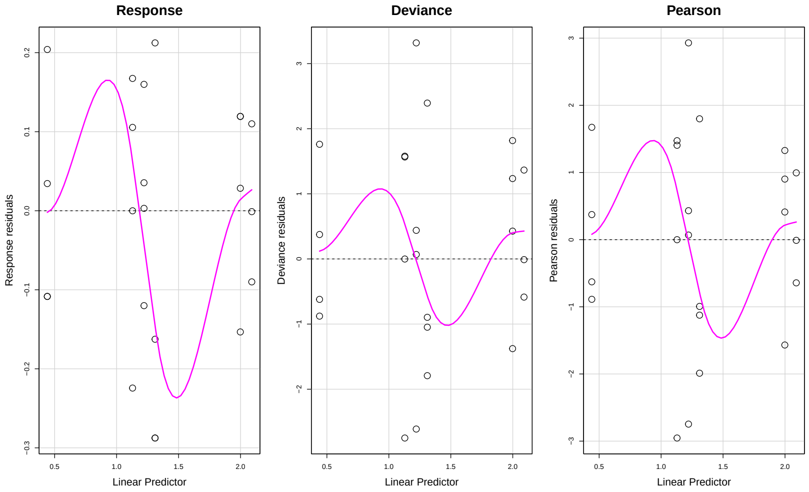
Figure 2: nwf.org

Lizards: recap

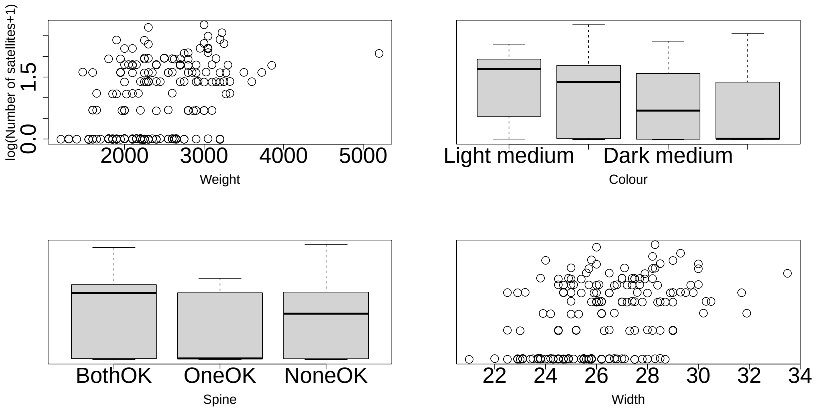


```
model1 <- glm(cbind(grahami, opalinus)~Time+Site,
              data = lizards, family="binomial")
```

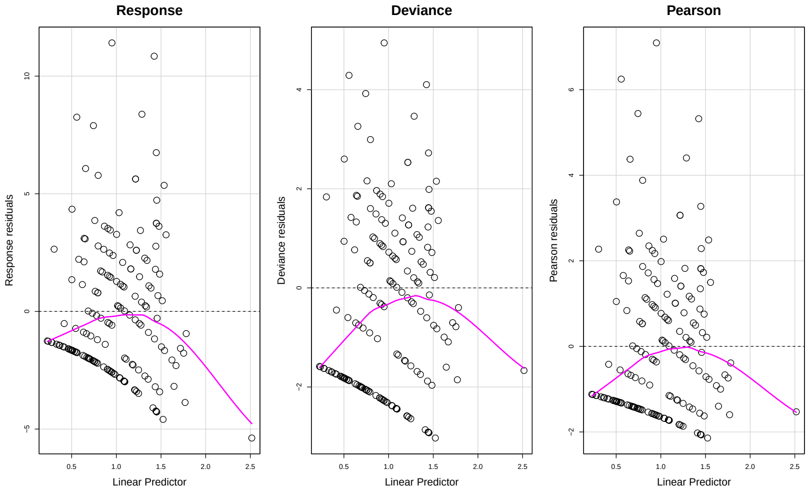
Lizards: residuals



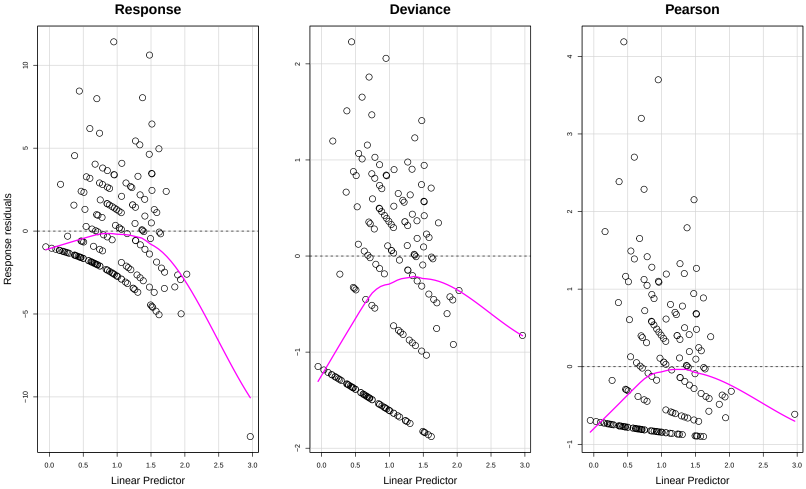
Horseshoe crabs: recap



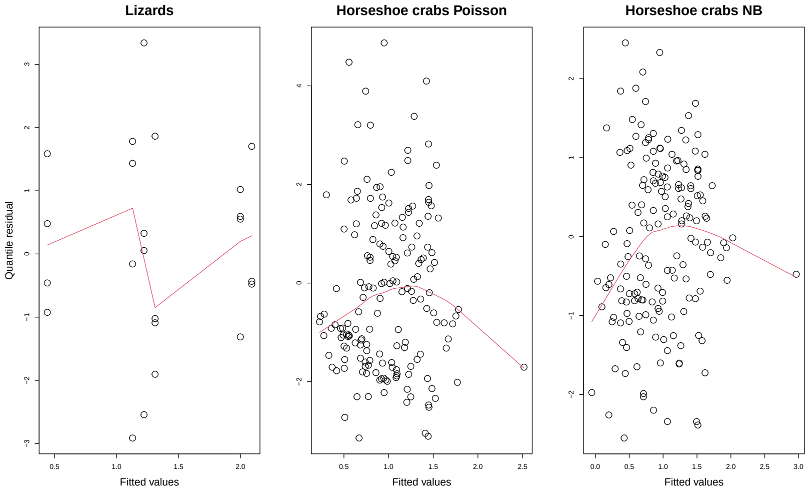
Horseshoe crabs: Poisson model residuals



Horseshoe crabs: NB model residuals



Lizards and Horseshoe crabs



Prediction

When you are only interested in predictions, e.g.:

- 20 years from now
- on a map
- new values of predictors x_i

Some assumptions might not matter. Generally, we still do not want structural deviations from the model.

Finding a model that predicts well

Usual procedure

1. Fit model to a part of the data (”train” and ”test”)
2. Predict for the remaining part (”test”)
3. Quantify prediction error (e.g., RMSE)
4. Adjust model to minimize 3)

Notes

Finding a model that predicts well: better

unUsual procedure

1. Fit model to ~~a part of~~ the data ~~("train" and "test")~~
2. ~~Predict for the remaining part ("test")~~ Collect new data
3. Quantify prediction error (e.g., RMSE)
4. Adjust model to minimize 3)

Notes

Finding a model that predicts well: better

unUsual procedure

1. Fit model to ~~a part of~~ the data ~~("train" and "test")~~
2. ~~Predict for the remaining part ("test")~~ Collect new data
3. Quantify prediction error (e.g., RMSE)
4. Adjust model to minimize 3)

Notes

1. Statistically leaving out data is weird
2. We want as much information for our model as possible.
3. The model should be applicable beyond just the data that we observed

Quantifying prediction error

MSE: $\frac{1}{n}\sum_{i=1}^{n}(y_i - \mu_i)^2$

RMSE: $\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \mu_i)^2}$

MAD: $\text{median}(|\mathbf{y} - \boldsymbol{\mu}|)^2$

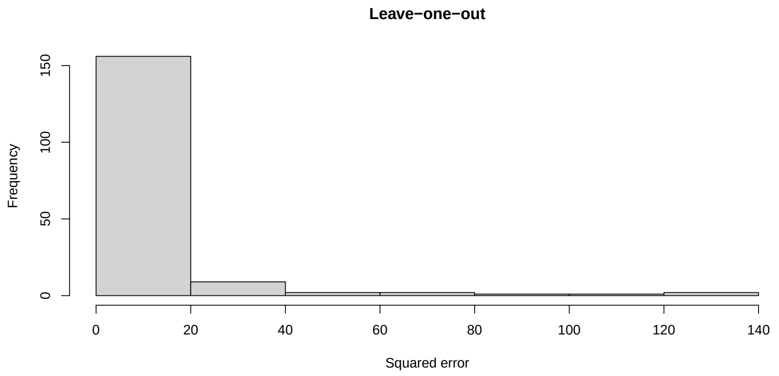
Or another metric. “AUC” is often used in binomial models.

For prediction, we want a model that performs well on one of these metrics.

Leave-one-out cross-validation Horseshoe crabs

```
SE <- NULL
for(i in 1:nrow(hcrabs)){
set.seed(i)
train <- hcrabs[-i,]
model2 <- update(model2, data = train)
SE <- c(SE,(hcrabs[i, "Sat"]-
  predict(model2,newdata = hcrabs[i, ], type="response"))^2)
}
```

Result



```
cat("RMSE: ", sqrt(mean(SE)))
```

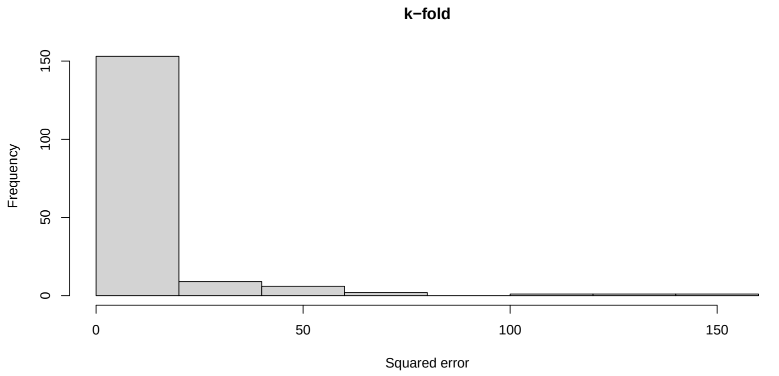
```
## RMSE:  3.168937
```

K-fold cross-validation Horseshoe crabs

```
k = 5
SE <- NULL
shuffle <- sample(1:nrow(hcrabs), nrow(hcrabs))
folds <- split(1:nrow(hcrabs), rep(1:k, length.out = nrow(hcr

for(i in 1:k){
idx <- folds[[i]]
test <- hcrabs[idx,]
train <- hcrabs[-idx, ]
model <- update(model2, data = train)
SE <- c(SE,(test$Sat-
  predict(model, newdata = test, type="response"))^2)
}
```

Result



```r
cat("RMSE: ", sqrt(mean(SE)))
```

```
## RMSE:  3.225183
```

Stratified cross-validation

- Cross-validation in a structured fashion: time or space.
- Select roughly equal partitions
  - Time: fit to a few years per partition
  - Space: areas
- I.e., blocking for non-independent responses

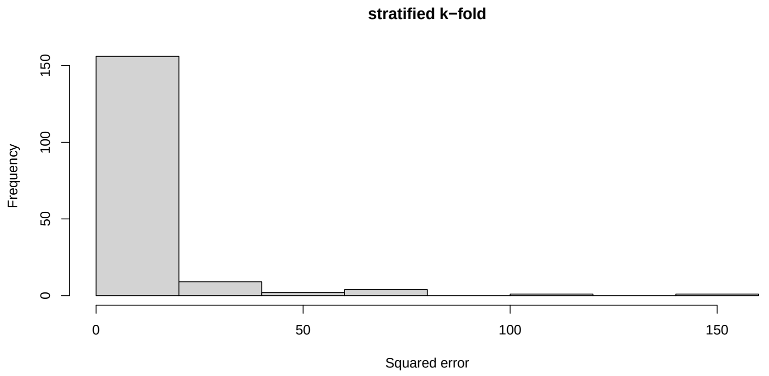
Stratified k-fold cross-validation Horseshoecrabs

Ensure a (roughly) similar distribution in every fold. For binary data, true stratified CV maintains the proportion.

```
k = 5
SE <- NULL
folds <- split(1:nrow(hcrabs),sample(rep(1:k, length.out = nrow(hcrabs))))

for(i in 1:k){
test <- hcrabs[order(hcrabs$Sat),][folds[[i]],]
train <- hcrabs[order(hcrabs$Sat),][-folds[[i]],]
model <- update(model2, data = train)
SE <- c(SE,(test$Sat-
predict(model, newdata = test, type="response"))^2)
}
```

Result



```
cat("RMSE: ", sqrt(mean(SE)))
```

```
## RMSE:  3.147592
```

Summary

- I did not mention Anscombe residuals, nor working residuals
- Do not draw conclusions before validating your model
- Use residual diagnostics/plots!
- If violated, adjust the model
- For prediction, look at cross-validation
  - Consider the structure of the data
  - Use independently collected data
  - There are many forms of cross-validation and quantifying error
  - Find model that minimizes the prediction error

Useful packages: `car`, `DHARMa`, `boot`