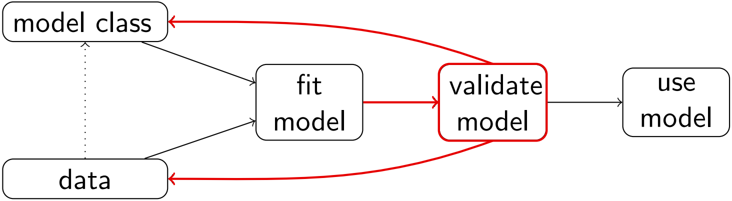
Lecture 11. Modelling Process and Model Diagnostics (cont.)

Nan Ye

School of Mathematics and Physics
University of Queensland

Recall: Some key modelling activities



- Check model assumption
  - Check goodness of fit, residual plot et al on training set.
  - A good fit on the training set may mean overfitting.
- Check predictive performance
  - Check cross-validation score, validation set performance.
- Reconsider model class or data if checks are not satisfactory.

This Lecture

- Checking model assumption
- Checking predictive performance

Residual Plots

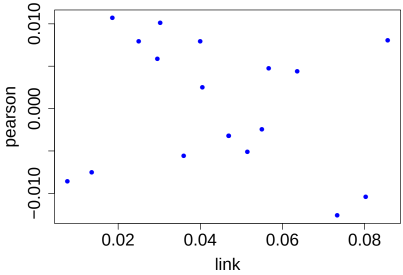
- Plot Pearson residuals/deviance residuals against link (i.e. linear predictor).
- If the model is correct, the points should be roughly uniformly scattered around 0.
- Plotting against the fitted mean (i.e. response) can be helpful but less popular.

Example
Consider plots of Pearson residuals againt the link (linear predictor) for models on the blood clotting time example.

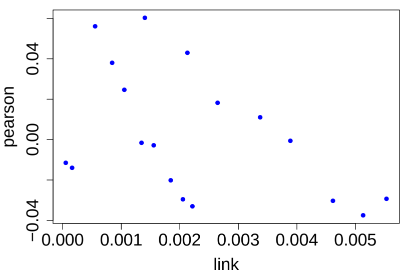
Recall the following models

```
> fit.ig.inv = glm(time ~ lot * log(conc), data=clot,
    family=inverse.gaussian(link='inverse'))
> fit.ig.invquad = glm(time ~ lot * log(conc), data=clot,
    family=inverse.gaussian)
> fit.ig.log = glm(time ~ lot * log(conc), data=clot,
    family=inverse.gaussian(link='log'))
> fit.gam.inv = glm(time ~ lot * log(conc), data=clot, family=Gamma)
...
```

Comparison of link functions



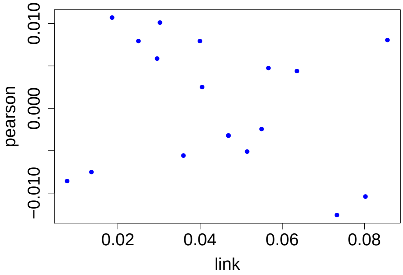
(a) fit.ig.inv



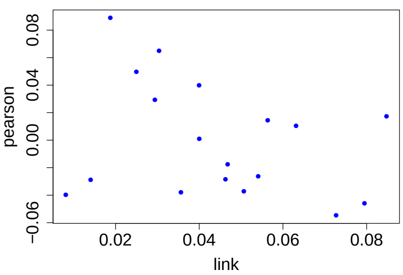
(b) fit.ig.invquad

- Residual decreases as link increases for inverse quadratic link
- No such obvious pattern for inverse link.
- Inverse link model is thus likely to be better.
- This is consistent with conclusions obtained using likelihood or residual deviance (see previous lectures).

Comparison of variance functions



(a) fit.ig.inv

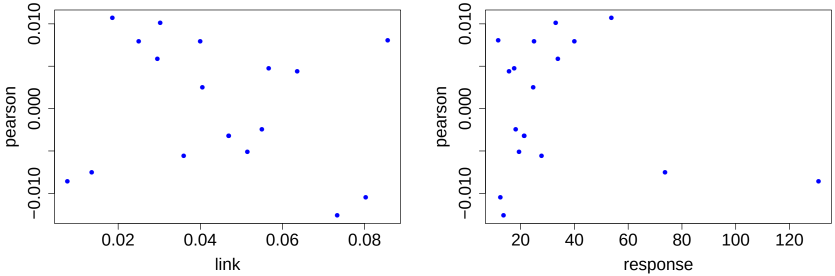


(b) fit.gam.inv

- Residuals on the RHS are close to 0 for Gamma.
- No such obvious pattern for inverse Gaussian.
- Inverse Gaussian thus likely has a better variance structure.
- This is consistent with conclusions obtained using likelihood.

```
> logLik(fit.gam.inv)
'log Lik.' -26.59759 (df=5)
> logLik(fit.ig.inv)
'log Lik.' -25.33805 (df=5)
```

Link scale vs. mean scale

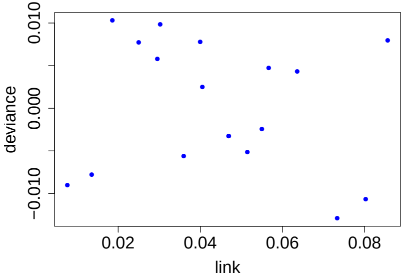


(a) link scale

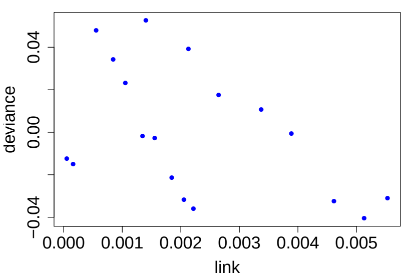
(b) mean scale

- Both are Pearson residual plots for fit.ig.inv.
- The mean scale spreads out the rightmost two points too much.
- These two points appear to be outliers on the mean scale, but not on the link scale.

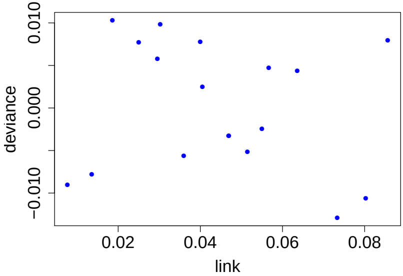
Deviance residual plots



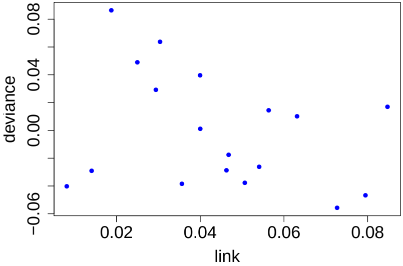
(a) fit.ig.inv



(b) fit.ig.invquad



(a) fit.ig.inv



(b) fit.gam.inv

- We get roughly the same plots, and thus roughly the same conclusions as using the Pearson residual plots.
- In fact, the Pearson residuals and the deviance residuals are almost the same for the models considered here.

Analysis of Deviance

- We successively fit a sequence of models by adding one term to the model.
- The deviance of a term is the difference between the deviance of the first model that contains it and the deviance of the previous model.
- Thus the deviance of a term depends on when it is added.

Example

```
> anova(fit.ig.inv)
Terms added sequentially (first to last)

              Df Deviance Resid. Df Resid. Dev       F    Pr(>F)
NULL                             17   0.247884
lot            1 0.034159        16   0.213725  492.04 2.630e-12 ***
log(conc)      1 0.203628        15   0.010097 2933.14 < 2.2e-16 ***
lot:log(conc)  1 0.009122        14   0.000975  131.40 1.679e-08 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

- The deviance of a term is F-distributed under the null hypothesis that the term is not significant.
- All terms are significant in this example.
- log(conc) has the largest contribution in the model.

```
> fit.ig.inv1 = glm(time ~ log(conc)*lot, data=clot,
    family=inverse.gaussian(link='inverse'))
> anova(fit.ig.inv1)
Terms added sequentially (first to last)

              Df Deviance Resid. Df Resid. Dev       F    Pr(>F)
NULL                             17   0.247884
log(conc)      1 0.206543        16   0.041341 2975.13 < 2.2e-16 ***
lot            1 0.031244        15   0.010097  450.06 4.829e-12 ***
log(conc):lot  1 0.009122        14   0.000975  131.40 1.679e-08 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

- The order of lot and log(conc) are swapped.
- The deviances are slightly different.
- However, we have the same qualitative conclusion about the signifiance of the terms.

- Often, we need to decide whether a factor should be included.
- This can be done by comparing the deviances of before and after including it.
- Again, the conclusion depends on the model on which the factor is added.

```
> fit1 = glm(time ~ log(conc), data=clot,
    family=inverse.gaussian(link='inverse'))
> fit2 = glm(time ~ lot*log(conc), data=clot,
    family=inverse.gaussian(link='inverse'))
> anova(fit1, fit2, test='F')
Analysis of Deviance Table

Model 1: time ~ log(conc)
Model 2: time ~ lot * log(conc)
  Resid. Df Resid. Dev Df Deviance      F    Pr(>F)
1        16   0.041341
2        14   0.000975  2 0.040367 290.73 3.971e-12 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

The lot factor is significant.

Checking Predictive Performance

Overfitting

- A model satisfying the model assumption does not necessarily make good predictions on test data.
- In particular, when there are many covariates, a model which better fits the training data may have poorer performance than one which fits less well.
- Overfitting: as model complex increases, the model fits the training set better and better, but the test set performance first improves and then drops.

Measuring predictive performance

- The validation set approach
  - If we have enough data, we can split the dataset into a training set a validation set.
  - Train models using the training set, and pick the one with best predictive performance on the validation set.
- Cross-validation (CV)
  - We split the dataset into K folds (parts).
  - For each model class, train K models by leaving one fold out each time, and make predictions on the left-out fold.
  - The performance of predictions obtained using CV is the predictive performance of the model class.

```
> library(caret)
> train(time ~ lot*log(conc), method="glm", data=clot,
    family=inverse.gaussian(link='inverse'),
    trControl=trainControl(method="LOOCV"))
Resampling results:

  RMSE      Rsquared   MAE
  15.98637  0.9575552  5.65666
```

- In leave-one-out CV, each fold has only one example.
- The caret library provides a simple way to do CV for many models, including GLMs.

```
> train(time ~ lot*log(conc), method="glm", data=clot,
    family=inverse.gaussian(link='log'),
    trControl=trainControl(method="LOOCV"))
Resampling results:

  RMSE      Rsquared   MAE
  13.34795  0.8315472  6.159968
```

- Using the log link improves RMS, but decreases R^2 and MAE.
- This is what usually happens: no single model performs the best for all performance measures.

```
> train(time ~ lot*log(conc), method="glm", data=clot,
    family=inverse.gaussian(link='1/mu^2'),
    trControl=trainControl(method="LOOCV"))
Resampling results:

  RMSE      Rsquared   MAE
  5.791858  0.9130303  3.973965

Warning messages:
1: In sqrt(eta) : NaNs produced
2: In sqrt(eta) : NaNs produced
```

- Inverse quadratic link is only legitimate when one can ensure on a new $\mathbf{x}$, $\beta^\top \mathbf{x} > 0$.
- In this example, it happens that this positivity constraint is violated twice (eta refers to the linear predictor).

What You Need to Know

- Checking model assumption: residual plots, analysis of deviance.
- Checking predictive performance: validation set, cross-validation.