

Model options

- Exponential family
Gamma, inverse Gaussian
- Several links are commonly used

name	$g(\mu)$	$\mu \geq 0$	canonical link
log	$\ln(\mu)$	yes	
inverse	μ^{-1}	no	for Gamma distribution
inverse-quadratic	μ^{-2}	yes	for inverse-Gaussian distribution

Gamma Regression

Recall

- When Y is a non-negative continuous random variable, we can choose the systematic and random components as follows.

$$\text{(systematic)} \quad \mathbb{E}(Y \mid \mathbf{x}) = \exp(\beta^\top \mathbf{x})$$

(random) $Y \mid \mathbf{x}$ is Gamma distributed.

- We further assume the variance of the Gamma distribution is μ^2/ν (ν treated as known), thus

$$Y \mid \mathbf{x} \sim \Gamma(\mu = \exp(\beta^\top \mathbf{x}), \text{var} = \mu^2/\nu),$$

where $\Gamma(\mu = a, \text{var} = b)$ denotes a Gamma distribution with mean a and variance b .

Parameter interpretation

- Using log-link, $\mu = \exp(\mathbf{x}^\top \boldsymbol{\beta})$.
- One unit increase in x_i changes the mean by a factor of $\exp(\beta_i)$.
- No such simple interpretation for inverse link and inverse quadratic link.

Fisher scoring

- Consider the case of log link

$$Y \mid \mathbf{x} \sim \Gamma(\mu = \exp(\beta^\top \mathbf{x}), \text{var} = \mu^2/\nu),$$

- Let $\mu_i = \exp(\mathbf{x}_i^\top \beta)$.
- The gradient and the Fisher information are

$$\nabla \ell(\beta) = \sum_i \frac{\nu(y_i - \mu_i)}{\mu_i} \mathbf{x}_i,$$

$$I(\beta) = \sum_i \nu \mathbf{x}_i^\top \mathbf{x}_i,$$

- Fisher scoring updates β to

$$\beta' = \beta + I(\beta)^{-1} \nabla \ell(\beta).$$

Note that ν actually has no effect on the update.

- Let \mathbf{X} be the design matrix,

$$\begin{aligned}\boldsymbol{\mu} &= (\mu_1, \dots, \mu_n), \\ A &= \text{diag}(\nu(y_1 - \mu_1), \dots, \nu(y_n - \mu_n)),\end{aligned}$$

- In matrix notation, the gradient and the Fisher information are

$$\begin{aligned}\nabla \ell(\beta) &= \mathbf{X}^\top A(\mathbf{y} - \boldsymbol{\mu}), \\ I(\beta) &= \nu \mathbf{X}^\top \mathbf{X}.\end{aligned}$$

Example

Data

	id	conc	time	lot	id	conc	time	lot
1	5	118	1	10	5	69	2	
2	10	58	1	11	10	35	2	
3	15	42	1	12	15	26	2	
4	20	35	1	13	20	21	2	
5	30	27	1	14	30	18	2	
6	40	25	1	15	40	16	2	
7	60	21	1	16	60	13	2	
8	80	19	1	17	80	12	2	
9	100	18	1	18	100	12	2	

- Blood clotting times in seconds under different plasma concentration and two lots of thromboplastin.
- Normal plasma diluted to nine different concentrations.
- Two lots of thromboplastin.

Gamma: inverse link (canonical)

```
> fit.gam.inv = glm(time ~ lot * log(conc), data=clot, family=Gamma)
> summary(fit.gam.inv)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)

(Intercept) -0.0165544  0.0008655 -19.127 1.97e-11 ***
lot2        -0.0073541  0.0016780  -4.383 0.000625 ***
log(conc)    0.0153431  0.0003872   39.626 8.85e-16 ***
lot2:log(conc) 0.0082561  0.0007353  11.228 2.18e-08 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

(Dispersion parameter for Gamma family taken to be 0.002129707)

$$\mu = \begin{cases} (-0.0165544 + 0.0153431 * \log(\text{conc}))^{-1}, & \text{if } \text{lot}=1. \\ (-0.0073744 + 0.0082575 * \log(\text{conc}))^{-1}, & \text{if } \text{lot}=2. \end{cases}$$

Gamma: inverse quadratic link

```
> fit.gam.invquad = glm(time ~ lot * log(conc), data=clot,
  family=Gamma(link='1/mu^2'))
> summary(fit.gam.invquad)

Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.004e-03 1.470e-04 -6.831 8.18e-06 ***
lot2         -1.486e-03 4.056e-04 -3.664 0.002551 **
log(conc)     6.649e-04 8.795e-05 7.560 2.63e-06 ***
lot2:log(conc) 1.002e-03 2.403e-04 4.171 0.000941 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

(Dispersion parameter for Gamma family taken to be 0.03015227)

Gamma: log-link

```
> fit.gam.log = glm(time ~ lot * log(conc), data=c1ot,
  family=Gamma(link='log'))
> summary(fit.gam.log)

Coefficients:
Estimate Std. Error t value Pr(>|t| )
(Intercept) 5.50323 0.18794 29.282 5.83e-14 ***
lot2        -0.58447 0.26578 -2.199 0.0452 *
log(conc)   -0.60192 0.05462 -11.020 2.77e-08 ***
lot2:log(conc) 0.03448 0.07725 0.446 0.6621
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

(Dispersion parameter for Gamma family taken to be 0.02375284)

- The lot factor does not show strong effect when we use log link.
- This is qualitatively different from the cases for the inverse link and inverse quadratic link.

```
> logLik(fit.gam.inv)
'log Lik.' -26.59759 (df=5)
> logLik(fit.gam.invquad)
'log Lik.' -50.13667 (df=5)
> logLik(fit.gam.log)
'log Lik.' -47.98692 (df=5)
```

Gamma regression with inverse link has the best fit (much better than the other two).

Inverse Gaussian Regression

Inverse Gaussian distribution

- The PDF is given by

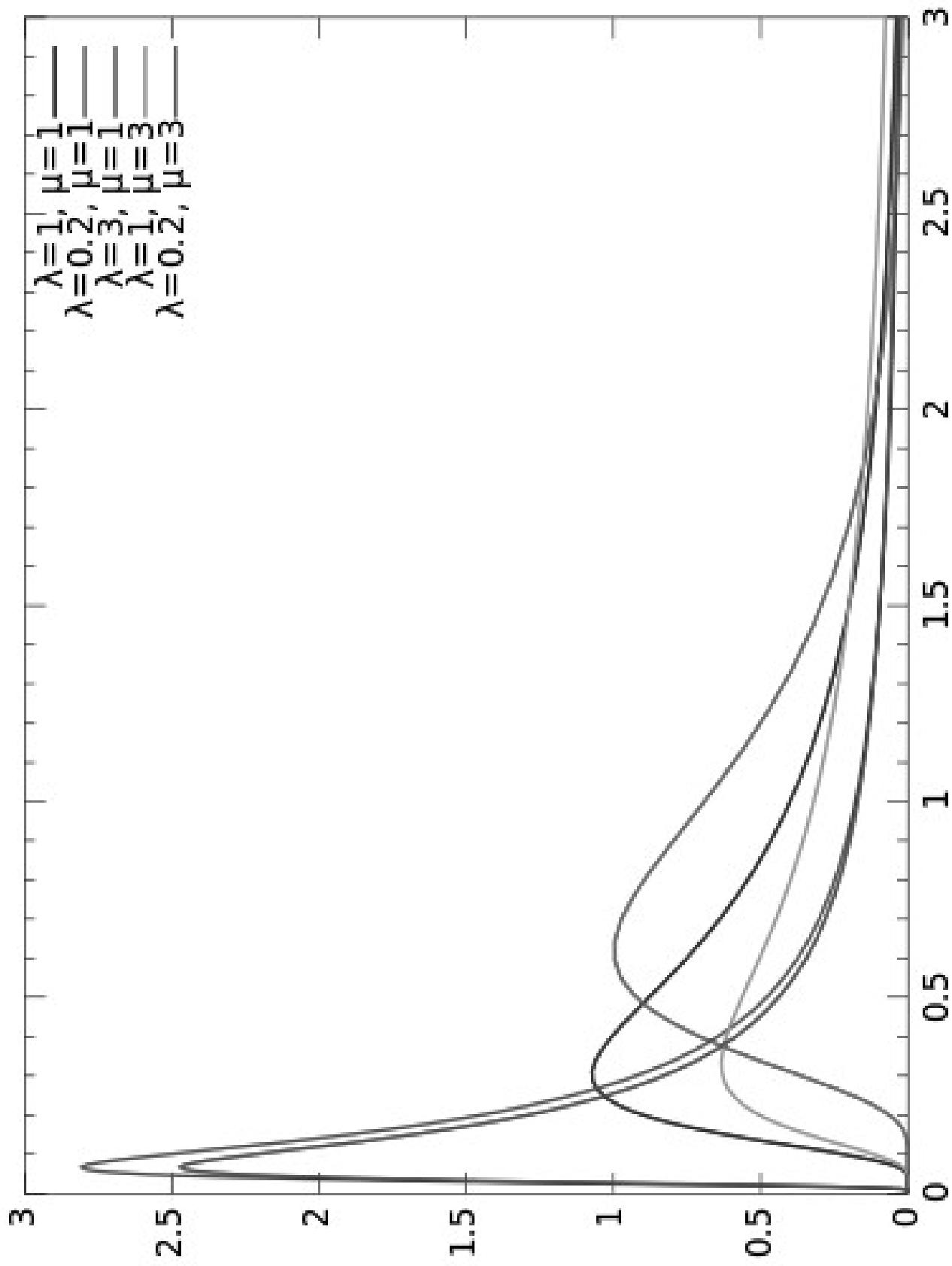
$$f(y \mid \mu, \lambda) = \left[\frac{\lambda}{2\pi y^3} \right]^{1/2} \exp \left\{ \frac{-\lambda(y - \mu)^2}{2\mu^2 y} \right\},$$

where μ is the mean and λ is the shape.

- The variance is cubic in the mean

$$\text{var}(X) = \mu^3 / \lambda.$$

PDF of inverse Gaussians.



Example (cont.)

Inverse Gaussian: inverse link

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

Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.0177893 0.0012377 -14.373 8.95e-10 ***
lot2         -0.0073744 0.0020333 -3.627 0.00275 **
log(conc)     0.0158014 0.0004350 36.327 2.96e-15 ***
lot2:log(conc) 0.0082575 0.0007075 11.671 1.33e-08 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

(Dispersion parameter for inverse.gaussian family taken to be
6.942317e-05)

Inverse Gaussian: inverse-quadratic link (canonical)

```
> fit.invquad = glm(time ~ lot * log(conc), data=clot,  
family=inverse.gaussian)  
> summary(fit.invquad)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-1.108e-03	1.761e-04	-6.291	1.99e-05 ***
lot2	-1.617e-03	4.024e-04	-4.018	0.001269 **
log(conc)	7.219e-04	9.954e-05	7.253	4.21e-06 ***
lot2:log(conc)	1.071e-03	2.233e-04	4.797	0.000284 ***

```
Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1
```

(Dispersion parameter for inverse.gaussian family taken to be
0.001216639)

Inverse Gaussian: log link

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

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	5.29038	0.23211	22.793	1.82e-12 ***
lot2	-0.56699	0.29495	-1.922	0.0752 .
log(conc)	-0.54163	0.06068	-8.925	3.75e-07 ***
lot2:log(conc)	0.02969	0.07725	0.384	0.7065

```
Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1
```

(Dispersion parameter for inverse.gaussian family taken to be
0.000758257)

```
> logLik(fit.ig.inv)
'log Lik.' -25.33805 (df=5)
> logLik(fit.ig.invquad)
'log Lik.' -50.26075 (df=5)
> logLik(fit.ig.log)
'log Lik.' -45.55859 (df=5)
```

Inverse Gaussian regression with inverse link has the best fit (much better than the other two).

Some Observations

- Link function plays an important role in fitting a good model.
inverse link is the best for both Gamma and inverse Gaussian in our example
- When the same link is used, the coefficients are similar for different exponential families
for each link, compare the coefficients for Gamma and inverse Gaussian in our example..