A Short Introduction to Generalized Linear Models

Katie Frank

2/12/2020

Overview

In this document, I introduce a special class of statistical models known as generalized linear models, focusing on their formulation and estimation. Towards the end of the document, I work through an example of fitting a generalized linear model to survival data.

Exponential family

The **exponential family** of distributions are an important class of distributions in statistics. For a random variable Y parameterized by θ and ϕ, the probability density function (pdf) of Y belongs to the exponential family if it can be expressed in the form

$$f(y;\theta,\phi) = \exp\left[\frac{y\theta - b(\theta)}{a(\phi)} + c(y,\phi)\right], \tag{1}$$

where $a(\phi), b(\theta)$, and $c(y,\phi)$ are known functions. Typically $a(\phi)$ has the form

$$a(\phi) = \frac{\phi}{p},$$

where p is a set *prior weight*; in most cases $p = 1$. The parameter θ is known as the *canonical parameter* and is usually the parameter of interest, while ϕ is a nuisance parameter that affects the variance. If Y is assumed to be generated by a distribution in the exponential family, then

$$\mathrm{E}[Y] = \mu = b'(\theta) \tag{2}$$

$$\mathrm{Var}(Y) = \sigma^2 = b''(\theta)a(\phi), \tag{3}$$

where $b'(\theta)$ and $b''(\theta)$ denote the first and second derivatives of $b(\theta)$. Many popular distributions, such as the normal, Bernoulli, and Poisson, are members of the exponential family.

Exponential example

The exponential distribution is a member of the exponential family. Let $Y \sim \text{Exp}(\lambda)$. The pdf of Y is

$$f(y;\lambda) = \lambda e^{-\lambda y}, \quad y > 0,$$

where $\lambda > 0$ is known as the *rate parameter* of the distribution. This expression can be rewritten as

$$f(y;\lambda) = \exp\left[-\lambda y + \log \lambda\right].$$

Here, $\theta = -\lambda$. Thus, $b(\theta) = -\log(-\theta)$. Also, $a(\phi) = \phi = 1$ and $c(y,\phi) = 0$. From equations (2) and (3), the mean and variance are

$$\begin{aligned} \text{E}[Y] &= b'(\theta) = -\frac{1}{\theta} = \frac{1}{\lambda} \\ \text{Var}(Y) &= b''(\theta)a(\phi) = \frac{1}{\theta^2} = \frac{1}{\lambda^2}. \end{aligned}$$

The exponential distribution is used to model the time until an event occurs, otherwise known as the waiting time. For instance, the time to failure of a specific brand of light bulb could be well-modeled by an exponential distribution, where λ controls the rate of failure. A large value for λ indicates that the bulbs tend to fail more quickly, while a small λ implies the opposite. Figure 1 illustrates this idea.

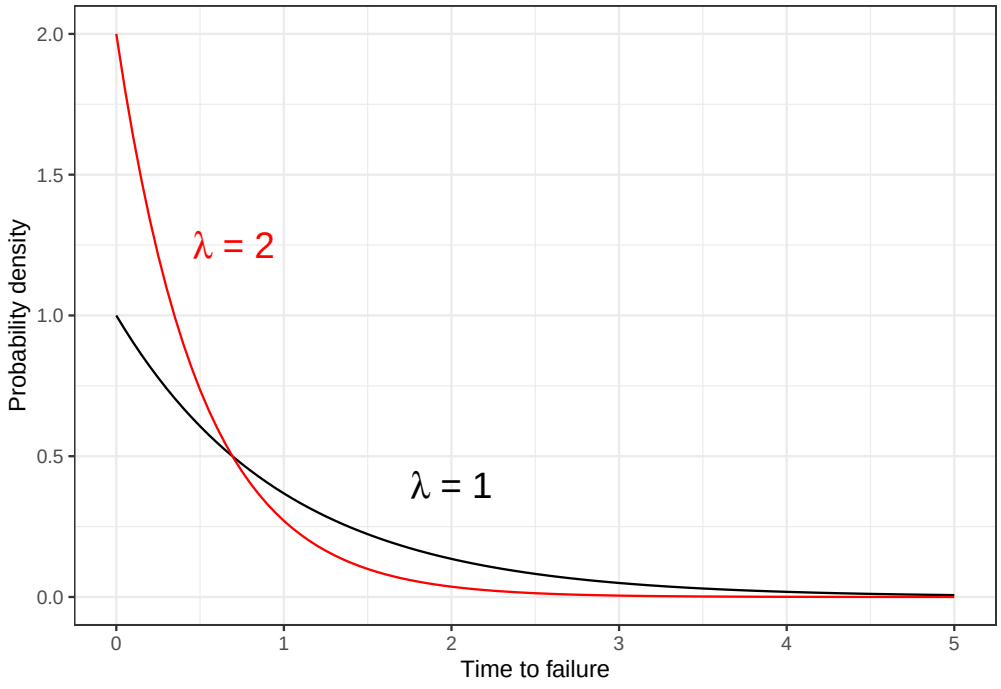


Figure 1: Probability density function of Exp(1) and Exp(2) in black and red, respectively.

Generalized linear models

The generalized linear model (GLM) can be thought of as an extension of linear regression to non-normal response variables. In a GLM, the *linear predictor* $\eta_i = \mathbf{x}_i'\boldsymbol{\beta}$ is related to the response variable y_i through a *link function*, denoted g. The function g must be monotone increasing and differentiable.

In the linear predictor,

- $\mathbf{x}_i$ is a $(p \times 1)$ vector of explanatory/predictor variables for the ith object/subject.

$$\mathbf{x}_i = \begin{bmatrix} x_{i1} \\ \vdots \\ x_{ip} \end{bmatrix} \quad \text{so} \quad \mathbf{x}_i' = \begin{bmatrix} x_{i1} & \cdots & x_{ip} \end{bmatrix}$$

  The vector $\mathbf{x}_i'$ constitutes the ith row of the design matrix $\mathbf{X}$.

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1' \\ \vdots \\ \mathbf{x}_n' \end{bmatrix} = \begin{bmatrix} x_{11} & \cdots & x_{1p} \\ \vdots & & \vdots \\ x_{n1} & \cdots & x_{np} \end{bmatrix}$$

  In usual cases, n (the number of objects/subjects) is much larger than p.

- $\boldsymbol{\beta}$ is the $(p \times 1)$ vector of model coefficients to be estimated.

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_p \end{bmatrix}$$

Given predictors $\mathbf{x}_i$, the response variable y_i is assumed to be generated from a distribution in the exponential family with mean μ_i. The expected value of y_i given explanatory variables $\mathbf{x}_i$ is defined

$$\mathrm{E}[y_i|\mathbf{x}_i] = \mu_i = g^{-1}(\eta_i),$$

where $\eta_i = \mathbf{x}_i'\boldsymbol{\beta}$ is the linear predictor and g is the link function.

A GLM consists of three components.

1. *Random component* - the probability distribution of the response variable $y_i|\mathbf{x}_i$ (e.g., $y_i|\mathbf{x}_i \sim \mathrm{N}(\mu_i, \sigma^2)$ in linear regression and $y_i|\mathbf{x}_i \sim \mathrm{Bern}(\pi_i)$ in logistic regression).
2. *Systematic component* - the linear combination of explanatory variables used to create the linear predictor (i.e., $\eta_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}$).
3. *Link function* - the connection between the random and systematic components. it says how μ_i, the mean of $y_i|\mathbf{x}_i$, is related to the linear predictor $\eta_i = \mathbf{x}_i'\boldsymbol{\beta}$.

If the link function connects the linear predictor η_i and the canonical parameter θ_i such that $\eta_i = \theta_i$, then the link function is *canonical*.

Estimation

Estimation of the model parameters $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_p)'$ in a GLM are obtained through maximum likelihood using an iteratively reweighted least-squares (IRLS) algorithm. Here is how it works. Start off with an initial estimate of the parameters $\hat{\boldsymbol{\beta}}$.

1. Calculate the $n \times 1$ *working response* vector $\mathbf{z}$. The ith element of $\mathbf{z}$ is

$$z_i = \hat{\eta}_i + (y_i - \hat{\mu}_i)\left(\frac{\partial \eta_i}{\partial \mu_i}\right),$$

   where $\frac{\partial \eta_i}{\partial \mu_i}$ is evaluated at $\hat{\boldsymbol{\beta}}$.

2. Calculate the $n \times n$ diagnonal *weight matrix* $\mathbf{W}$, where the ith element is

$$w_{ii} = \frac{1}{\text{Var}(Y_i|\mathbf{x}_i)}\left(\frac{\partial \mu_i}{\partial \eta_i}\right)^2.$$

3. Calculate the (new) weighted least-squares estimate

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}\mathbf{z}. \tag{4}$$

Repeat this procedure until the difference between successive approximations $\hat{\boldsymbol{\beta}}^{(m)} - \hat{\boldsymbol{\beta}}^{(m-1)}$ is smaller than some tolerance level. Equation 4 is a convenient way of expressing the weighted least-squares estimate. It is equivalent to

$$\hat{\boldsymbol{\beta}}^{(m)} = \hat{\boldsymbol{\beta}}^{(m-1)} + \mathcal{I}\left(\hat{\boldsymbol{\beta}}^{(m-1)}\right)^{-1} U(\hat{\boldsymbol{\beta}}^{(m-1)}), \tag{5}$$

where $\mathcal{I}(\hat{\boldsymbol{\beta}}^{(m-1)})$ is the expected information matrix and $U(\hat{\boldsymbol{\beta}}^{(m-1)})$ is the score function.

GLM example with survival data

This example comes from Exercise 4.2 on p. 77-78 of Dobson and Barnett (2018). The column `surv_weeks` in the truncated data frame below contains survival times (in weeks) from diagnosis to death for seventeen patients with leukemia. The `log_wbc` column denotes each patient's $\log_{10}$(initial white blood cell count).

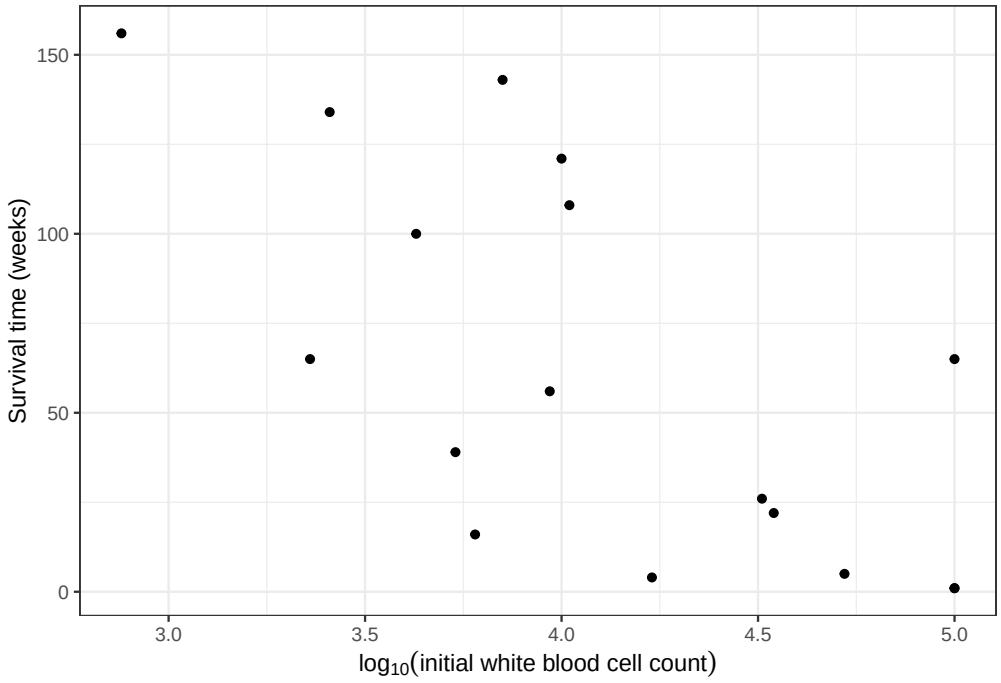
```
dat <- read.table("leukemia.txt", header = TRUE, sep = "\t")
head(dat)
```

```
##   surv_weeks log_wbc
## 1         65    3.36
## 2        156    2.88
## 3        100    3.63
## 4        134    3.41
## 5         16    3.78
## 6        108    4.02
```

If we plot `surv_time` against `log_wbc`, we see that survival time decreases approximately exponentially as initial log white blood cell (WBC) count increases. Thus, we assume $Y_i|x_i \sim \text{Exp}(\lambda_i)$, for $i = 1, \ldots, n$. Also, we assume the Y_i's are independent.

```
library(ggplot2); theme_set(theme_bw())

ggplot(dat, aes(log_wbc, surv_weeks)) +
  geom_point() +
  xlab(expression(log[10]("initial white blood cell count"))) +
  ylab("Survival time (weeks)")
```



If we want to model the survival times based on $\log_{10}$(initial WBC count) one possible specification is

$$\mathrm{E}[y_i|x_i] = \mu_i = \exp(\beta_1 + \beta_2 x_i),$$

where y_i is the survival time and x_i is the log WBC count. Here, the exp() in $\exp(\beta_1 + \beta_2 x_i)$ is the inverse link function. So, we are working with the log link: $g(\mu_i) = \log(\mu_i)$. Unlike the canonical link, which is $g(\mu_i) - \mu_i^{-1}$, the log link ensures that μ_i is non-negative for all values of the parameters and x_i.

In our specification for the conditional expectation of Y_i, we have

$$\begin{aligned}\mathrm{E}[Y_i|x_i] = \mu_i &= \exp(\beta_1 + \beta_2 x_i) \\ &= \mathbf{x}_i'\boldsymbol{\beta},\end{aligned}$$

where $\boldsymbol{\beta} = \begin{bmatrix}\beta_1 \\ \beta_2\end{bmatrix}$ and $\mathbf{x}_i = \begin{bmatrix}1 \\ x_i\end{bmatrix}$ for $i = 1, \ldots, n$.

The link function is

$$\log(\mu_i) = \mathbf{x}_i'\boldsymbol{\beta} = \eta_i.$$

Thus, $\frac{\partial \mu_i}{\partial \eta_i} = \exp(\eta_i)$ and $\frac{\partial \eta_i}{\partial \mu_i} = \frac{1}{\mu_i}$.

The ith element of the $n \times 1$ working response vector $\mathbf{z}$ is

$$\begin{aligned}z_i &= \eta_i + (y_i - \mu_i)\left(\frac{\partial \eta_i}{\partial \mu_i}\right) \\ &= \log(\mu_i) + (y_i - \mu_i)\left(\frac{1}{\mu_i}\right) \\ &= \mathbf{x}_i'\boldsymbol{\beta} + (y_i - \exp(\mathbf{x}_i'\boldsymbol{\beta}))\frac{1}{\exp(\mathbf{x}_i'\boldsymbol{\beta})} \\ &= \mathbf{x}_i'\boldsymbol{\beta} + \frac{y_i}{\exp(\mathbf{x}_i'\boldsymbol{\beta})} - 1.\end{aligned}$$

The ith weight in the $n \times n$ diagonal weight matrix $\mathbf{W}$ is

$$\begin{aligned}w_{ii} &= \frac{1}{\mathrm{Var}(Y_i|\mathbf{x}_i)}\left(\frac{\partial \mu_i}{\partial \eta_i}\right)^2 \\ &= \lambda_i^2\left[\exp(\eta_i)\right]^2 \\ &= \lambda_i^2\left[\exp(\log(\mu_i))\right]^2 \\ &= \lambda_i^2\mu_i^2 \\ &= 1.\end{aligned}$$

We now have everything we need to find the maximum likelihood estimates (MLEs). In the code below, I start the IRLS procedure with $\hat{\beta}_1 = 10$ and $\hat{\beta}_2 = -1$.

```
irls_exp <- function(x, y, be1, be2, max_iter = 100, tol = 1e-5){
  X <- matrix(c(rep(1, length(x)), x), ncol = 2)
  be <- matrix(c(be1, be2), ncol = 1)
  be_prev <- be

  for(iter in 1:max_iter){
    z <- X %*% be + y / (exp(X %*% be)) - 1
    cov_mat <- solve(t(X) %*% X)
    be <- cov_mat %*% t(X) %*% z

    if (abs(be[1] - be_prev[1]) < tol && abs(be[2] - be_prev[2]) < tol) {
      break
    } else {
      be_prev <- be
    }
  }
  list(be = be, cov_mat = cov_mat, num_iter = iter)
}

irls_exp(dat$log_wbc, dat$surv_weeks, 10, -1)
```

```
## $be
##           [,1]
## [1,]  8.477497
## [2,] -1.109298
##
## $cov_mat
##            [,1]       [,2]
## [1,]  2.7383886 -0.6542095
## [2,] -0.6542095  0.1597237
##
## $num_iter
## [1] 8
```

For our given tolerance level of 1×10^{-5}, it took 8 iterations to obtain the MLEs. These estimates are $\hat{\beta}_1 = 8.4775$ and $\hat{\beta}_2 = -1.1093$, and their estimated covariance matrix is given in `cov_mat`. The estimated standard error for $\hat{\beta}_1$ is $\sqrt{2.7384} = 1.6548$ and for $\hat{\beta}_2$ it is $\sqrt{0.1597} = 0.3997$.

An approximate 95% Wald confidence interval for β_2 is

$$-1.1093 \pm 1.96(0.3997) = (-1.8927, -0.3259).$$

Interpretation: for every 1 unit increase in $\log_{10}$(initial WBC count) the average survival time multiplies by $\exp(\hat{\beta}_2) = 0.3298$.

Instead of creating our own function to fit the model, we could have used R's `glm()` function.

```
mod <- glm(surv_weeks ~ log_wbc, family = Gamma(link = "log"), data = dat)
summary(mod)
```

```
##
## Call:
## glm(formula = surv_weeks ~ log_wbc, family = Gamma(link = "log"),
##     data = dat)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9922  -1.2102  -0.2242   0.2102   1.5646
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   8.4775     1.6034   5.287 9.13e-05 ***
## log_wbc      -1.1093     0.3872  -2.865   0.0118 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 0.9388638)
##
##     Null deviance: 26.282  on 16  degrees of freedom
## Residual deviance: 19.457  on 15  degrees of freedom
## AIC: 173.97
##
## Number of Fisher Scoring iterations: 8
```

Observe that the estimated standard errors of $\hat{\beta}_1$ and $\hat{\beta}_2$ differ from what we obtained from our function `irls_exp()`. This is because in the call to `glm()` the `family = Gamma()` component fits the more general gamma distribution rather than the exponential distribution. The gamma distribution is parameterized by two parameters (mean and dispersion), which is why we see the line "Dispersion parameter for Gamma family taken to be 0.9388638". Since the exponential distribution is parameterized by a single parameter (mean), the dispersion is not estimated: it is just 1.

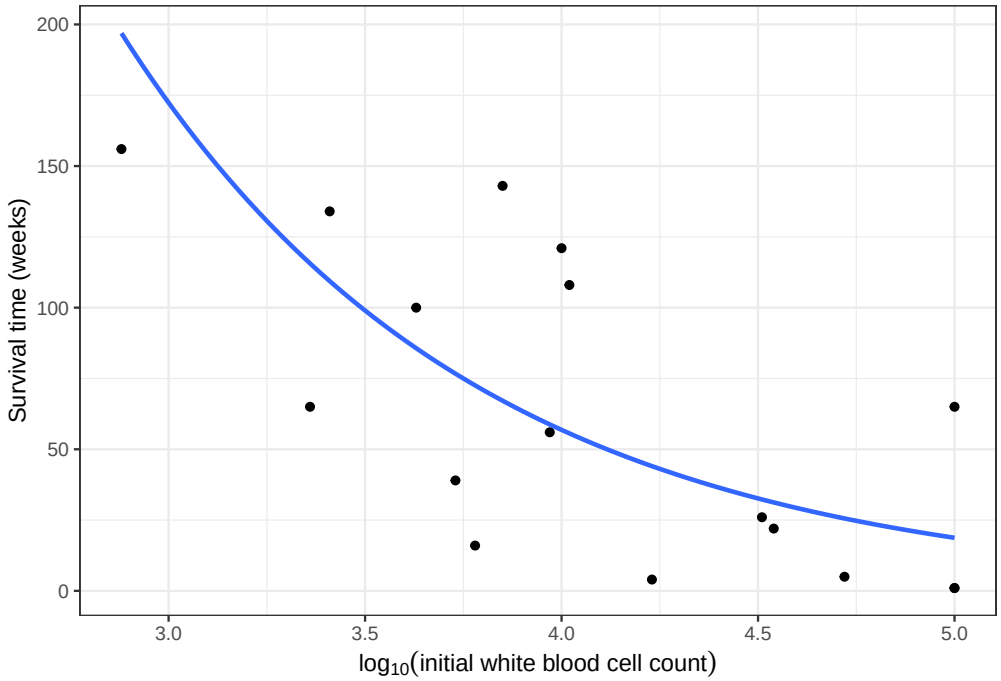
If we specify `dispersion = 1` in the call to `summary()`, then the standard error estimates will correspond to the values we obtained from `irls_exp()`.

```
summary(mod, dispersion = 1)
```

```
##
## Call:
## glm(formula = surv_weeks ~ log_wbc, family = Gamma(link = "log"),
##     data = dat)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9922  -1.2102  -0.2242   0.2102   1.5646
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   8.4775     1.6548   5.123 3.01e-07 ***
## log_wbc      -1.1093     0.3997  -2.776  0.00551 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 1)
##
##     Null deviance: 26.282  on 16  degrees of freedom
## Residual deviance: 19.457  on 15  degrees of freedom
## AIC: 173.97
##
## Number of Fisher Scoring iterations: 8
```

We can plot the fitted curve for our model to the data.

```
ggplot(dat, aes(log_wbc, surv_weeks)) +
  geom_point() +
  geom_smooth(method = "glm", formula = y ~ x, se = FALSE,
    method.args = list(family = Gamma(link = "log"))) +
  xlab(expression(log[10]("initial white blood cell count"))) +
  ylab("Survival time (weeks)")
```



To assess the adequacy of the model fit, we can compare the observed values y_i and the fitted values $\hat{y}_i = \hat{\mu}_i = \exp(\hat{\beta}_1 + \hat{\beta}_2 x_i)$ using standardized residuals, denoted r_i.

$$
\begin{aligned}
r_i &= \frac{(y_i - \hat{y}_i)}{\hat{\sigma}_i} \\
&= \frac{(y_i - \hat{y}_i)}{\sqrt{\text{Var}(Y_i|\mathbf{x}_i)}} \\
&= \frac{(y_i - \hat{y}_i)}{\sqrt{\frac{1}{\hat{\lambda}_i^2}}} \\
&= \frac{(y_i - \hat{y}_i)}{\frac{1}{\hat{\lambda}_i}} \\
&= \frac{(y_i - \hat{y}_i)}{\hat{y}_i}.
\end{aligned}
$$

In the plot of the standardized residuals versus the fitted values, observe that all but one of the residuals are relatively close to 0. Thus, the model appears to fit the data well. The one outlying residual at $r = 2.467$ corresponds to the observation $(y = 65, x = 5)$.

```
r <- (dat$surv_weeks - mod$fitted.values) / mod$fitted.values

# plot of standardized residuals vs. fitted values
ggplot(data.frame(fitted_values = mod$fitted.values, residual = r),
  aes(fitted_values, residual)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed") +
  ylim(c(-2.75, 2.75)) +
  xlab("Fitted value") +
  ylab("Residual")
```