UNIVERSITÀ
DEGLI STUDI DI TRIESTE

Dipartimento di scienze economiche, aziendali, matematiche e statistiche
“Bruno de Finetti”

Bayesian Statistics

Generalized linear regression

Leonardo Egidi

A.A. 2022/23

Indice

1 Motivations and theory

2 Logistic regression

3 Probit regression

4 Discrete data regression

Motivations

The purpose of generalized linear models is to extend the idea of linear modelling to cases for which the linear relationship between X and $\mathrm{E}(y|X)$ or the normal distribution for each y is not appropriate, even after any transformation of the data.

Example: when y is discrete, for instance the number of phone calls received by a person in one hour. The mean of y may be linearly related to X, but the variation term cannot be described by the normal distribution.

We review generalized linear models from a Bayesian perspective, although this class of models may be usefully applied from a classical perspective too.

Motivations

Given a $n \times p$ predictor matrix X and a parameters vector $\beta = (\beta_1, \dots, \beta_p)^T$, a generalized linear model is specified in three stages:

1. The linear predictor, $\eta = X\beta$.
2. The link function $g(\cdot)$, twice differentiable, that relates the linear predictor to the mean of the outcome variable, μ:

$$g(\mu) = \eta \rightarrow g^{-1}(\eta) = \mu.$$

3. The random component specifying the distribution of the outcome variable y with mean $\mathrm{E}(y|X) = \mu = g^{-1}(X\beta)$. The distribution can also depend on a *dispersion parameter* ϕ.

Dispersion exponential family of distributions

The third stage is the most important in terms of statistical interpretation. In the linear regression we assume that $y_i \sim \mathcal{N}(\mu, \sigma^2)$, where $\mu = \eta = X\beta$. We say that y_i belongs to the dispersion exponential family of probability distributions:

$$y_i \sim \mathrm{EF}(b(\theta), \phi/\omega) \tag{1}$$

if the single y_i has probability density function (pdf):

$$p(y|\theta, \omega) = \exp\left(\frac{\omega}{\phi}(y\theta - b(\theta)) + c(y, \phi)\right), \tag{2}$$

where θ and ϕ are unknown parameters, ω is a known scalar, and $b(\cdot), c(\cdot)$ are known functions that characterize the particular distribution within the class.

Dispersion exponential family of distributions

The distributions that belong to the EF family of distributions satisfy the following relations:

$$\begin{aligned} \mathrm{E}(y) &= b'(\theta) \\ \mathrm{Var}(y) &= \phi b''(\theta)/\omega, \end{aligned} \tag{3}$$

where $V(\mu) \equiv b''(\theta)$ is known as *variance function*. We may rewrite the first equation as:

$$\mathrm{E}(y) \equiv \mu \equiv g^{-1}(X\beta) = b'(\theta) \tag{4}$$

It is easy to prove that the normal, the Poisson and the binomial distribution belong to the EF family.

Dispersion exponential family of distributions: Poisson

If $y_i \sim \text{Pois}(\lambda_i),\ i = 1, \ldots, n$, then:

$$\begin{aligned} p(y_i|\lambda_i) &= e^{-\lambda_i}\frac{\lambda_i^{y_i}}{y_i!} \\ &= \exp\left(y_i \log \lambda_i - \lambda_i - \log(y_i!)\right), \end{aligned}$$

where $\theta_i = \log(\lambda_i)$, $b(\theta_i) = \lambda_i = e^{\theta_i}$, $c(y_i, \phi) = \log(y_i!)$, $\phi = \omega = 1$. Thus:

$$\begin{aligned} \mathrm{E}(y_i) =& b'(\theta_i) = \frac{de^{\theta_i}}{d\theta_i} = e^{\theta_i} = \lambda_i \\ \mathrm{Var}(y_i) =& \phi b''(\theta_i)/\omega = \frac{d^2 e^{\theta_i}}{d(\theta_i)^2} = e^{\theta_i} = \lambda_i \end{aligned}$$

Dispersion exponential family of distributions: Normal

If $y_i \sim \mathcal{N}(\mu_i, \sigma^2),\ i = 1, \ldots, n$, then:

$$
\begin{aligned}
p(y_i|\mu_i, \sigma^2) &= (2\pi\sigma)^{-1/2} e^{-\frac{1}{2\sigma^2}(y_i-\mu_i)^2} \\
&= (2\pi\sigma)^{-1/2} \exp\left(-\frac{1}{2\sigma^2}(y_i^2 - 2y_i\mu_i + \mu_i^2)\right) \\
&= \exp\left(\frac{1}{2\sigma^2}(2y_i\mu_i - \mu_i^2) - \frac{1}{2}\log(2\pi\sigma) - \frac{1}{2\sigma^2}y_i^2\right)
\end{aligned}
$$

where $\theta_i = \mu_i$, $b(\theta_i) = \mu_i^2/2 = \theta_i^2$, $c(y_i, \phi) = \frac{1}{2}\log(2\pi\sigma) - \frac{1}{2\sigma^2}y_i^2$, $\phi = \sigma^2$, $\omega = 1$. Thus:

$$
\begin{aligned}
\mathrm{E}(y_i) =& b'(\theta_i) = \frac{d\theta_i^2/2}{d\theta_i} = 2\theta_i/2 = \mu_i \\
\mathrm{Var}(y_i) =& \sigma^2 b''(\theta_i)/\omega = \frac{d^2{\theta_i}^2/2}{d(\theta_i)^2} = \sigma^2
\end{aligned}
$$

Canonical link function

The link function $g(\cdot)$ has not particular restrictions, usually $g:(a,b)\to(-\infty,+\infty)$, with a and b the lower and the upper bound of the support of μ_i, respectively. However, there is an easy choice for g, called canonical link function, such that

$$g(\mu_i)\equiv\eta_i=\theta_i. \tag{5}$$

In the Poisson case:

$$g(\mu_i)=\theta_i\Leftrightarrow g(b'(\theta_i))=\theta_i\Leftrightarrow g(e^{\theta_i})=\theta_i\Leftrightarrow g(\cdot)=\log(\cdot),$$

the link function is the logarithm. In the normal case is the identity function, in the binomial is the logit function. (See next table for a summary of three distributions belonging to the EF family. Careful! The list is not exaustive...)

Canonical link function

Notation	Bin(n, p)	Pois(λ)	$\mathcal{N}(\mu, \sigma^2)$
Range of y	$\mathbb{N}$	$\mathbb{N}$	$\mathbb{R}$
Dispersion parameter: ϕ	1	1	σ^2
Cumulant function: $b(\theta)$	$nlog(1+e^\theta)$	e^θ	$\frac{\theta^2}{2}$
$c(y;\phi)$	$log\binom{n}{y}$	$-logy!$	$-\frac{1}{2}(\frac{y^2}{\sigma^2}+\log 2\pi\sigma^2)$
$\mu(\theta)$	$n\frac{e^\theta}{1+e^\theta}$	e^θ	θ
Variance function: $V(\mu)$	$n\mu(1-\mu)$	μ	1
Canonical link function:	logit	logarithm	identity

Table: Characteristics of some common univariate distributions in the dispersion exponential family.

Overdispersion, offsets

GLM represent a wide class of models allowing for modelling:

- overdispersion, the possibility of variation beyond that of the assumed sampling distribution.

  Example The proportion of democrat voters in North Carolina is assumed to be binomial with some explanatory variables (such as voters' age, sex, and so forth). The data might indicate more variation than expected under the binomial model, $\mathrm{Var}(y) > np(1-p)$.

- offsets, the possibility to include in the linear predictor η a known coefficient, able to take care of different exposures.

  Example The number of car accidents is assumed to follow a Poisson distribution with rate λ with some explanatory variables. The rate of occurrence is λ per units of time, so that with exposure T the expected number of accidents is λT, where T represents the vector of exposure times for each unit.

Bayesian inference and GLMs

We consider GLMs with noninformative and informative prior distributions on regression parameters β, similarly as what we have done for linear models. A prior distribution can be placed on the dispersion parameter ψ as well, and any prior information about β can be described conditional on ϕ, that is $p(\beta, \phi) = p(\beta|\phi)p(\phi)$.

As in LMs, the classical analysis of GLMs is obtained if a noninformative or flat prior distribution is assumed for β: the posterior mode corresponding to a noninformative uniform prior density is the maximum likelihood estimate for β.

Posterior inference in GLMs typically will require the *approximation* and sampling tools like Markov Chain Monte Carlo (MCMC). We will generally use Stan (`rstan` and `rstanarm` packages) to sample from their posterior distributions.

Indice

1 Motivations and theory

2 Logistic regression

3 Probit regression

4 Discrete data regression

Logistic regression

Logistic regression is the standard way to model binary outcomes (that is, data y_i that take on the values 0 or 1).

We model the probability that the single $y_i = 1$:

$$p_i \equiv \Pr(y_i = 1) = \mathrm{logit}^{-1}(x_i\beta), \tag{6}$$

where $\eta_i = x_i\beta$ is the linear predictor, and the logit function is expressed as:

$$\mathrm{logit}(p_i) \equiv \log \frac{p_i}{1 - p_i} = x_i\beta \tag{7}$$

It is easy to check that $\mathrm{logit}^{-1}(\eta_i) = \frac{e^{\eta_i}}{1+e^{\eta_i}}$.

Logistic regression - Interpreting the coefficients

Coefficients in logistic regression can be challenging to interpret because of the nonlinearity just noted.

To understand better, let's fit a simple model about some political US polls in 1992.

1992 polls

Conservative parties generally receive more support among voters with higher incomes. We use this pattern from the National Election Study in 1992. For each respondent i in this poll, we label $y_i = 1$ if he/she preferred Bush (the Republican candidate), or 0 if he/she preferred Bill Clinton (Democrate candidate). We predict preferences given the respondent's income level (our x), which is characterized on a five-points scale. $n = 1179$ respondents.

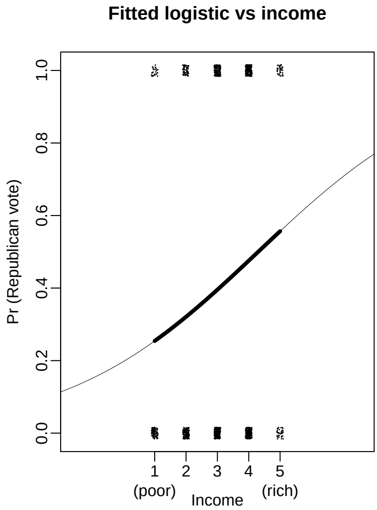
Logistic regression - Interpreting the coefficients

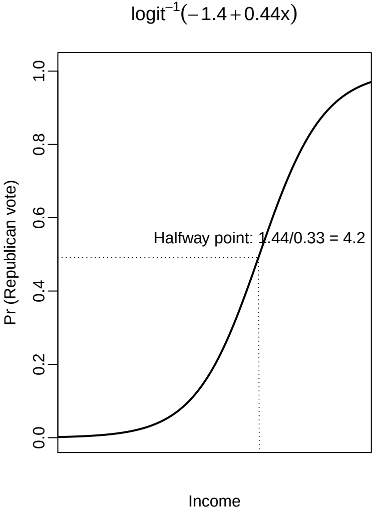
Let's fit the model in the classical way:

```
glm(formula = vote ~ income,
               family = binomial(link = "logit"))
            coef.est coef.se
(Intercept) -1.40     0.19
income       0.33     0.06
---
  n = 1179, k = 2
```

Thus, the fitted model is $\mathrm{Pr}(y_i = 1) = \mathrm{logit}^{-1}(-1.40 + 0.33\mathrm{income}_i)$.

Logistic regression - Interpreting the coefficients





Logistic regression - Interpreting the coefficients

- As with linear regression, the intercept can only be interpreted assuming zero values for the other predictors. When zero is not interesting or not even in the model (as in this case), we may evaluate Pr(Bush support) at the mean of respondents' incomes, $\bar{x}$, $\mathrm{logit}^{-1}(-1.40 + 0.33\bar{x}) = 0.4$.
- A difference of 1 in outcome (on this 1-5 scale) corresponds to a positive difference of 0.33 in the logit probability of supporting Bush.
  - $\mathrm{logit}^{-1}(-1.40 + 0.33 \times 3) - \mathrm{logit}^{-1}(-1.40 + 0.33 \times 2) = 0.08$. A difference of 1 in income category corresponds to a positive difference of 8% in the probability of supporting Bush.
  - consider the derivative of the logistic curve at $\bar{x} = 3.1$, this is: $\beta e^{\bar{\eta}}/(1 + e^{\bar{\eta}})^2$. Thus, the change in $\mathrm{Pr}(y_i = 1)$ per small unit of change in x at the mean value is $0.33e^{-0.39}/(1 + e^{-0.39})^2 = 0.13$.
  - *divide by 4 rule*: $\beta/4 = \beta e^0/(1 + e^0)^2 = 0.08$. As a rule of convenience, we can divide coreefficients by 4 to get an upper bound of the predictive difference corrpesponding to a change of 1 in x.

Logistic regression - Interpreting the coefficients

- There is another popular way to interpret the logistic regression coefficients, in terms of *odds ratios*.
- If two outcomes have the probabilities $(p, 1-p)$, $p/(1-p)$ is called the odds. An odds of 1 is equivalent to a probability of 0.5, that is, equally likely outcomes.
- Taking the logarithm of the odds ratio yields the log odds ratios, in our previous example with one predictor:

$$\log \frac{p_i}{1-p_i} = \alpha + \beta \mathrm{income}_i \tag{8}$$

Adding 1 to x in the equation above has the effect of adding β to both sides of the equation. A units difference in x corresponds to a multiplicative change of $e^{0.33} = 1.39$ in the odds.

Logistic regression: Stan model (1992polls.stan)

```
data{
  int N;          // number of voters
  int vote[N];    // vote: 0 (Clinton), 1 (Bush)
  int income[N];  // 1-5 income scale
}
parameters{
  real alpha;    // intercept
  real beta;     // income coefficient
}
model{
  for (n in 1:N){
    vote[n] ~ bernoulli_logit(alpha+income[n]*beta);
                // likelihood
  }
  alpha ~ normal(0, 10);  // intercept weakly-inf prior
  beta ~ normal(0, 2.5);  // income weakly-inf prior
}
```

Logistic regression: Bayesian estimation

Let's fit now the same model under the Bayesian approach, first of all with noninformative priors, using the `stan_glm` function in the `rstanarm` package, $\alpha \sim \mathcal{N}(0, 100^2)$, $\beta \sim \mathcal{N}(0, 100^2)$:

```
fit.2 <-  stan_glm (vote ~ income,
                    family=binomial(link="logit"),
                    prior=normal(0, 100),
                    prior_intercept=normal(0,100))
print(fit.2)
            Median MAD_SD
(Intercept) -1.4    0.2
income       0.3    0.1
```

The estimates are the same as those obtained from the `glm` function.

Logistic regression: Bayesian estimation

We use now some weakly informative priors, $\alpha \sim \mathcal{N}(0, 10^2)$, $\beta \sim \mathcal{N}(0, 2.5^2)$:

```
fit.3 <-  stan_glm (vote ~ income,
                    family=binomial(link="logit"),
                    prior=normal(0, 2.5),
                    prior_intercept=normal(0,10))
print(fit.3)
            Median MAD_SD
(Intercept) -1.4    0.2
income       0.3    0.1
```

The estimates are the same as those obtained from previous analysis. This means that we have enough observations to weaken the role of the prior distribution.

Logistic regression - Role of the prior

Thus, one may ask him(her)self: what is the advantage to use the Bayesian approach in place of the classical approach, given that the final results coincide?

The Bayesian juggler (analyst) may enjoy more! The prior is *part* of the model and its role may be very useful (to be continued).

LIKELIHOOD
PRIOR
LIKELIHOOD

(a) Classical juggler

(b) Bayesian juggler

Logistic regression - Separation

Nonidentifiability is a common problem in logistic regression. In addition to the problem of collinearity, familiar from linear regression, discrete-data regression can also become unstable from complete separation, which arises when a linear combination of the predictors is perfectly predictive of the outcome.

A common solution to separation is to remove predictors until the resulting model is identifiable, which typically results in removing the strongest predictors from the model.

An alternative approach to obtain stable logistic regression coefficients is to use Bayesian inference: precisely, suitable prior distributions on β.

Logistic regression

Consider to simulate $n = 100$ data $y_i \sim \text{Bernoulli}(p_i)$, where $\text{logit}(p_i) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2}$, $\beta_0 = 1, \beta_1 = 1.5, \beta_2 = 2$, and we draw $x_1 \sim \mathcal{N}(0,1), x_2 \sim \text{Bin}(n, 0.5)$.

We fit now a simple logistic regression for y using the `glm` function and the `stan_glm` contained in the R package `rstanarm`.

The idea is to compare the estimates of a bunch of simulated datasets under the classical and the Bayesian approach.

- Dataset 1: no separation.
- Dataset 2: separation ($y = 1 \Leftrightarrow x_2 = 1$).

Logistic regression: dataset 1. Classical vs noninformative

```
# classical
glm(formula = y ~ x1 + x2, family = binomial(link = "logit"))
            coef.est coef.se
(Intercept) 1.08     0.37
x1          1.45     0.36
x2          1.88     0.65
---
  n = 100, k = 3

# noninformative prior
stan_glm (y ~ x1 + x2,
                family=binomial(link="logit"),
                prior=normal(0,100),
                prior_intercept = normal(0,100))
            Median MAD_SD
(Intercept) 0.9    0.3
x1          1.1    0.3
x2          2.1    0.6
```

Logistic regression: dataset 1. Noninf. vs weakly-inf.

```
# noninformative prior
stan_glm (y ~ x1 + x2,
                family=binomial(link="logit"),
                prior=normal(0,100),
                prior_intercept = normal(0,100))
            Median MAD_SD
(Intercept) 0.9    0.3
x1          1.1    0.3
x2          2.1    0.6

# weakly-informative priors (normal(0,10^2) and normal(0,2.5^2))
stan_glm (y ~ x1 + x2,
                family=binomial(link="logit"))
            Median MAD_SD
(Intercept) 0.9    0.3
x1          1.1    0.3
x2          1.9    0.6
```

Logistic regression: dataset 2. Classical vs noninformative

```
# classical
glm(formula = y ~ x1 + x2, family = binomial(link = "logit"))
            coef.est coef.se
(Intercept)    0.91     0.36
x1             1.26     0.43
x2            20.15  2370.96
---
  n = 100, k = 3

# noninformative priors (normal(0,100^2))
stan_glm (y ~ x1 + x2,
                family=binomial(link="logit"),
                prior=normal(0,100),
                prior_intercept=normal(0,100))
            Median MAD_SD
(Intercept)  1.0    0.4
x1           1.3    0.5
x2          62.2   51.6
```

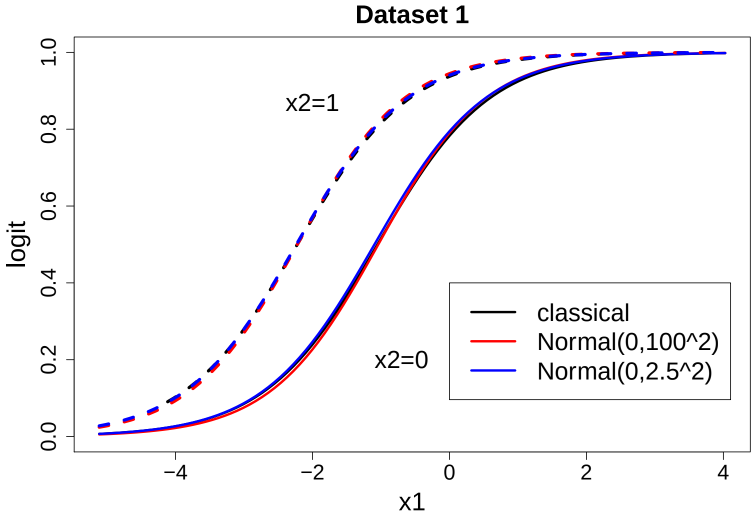
Logistic regression: dataset 2. Noninf. vs weakly-inf.

```
# noninformative priors (normal(0,100^2))
stan_glm (y ~ x1 + x2,
                family=binomial(link="logit"),
                prior=normal(0,100),
                prior_intercept=normal(0,100))
            Median MAD_SD
(Intercept)  1.0    0.4
x1           1.3    0.5
x2          62.2   51.6

# weakly-informative priors (normal(0,10^2) and normal(0,2.5^2))
stan_glm (y ~ x1 + x2,
                family=binomial(link="logit"))
           Median MAD_SD
(Intercept) 1.0    0.4
x1          1.2    0.4
x2          4.3    1.2
```

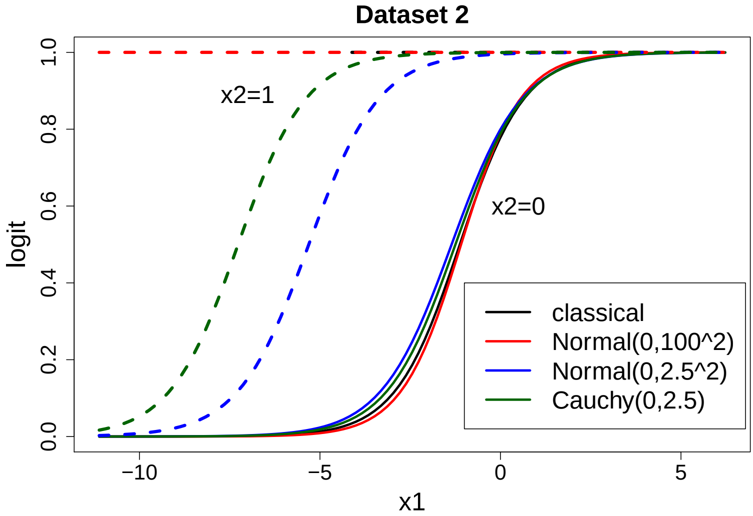
Logistic regression: dataset 2. weakly-inf. vs weakly-inf.

```
# weakly-informative priors (normal(0,10^2) and normal(0,2.5^2))
stan_glm (y ~ x1 + x2,
                family=binomial(link="logit"))
           Median MAD_SD
(Intercept) 1.0    0.4
x1          1.2    0.4
x2          4.3    1.2

# weakly-informative priors (cauchy(0,10^2) and cauchy(0,2.5^2))
stan_glm (y ~ x1 + x2,
                family=binomial(link="logit"),
                prior=cauchy(0,2.5),
                prior_intercept = cauchy(0,10))
            Median MAD_SD
(Intercept) 0.9    0.4
x1          1.2    0.4
x2          6.4    3.2
```

Logistic regression: dataset 1



Logistic regression: dataset 2



Logistic regression: stable estimates

Comments:

- Weakly informative priors allow to obtain stable logistic regression coefficients.
- Noninformative priors do not solve separation.
- Prior choice is a fundamental *part* of our models, especially as the complexity grows.

Indice

1 Motivations and theory

2 Logistic regression

3 Probit regression

4 Discrete data regression

Probit regression

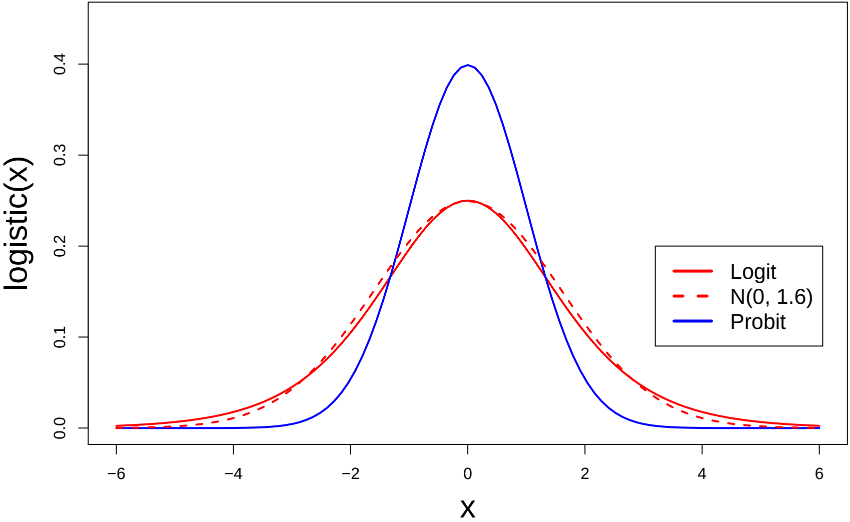
The probit model is the same as the logit, except it replaces the logistic by the normal distribution. We can write the model directly as

$$\Pr(y_i = 1) = \Phi(x_i \beta), \tag{9}$$

where Φ is the standard normal cumulative distribution.

As shown in the next plot, the probit model is close to the logit model with the residual standard deviation set to 1.6 rather than 1. As a result, coefficients in a probit regression are typically close to logistic regression coefficients divided by 1.6.

Probit regression



Probit regression: 1992 polls

We estimate the conservative support for the 1992 US elections, but this time with probit regression:

```
fit.4 <- stan_glm (vote ~ income,
                   family=binomial(link="probit"),
                   prior=normal(0, 2.5),
                   prior_intercept=normal(0,10))
print(fit.4)
            Median MAD_SD
(Intercept) -0.9    0.1
income       0.2    0.0
```

Rule of thumb: $-0.89 \approx -1.40/1.6$, and $0.2 \approx 0.33/1.6$. (Red: logistic coefficients)

Indice

1. Motivations and theory
2. Logistic regression
3. Probit regression
4. Discrete data regression

Discrete data regression: cockroaches data

Cockroaches data

A company that owns many residential buildings throughout New York City tells that they are concerned about the number of cockroach complaints that they receive from their 10 buildings. They provide you some data collected in an entire year for each of the buildings and ask you to build a model for predicting the number of complaints over the next months.

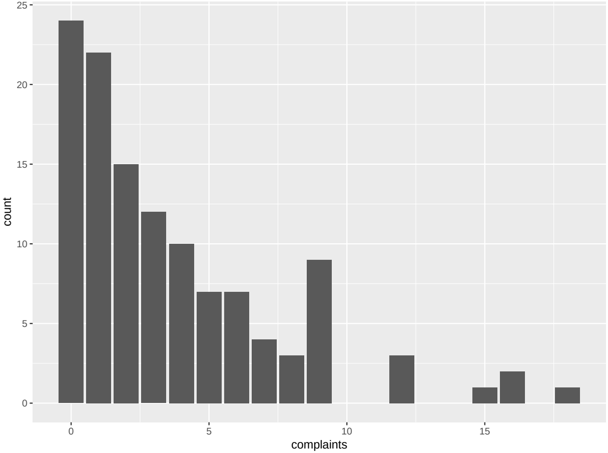
Discrete data regression: cockroaches data

We have access to the following fields (`pest_data.RDS`):

- `complaints`: Number of complaints per building in the current month
- `traps`: The number of traps used per month per building
- `live_in_super`: An indicator for whether the building has a live-in super
- `age_of_building`: The age of the building
- `total_sq_foot`: The total square footage of the building
- `average_tenant_age`: The average age of the tenants per building
- `monthly_average_rent`: The average monthly rent per building
- `floors`: The number of floors per building

Discrete data regression: cockroaches data

Let's make some plots of the raw data, such as the distribution of the complaints:



Poisson regression: cockroaches data

A common way of modeling this sort of skewed, single bounded count data is as a Poisson random variable. For simplicity, we will start assuming:

- ungrouped data, with no building distinction
- no time-trend structures

We use the number bait stations placed in the building, denoted below as `traps`, as explanatory variable. This model assumes that the mean and variance of the outcome variable `complaints` (number of complaints) is the same. For the i-th complaint, $i = 1, \ldots, n$, we have

$$\begin{aligned} \text{complaints}_i &\sim \text{Poisson}(\lambda_i) \\ \lambda_i &= \exp\left(\eta_i\right) \\ \eta_i &= \alpha + \beta \, \text{traps}_i \end{aligned}$$

Poisson regression: cockroaches data

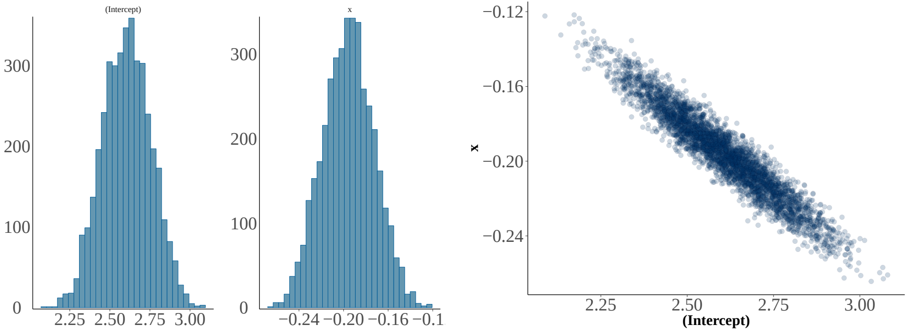
Let's fit this simple model via the stan_glm function of the rstanarm package:

```
y <- pest_data$complaints
x <- pest_data$traps
M_pois <- stan_glm(y~x, family=poisson(link="log"))
print(M_pois)
```

```
            Median MAD_SD
(Intercept)  2.6    0.2
x           -0.2    0.0
```

Poisson regression: cockroaches. Posterior plots

Let's have a glimpse of simulated posterior distributions for α and β:



As we expected, it appears the number of bait stations set in a building is associated with the number of complaints about cockroaches that were made in the following month.

Poisson regression: cockroaches. Overdispersion

Comments:

- Taking the posterior means of the parameters as point estimates, a building with $\bar{x} = 7$ traps will have a predicted average amounting at:

$$\lambda = \exp(2.61 - 0.2\bar{x}) \approx 3.35$$

- Under this model, $\mathrm{E}(\mathrm{complaints}) = \mathrm{Var}(\mathrm{complaints}) \approx 3.35$.
- However, the raw mean of the data is 3.66 and its variance is 14.9...maybe the Poisson model is not well suited for this dataset? There is much overdispersion.

Poisson regression: cockroaches. Extending the model

Modelling the relationship between complaints and bait stations is the simplest model. However, we can expand the model.

Currently, our model's mean parameter is a rate of complaints per 30 days, but we're modelling a process that occurs over an area as well as over time. We have the square footage of each building, so if we add that information into the model, we can interpret our parameters as a rate of complaints per square foot per 30 days. For the *i*-th complaint, we assume:

$$\begin{aligned} \text{complaints}_i &\sim \text{Poisson}(\text{sq\_foot}_i \, \lambda_i) \\ \lambda_i &= \exp\left(\eta_i\right) \\ \eta_i &= \alpha + \beta \, \text{traps}_i \end{aligned}$$

Poisson regression: cockroaches. Offset term

The term sq_foot is called an exposure term. If we log the term, we can put it in η_i:

$$\begin{aligned}
\text{complaints}_i &\sim \text{Poisson}(\lambda_i)\\
\lambda_i &= \exp(\eta_i)\\
\eta_i &= \alpha + \beta\,\text{traps}_i + \text{log\_sq\_foot}_i
\end{aligned}$$

```
exposure <- log(pest_data$total_sq_foot/1e4)
M_pois_exposure <- stan_glm(y~x+offset(exposure),
                            family=poisson(link="log"))
print(M_pois_exposure)
            Median MAD_SD
(Intercept)  0.8    0.2
x           -0.2    0.0
```

Poisson regression: cockroaches. Offset term

Comments:

- Let's compute now a naive estimates for λ using the posterior estimates, considering a building with $\bar{x} = 7$ and exposure equal to 1.77:

$$\lambda = \exp(0.8 - 0.2 \times \bar{x} + \mathrm{log\_sq\_foot}) \approx 3.22$$

- This again looks like we haven't captured the smaller counts very well, nor have we captured the larger counts. We need something different to model the overdispersion.

Poisson regression: cockroaches. Overdispersion

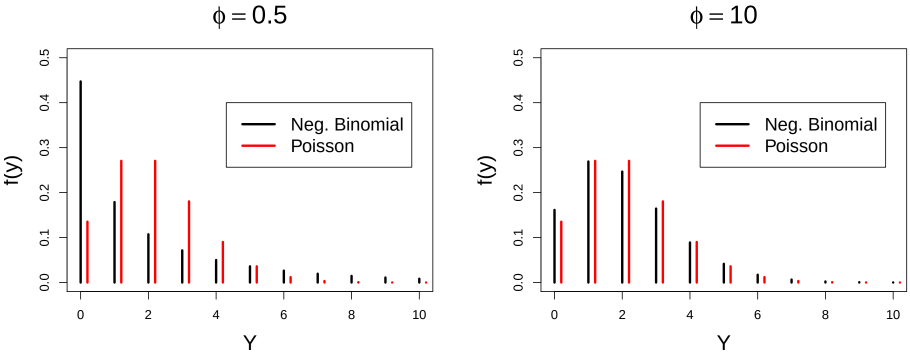
A possible drawback of the Poisson distribution is that the mean coincides with the variance. It may be not well suited when data reveals much more variation than that assumed by the Poisson distribution

Negative binomial If $Y \sim$ Neg-Binomial(λ, ϕ), where λ has the same meaning as before and ϕ is the *dispersion parameter*, we have;

$$\begin{aligned} \mathrm{E}(Y) =& \lambda \\ \mathrm{Var}(Y) =& \lambda + \lambda^2/\phi. \end{aligned}$$

The variance grows as the dispersion parameter ϕ tends to 0. As $\phi \to \infty$, the two distributions coincide.

Poisson vs Negative binomial: $\lambda = 2$.



Negative binomial regression: cockroaches. Overdispersion

Thus, we assume the following model to allow for overdispersion:

$$
\begin{aligned}
\text{complaints}_i &\sim \text{Neg-Binomial}(\lambda_i, \phi)\\
\lambda_i &= \exp(\eta_i)\\
\eta_i &= \alpha + \beta\, \mathrm{traps}_i
\end{aligned}
$$

```
M_negbin <- stan_glm(y ~ x,
                     family =neg_binomial_2(link="log"))
print(M_negbin)
            Median MAD_SD
(Intercept)  2.7    0.4
x           -0.2    0.1
```

Negative binomial regression: cockroaches. Overdispersion

Comments:

- Taking again the posterior means of the parameters as point estimates, a building with $\bar{x} = 7$ traps will have a predicted average amounting at:

  $$\lambda = \exp(2.60 - 0.19\bar{x}) \approx 3.56,$$

  and the variance may be approximately computed as:

  $$\lambda + \lambda^2/\phi = 3.56 + (3.56)^2/3.3 = 7.4,$$

  that seems a more realistic assumption.
- A Poisson model doesn't fit over-dispersed count data very well because the same parameter λ controls both the expected counts and the variance of these counts.

Negative binomial regression: cockroaches. Overdisp.+offset

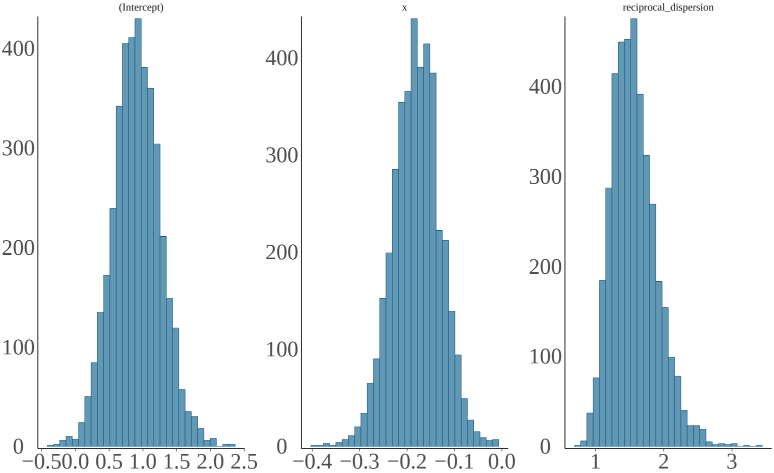
Let's consider now the exposure in the negative binomial model as well:

$$\begin{aligned}
\text{complaints}_i &\sim \text{Neg-Binomial}(\lambda_i, \phi)\\
\lambda_i &= \exp\left(\eta_i\right)\\
\eta_i &= \alpha + \beta\, \mathrm{traps}_i + \text{log\_sq\_foot}_i
\end{aligned}$$

```
M_negbin_exp <- stan_glm(y ~ x,
                            family =neg_binomial_2(link="log"),
                             offset=exposure)
print(M_negbin_exp)
            Median MAD_SD
(Intercept)  0.9    0.4
x           -0.2    0.1
```

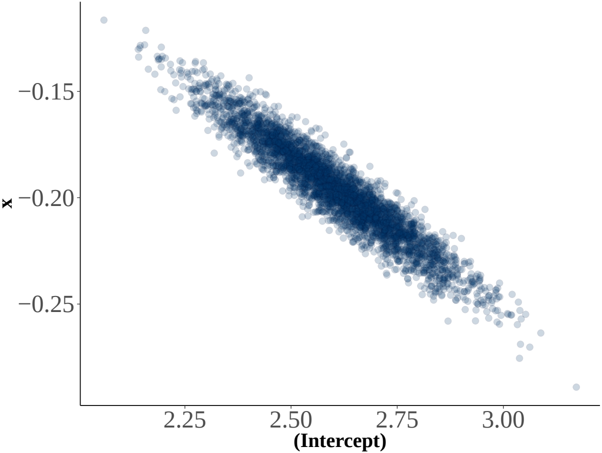
Negative binomial regression: cockroaches.

Let's take a look at the simulated posterior distribution for α, β and $1/\phi$.



Negative binomial regression: cockroaches.

Let's take a look at the scatterplot between α and β:



Negative binomial regression: cockroaches. Residuals

Comments:

- We had a glimpse that the negative binomial model outperforms the Poisson model when discrete data present much variation and heavy tails.

However:

- we should check the residuals , similarly as what we have done for the linear model.

Negative binomial regression: cockroaches. Residuals

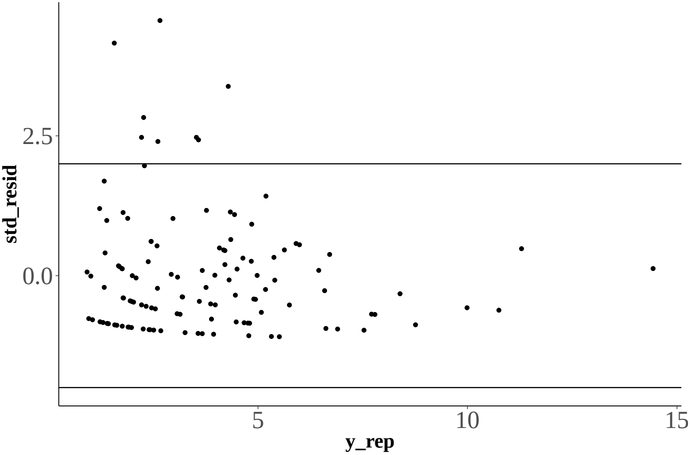
We need simulation:

- Generate `nsims` hypothetical samples y^{rep} from our model.
- Run `nsims` regression on each y^{rep}.
- Compute the standardized residuals as:

$$\frac{y - \tilde{\lambda}}{\sqrt{\tilde{\lambda} + \tilde{\lambda}^2/\tilde{\phi}}},$$

where $\tilde{\lambda}$ is the mean over the y^{rep} replications, and $\tilde{\phi}$ is the mean of the posterior estimates.

Negative binomial regression: cockroaches. Residuals



Negative binomial regression: cockroaches. Residuals

Comments:

- Looks ok, but we still have some very large standardized residuals. This might be because we are currently ignoring that the data are clustered by buildings, and that the probability of roach issue may vary substantially across buildings.
- It looks like we would need a sort of hierarchical structure: complaints within buildings. (to be continued...)
- Maybe ungrouped structure is poor here!

Further reading

Further reading:

- Chapter 16 from *Bayesian Data Analysis*, A. Gelman et al.

Weakly informative priors in logistic regression:

- Gelman, A., Jakulin, A., Pittau, M.G. and Su, Y-S. (2008). A weakly informative default prior distribution for logistic and other regression models. *Annals of Applied Statistics*, 2(4), 1360–1383. Here is the ▸ pdf.