Recitation 4

ABA Spring 2024
02/16/2024

Agenda

-GLM short review

-Poisson for count data

-Poisson Regression in Python

GLM

Assumptions of linear regression are too restrictive for many real-world prediction problems. Outcome can be:

- a category (cancer vs.healthy)
- count (number of children)
- the time to the occurrence of an event (time to failure of a machine) -
- a skewed outcome with a few very high values

This linear model can be extended to model all these types of outcomes. This extension is called Generalized Linear Models or GLMs for short.

It follows the same structure as a linear regression: $Y = X\beta + \varepsilon$

But we do not make the assumption that Y is normally distributed.

GLM: Components

1. **Distribution of Y:**

The distribution of the response variable given x is a member of the **exponential family** of distributions: normal (Gaussian), binomial, Poisson, $P(y,\eta) = b(y)\exp(\eta^T T(y) - \alpha(\eta))$

2. **Natural parameter** η **and covariates:**

η It is a linear weighted sum of the covariates $\eta = x'\beta$

3. **Link Function:**

Relates the expected value of the response variable E(Y) (i.e μ) to the linear combination of predictors $\eta = g(\mu)$

Invertible and one to one: $\mu = g^{-1}(\eta) = g^{-1}(x'\beta)$

Link functions

- Example of link functions

(Wikipedia)

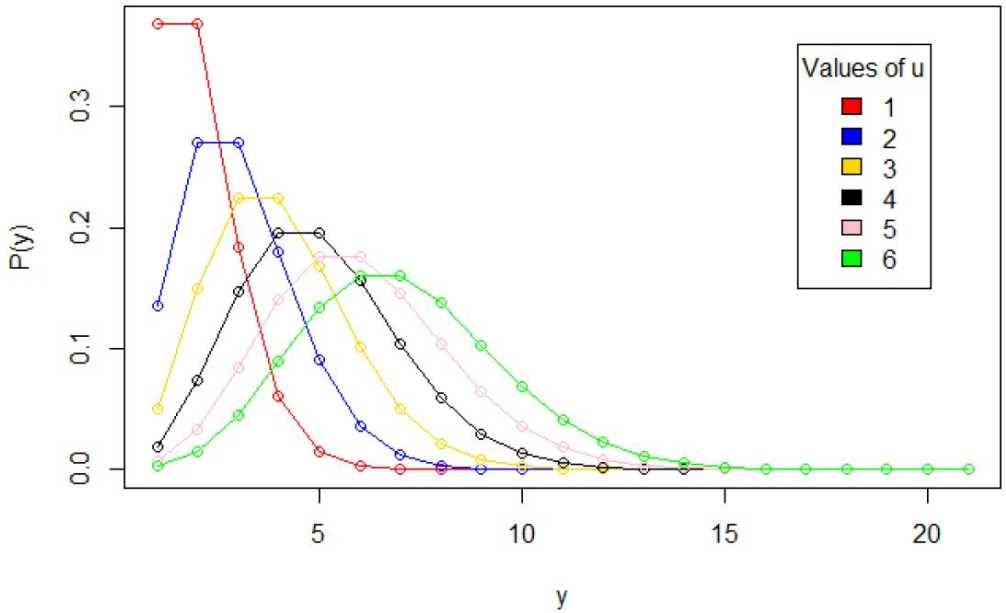
Common distributions with typical uses and canonical link functions

Distribution	Support of distribution	Typical uses	Link name	Link function, $\mathbf{X}\boldsymbol{\beta} = g(\mu)$	Mean function
Normal	real: $(-\infty, +\infty)$	Linear-response data	Identity	$\mathbf{X}\boldsymbol{\beta} = \mu$	$\mu = \mathbf{X}\boldsymbol{\beta}$
Exponential Gamma	real: $(0, +\infty)$	Exponential-response data, scale parameters	Negative inverse	$\mathbf{X}\boldsymbol{\beta} = -\mu^{-1}$	$\mu = -(\mathbf{X}\boldsymbol{\beta})^{-1}$
Inverse Gaussian	real: $(0, +\infty)$		Inverse squared	$\mathbf{X}\boldsymbol{\beta} = \mu^{-2}$	$\mu = (\mathbf{X}\boldsymbol{\beta})^{-1/2}$
Poisson	integer: $0, 1, 2, \ldots$	count of occurrences in fixed amount of time/space	Log	$\mathbf{X}\boldsymbol{\beta} = \ln(\mu)$	$\mu = \exp(\mathbf{X}\boldsymbol{\beta})$
Bernoulli	integer: $\{0, 1\}$	outcome of single yes/no occurrence	Logit	$\mathbf{X}\boldsymbol{\beta} = \ln\left(\frac{\mu}{1-\mu}\right)$	$\mu = \frac{\exp(\mathbf{X}\boldsymbol{\beta})}{1+\exp(\mathbf{X}\boldsymbol{\beta})} = \frac{1}{1+\exp(-\mathbf{X}\boldsymbol{\beta})}$
Binomial	integer: $0, 1, \ldots, N$	count of # of "yes" occurrences out of N yes/no occurrences		$\mathbf{X}\boldsymbol{\beta} = \ln\left(\frac{\mu}{n-\mu}\right)$	
Categorical	integer: $[0, K)$ K-vector of integer: $[0, 1]$, where exactly one element in the vector has the value 1	outcome of single K-way occurrence		$\mathbf{X}\boldsymbol{\beta} = \ln\left(\frac{\mu}{1-\mu}\right)$	
Multinomial	*K*-vector of integer: $[0, N]$	count of occurrences of different types (1 .. *K*) out of *N* total *K*-way occurrences			

We have count data, so we look at Poisson: $g(u) = \log(\mu) = X\beta$

Poisson Distribution

- Poisson distribution is a starting point for count data. Poisson distribution has a pdf $P(y) = \frac{e^{-\lambda}\lambda^{y}}{y!}$



Poisson

- A log(.) function is logical because log(.) ensures that it converts range $(-\infty, \infty)$ to $(0, \infty)$ which is outcome we observe. Thus, any prediction will be bounded in this range.
- It can be shows that link function g(.) = log(μ) satisfies the requirement for GLM for Poisson.

- Since μ = E(y) = λ (the mean of Poisson). This also means that
  - log(λ)= $\alpha + \beta_1 x_1$ (because our link function is log()
  - $\lambda = e^{(\alpha+\beta_1 x_1)}$
  - $\lambda = e^{\propto} e^{\beta_1 x_1}$
  - $\lambda = \lambda_0 e^{\beta_1 x_1}$
- λ_0 is the baseline Poisson mean and covariates shift the mean. A unit increase in covariate changes the mean λ by e^{β} unit.
- When there are no covariates, we simply estimate α and hence λ_0.

Problem overview

- Data: # affairs
- Plot (counts histogram)
- Dmatrices
    - To fit most of the models in statsmodels, we create two design matrices. The first one is a matrix of endogenous variable(s) (dependent variable). The second is a matrix of exogenous variable(s) (i.e. independent, regressor, etc.)
- Split into Train (fit) and test (for prediction)
- Evaluate predictions and assumptions

GLM and Python

There are two key libraries one should download:

```
import statsmodels.api as sm

from patsy import dmatrices
```

See Jupyter notebook