Probability models for discrete data

B.D. in Business Administration and Economics
Course in Quantitative Methods III

Rosario Barone
University of Rome "Tor Vergata"

rosario.barone@uniroma2.it

What are we going to study?

- Binary outcome models
- Multinomal outcome models
- Mixture Models
- Models of Count Data
- Panel data Models

Before starting...

We need an introduction on the basic probability models for categorical variables.

- Binary: "Agree/Disagree","Presence/Absence"
- Nominal: brand, preferred political party
- Ordinal: levels of agreement, degree
- Count: number of events, number of subjects, number of clicks

Note: for simplicity we will treat ordinal variables as nominal.

How do we proceed?

For each presented model, we will specify:

- probabilistic properties: model definition, support of the variable, number of unknown parameters, analytical form of the probability mass function (pmf), moments.
- statistical properties: Maximum Likelihood Estimator (MLE), Fisher Information and Standard Errors.

Continuous probability distributions

Let X be a random variable, such that $X \in \mathbb{R}$. The main function we should consider are:

- probability density function: $f(x)$, $f : \mathbb{R} \to \mathbb{R}^+$; $f(x) \geq 0$ and $\int_{\mathbb{R}} f(x)dx = 1$.
- probability distribution function: $P(X \leq x) = F(x)$, with:
  - $F(x) = \int_{-\infty}^{x} f(x)dx$
  - $F : \mathbb{R} \to [0, 1]$.
  - $\lim_{x \to -\infty} F(x) = 0$
  - $\lim_{x \to +\infty} F(x) = 1$
- quantile function, defined as $Q(F(x)) = F(x)^{-1}$, such that $Q : [0, 1] \to \mathbb{R}$.

Discrete probability distributions

Let X be a random variable, such that $X \in \mathbb{I}$. The main function we should consider are:

- probability mass function: $Pr(X = x)$, with $Pr : \mathbb{I} \rightarrow [0, 1]$, $Pr(X = x) \geq 0$ and $\sum_i Pr(X = x_i) = 1$.
- probability distribution function: $Pr(X \leq x) = F(x)$, with:
  - $F(x) = \sum_{i:x_i \leq x} Pr(X = x_i)$
  - $F : \mathbb{I} \rightarrow [0, 1]$.
  - $\lim_{x \rightarrow -\infty} F(x) = 0$
  - $\lim_{x \rightarrow +\infty} F(x) = 1$
  - $\lim_{x \rightarrow x_{0^+}} F(x) = F(x_0)$
- quantile function, defined as $Q(F(x)) = F(x)^{-1}$, such that $Q : [0, 1] \rightarrow \mathbb{I}$.

Moments of a distribution

- Continuous random variables
  - $E(X) = \mu = \int_{\mathbb{R}} f(x) x dx$
  - $V(X) = \sigma^2 = \int_{\mathbb{R}} f(x)(x-\mu)^2 dx$
- Discrete random variables
  - $\mathrm{E}(X) = \sum_{i=1}^{n} p_i \cdot x_i$
  - $\mathrm{V}(X) = \sum_{i=1}^{n} p_i \cdot (x_i - \mu)^2,$

Maximum Likelihood Inference

Let assume to observe an $i.i.d.$ sample $X_1, \ldots, X_n$, with $X_i \sim f_\theta(X)$ depending on the parameter (or vector of parameters) $\theta \in \Theta$, and $X \in \mathbb{R}$.

We define the likelihood function as

$$\mathcal{L}(\theta) = \prod_{i=1}^{n} f_\theta(X_i),$$

and the log-likelihood function as

$$\ell(\theta) = \sum_{i=1}^{n} log(f_\theta(X_i)).$$

Maximum Likelihood Estimation

The Maximum Likelihood Estimation (MLE) consists in maximizing the Likelihood function w.r.t. the unknown parameter θ.

- **Analytic** Differentiate the likelihood function with respect to the parameter vector and set the resulting gradient vector to zero. Solve the system of equations to find extrema. Take the second derivative to make sure that you have a maximum rather than a minimum. This method only works if there is an analytical solution.

Maximum Likelihood Estimation

- **Grid Search** If you know θ lies in a subspace of $\mathbb{R}$, do an exhaustive search over that region for the θ that produces the largest likelihood. In other words, try each possible value of θ and find $\hat{\theta}$, which is the θ that produces the largest likelihood. This is a good way of showing that you can find the maximum of the likelihood function by repeated approximation and iteration. However, it is not practical in most cases and becomes much more difficult when the parameter space is high dimensional (even just three-dimensional).
- **Numerical** This is the most common (when analytical solutions are unavailable). Basically, we give the computer a set of starting values θ^0 for che vector θ, and let a hill climbing algorithm (Newton-Raphson, BHHH, DFP, etc.) find the maximum $\hat{\theta}_{ML}$.

Fisher Information and Standard Errors

By CLT we know that $\sqrt{n}(\hat{\theta} - \theta) \xrightarrow{D} N(0, I(\theta)^{-1})$.

Therefore, the (asymptotic) standar error of an ML estimator, $\hat{\theta}_{ML}$, is calculated by the inverse of the Fisher Information matrix:

$$se(\theta) = \sqrt{[I(\theta)]^{-1}},$$

where the Fisher Information matrix is

$$I(\theta) = -\mathbb{E}\left[H(\theta)\right]$$

and the the Hessian matrix H, is

$$H(\theta) = \frac{\partial^2 \ell(\theta)}{\partial \theta \partial \theta'}.$$

Binary data: Bernoulli distribution

- $Y \sim \text{Ber}(p)$
- $\text{supp}\,(Y) = \{0, 1\}$
- One parameter: $p \in (0, 1)$.
- Probability mass function: $\Pr(Y = y) = p^y (1 - p)^{(1-y)}$.
- $E[Y] = p$.
- $V[Y] = p(1 - p)$.

Binary data: Bernoulli distribution

- Maximum Likelihood Estimation

$$\mathcal{L}(p) = \prod_{i=1}^{n} p^{y_i}(1-p)^{(1-y_i)}$$

$$\ell(p) = \log p \sum_{i=1}^{n} y_i + \log(1-p) \sum_{i=1}^{n} (1-y_i)$$

MLE: value of the unknown parameter p that satisfies the first derivative of the log-likelihood (score function) equal to zero

$$\frac{\partial \ell(p)}{\partial p} = \frac{\sum_{i=1}^{n} y_i}{p} - \frac{\sum_{i=1}^{n} (1-y_i)}{(1-p)} = 0.$$

Binary data: Bernoulli distribution

$$\sum_{i=1}^{n} y_i - \hat{p} \sum_{i=1}^{n} y_i = \hat{p} \sum_{i=1}^{n} (1 - y_i)$$

$$\hat{p} = \frac{1}{n} \sum_{i=1}^{n} y_i$$

$$\frac{\partial^2 \ell(p)}{\partial p^2} = -\frac{\sum_{i=1}^{n} y_i}{p^2} - \frac{\sum_{i=1}^{n} (1 - y_i)}{(1-p)^2}$$

Since $p \in [0, 1]$ and $y_i \in \{0, 1\}$, the second derivative of the log-likelihood (the derivative of the score function) is negative: $\hat{p}$ is the Maximulm Likelihood estimator.

Binary data: Binomial distribution

By defining $X = \sum_{i=1}^{n} Y_i$ where $Y_i \overset{\text{iid}}{\sim} Ber(p)$, we get:

- $X \sim \text{Bin}(n, p)$
- $\text{supp}(X) = \{0, 1\}$
- Two parameters:
  - $p \in (0, 1)$
  - $n \in \mathbb{N}$
- Probability mass function:
  - Let $x \in \{0, 1, \ldots, n\}$ represent the number of successes
  - $\Pr(X = x) = \binom{n}{x} p^x (1-p)^{(n-x)}$.
- $E[X] = np$.
- $V[X] = np(1-p)$.

R commands: Binomial distribution


```
dbinom(x, size, prob, log = FALSE)

pbinom(q, size, prob, lower.tail = TRUE, log.p =
FALSE)

qbinom(p, size, prob, lower.tail = TRUE, log.p =
FALSE)

rbinom(n, size, prob)
```

R commands: Bernoulli distribution


```
dbinom(x, size=1, prob, log = FALSE)

pbinom(q, size=1, prob, lower.tail = TRUE, log.p =
FALSE)

qbinom(p, size=1, prob, lower.tail = TRUE, log.p =
FALSE)

rbinom(n, size=1, prob)
```

Categorical data: Multinomial distribution

Let us consider a generalization of the Binomial distribution. By extending $\text{supp}(X) = \{1, \ldots, n\}$, $i \in \{1, \ldots, k\}$ with $\sum_{i=1}^{k} x_i = n$, we get:

- $X \sim \text{Multinomial}(\mathbf{p})$
- Three parameters:
  - $\mathbf{p} \in [0,1]^k$ event probabilities such that $\sum_{i=1}^{k} p_i = 1$
  - $k \in \mathbb{N}$ number of mutually exclusive events
  - $n \in \mathbb{N}$ number of trials
- Probability mass function:
  - $\Pr(X = \mathbf{x}) = \frac{n!}{x_1! \cdots x_k!} p_1^{x_1} \cdots p_k^{x_k}$
  - $\mathbf{x} = \{x_i : i = 1, \ldots, k\}$ and $\sum_{i=1}^{k} x_i = n$

Categorical data: Multinomial distribution

- Moments of the distributon:
  - $E[X] = (E(X_1), \ldots, E(X_k))$ and $E[X_i] = np_i$
  - $V[X] = (V(X_1), \ldots, V(X_k))$ and $V[X_i] = np_i(1 - p_i)$
  - $Cov[X_i, X_j] = -np_i p_j$ for $i \neq j$
- Maximum Likelihood Estimation of p (n and k are known):

$$\mathcal{L}(\mathbf{p}) = \frac{n!}{x_1! \cdots x_k!} p_1^{x_1} \cdots p_k^{x_k}$$

$$\mathcal{L}(\mathbf{p}) = n! \prod_{i=1}^{k} \frac{p_i^{x_i}}{x_i!}$$

Categorical data: Multinomial distribution

$$\ell(\mathbf{p}) = \log(n!) + \sum_{i=1}^{k} x_i \log(p_i) - \log(x_i!)$$

By model assumption $\ell(\mathbf{p})$ has to be maximized under the constraint $\sum_{i=1}^{k} p_i = 1$. Therefore, we define the Lagrangian function:

$$\ell(\mathbf{p}, \lambda) = \ell(\mathbf{p}) + \lambda(1 - \sum_{i=1}^{k} p_i)$$

$$\ell(\mathbf{p}, \lambda) = \log(n!) + \sum_{i=1}^{k} x_i \log(p_i) - \log(x_i!) + \lambda(1 - \sum_{i=1}^{k} p_i)$$

$$\frac{\ell(\mathbf{p}, \lambda)}{\partial p_i} = \frac{x_i}{p_i} - \lambda = 0$$

$$\hat{p}_i = \frac{x_i}{\lambda}$$

Categorical data: Multinomial distribution

$$\frac{\partial^2 \ell(\mathbf{p}, \lambda)}{\partial^2 p_i} = -\frac{x_i}{p_i^2}.$$

Since:

$$\sum_{i=1}^{k} p_i = \sum_{i=1}^{k} \frac{x_i}{\lambda}$$

$$1 = \frac{1}{\lambda} \sum_{i=1}^{k} x_i$$

$$n = \lambda.$$

Therefore, we get:

$$\hat{p}_i = \frac{x_i}{n}$$

$$\hat{\mathbf{p}} = \left(\frac{x_1}{n}, \ldots, \frac{x_k}{n} \right).$$

R commands: Multinomial distribution


```
rmultinom(n, size, prob)
```

```
dmultinom(x, size = NULL, prob, log = FALSE)
```

No commands available for quantile and cumulative distribution functions.

Count data: Poisson Distribution

- $Y \sim \text{Pois}(\lambda)$
- One parameter $\lambda > 0$
- $\text{supp}(Y) = \mathbb{N}_0$
- Probability mass function: $\Pr(Y = y) = e^{-\lambda}\lambda^y / y!$
- $E[Y] = V[Y] = \lambda$

Count data: Poisson Distribution

- Maximum likelihood Estimation

$$\mathcal{L}(\lambda) = \prod_{i=1}^{n} e^{-\lambda} \lambda^{y_i} / y_i!$$

$$\ell(\lambda) = \sum_{i=1}^{n} \log \left(e^{-\lambda} \lambda^{y_i} / y_i! \right)$$

$$\ell(\lambda) = \sum_{i=1}^{n} \left(\log(e^{-\lambda}) + \log(\lambda^{y_i}) - log(y_i!) \right)$$

$$\ell(\lambda) = -n\lambda + \log(\lambda) \sum_{i=1}^{n} y_i - \sum_{i=1}^{n} log(y_i!)$$

Count data: Poisson Distribution

$$\frac{\partial \ell(\lambda)}{\partial \lambda} = -n + \frac{1}{\lambda}\sum_{i=1}^{n} y_i = 0$$

$$\hat{\lambda} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

and

$$\frac{\partial^2 \ell(\lambda)}{\partial^2 \lambda} = -\frac{1}{\lambda^2}\sum_{i=1}^{n} y_i.$$

R commands: Poisson distribution


```
dpois(x, lambda, log = FALSE)

ppois(q, lambda, lower.tail = TRUE, log.p = FALSE)

qpois(p, lambda, lower.tail = TRUE, log.p = FALSE)

rpois(n, lambda)
```