Maximum Likelihood Estimation

Apoorva Lal

April 2, 2021

Stanford

Overview

1. Overview
2. Likelihood: An Intuition
3. Likelihood: A Recipe
4. Likelihood: An Example
5. MLE and Uncertainty

Motivating the Likelihood approach: modeling choices

- Social science is about modeling choices made by strategic agents (countries, politicians, armies)
- Choices can be thought of as being grounded in utility maximisation (latent variable y^* defined net of costs)
- A military attempts a coup in year i ($y_i = 1$) if its utility $y_i^* > 0$
- That is, $y_i = \mathbb{1}_{y_i^* > 0}$
- Model: $y_i^* = \beta + \varepsilon_i \ \ , \varepsilon_i \sim \mathcal{N}(0, 1)$
- Simplest model: β is the average utility of a coup in that country; ε_is are idiosyncratic shocks. Can put covariates on y_i^*.
- Cons: ε_i distribution is paramount: model hinges entirely on noise distribution. Stark contrast to many estimators covered in 450b.
- Pros: Well understood properties, certain parametrisations are 'robust', used widely for classification
    - *Structural econometrics* uses this approach frequently because it allows one to perform 'counterfactual experiments' and quantify welfare implications (again, under the model)

Likelihood analysis

- $y_i = \mathbb{1}_{y_i^* * > 0}$ with $u_i = \beta + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, 1)$

$$\begin{aligned} P(y_i = 1) &= P(\beta + \epsilon_i > 0) \\ &= P(\epsilon_i > -\beta) \\ &= 1 - P(\epsilon_i \leq -\beta) \\ &= 1 - \Phi(-\beta) \\ &= \Phi(\beta). \end{aligned}$$

- Thus, $y_i \sim$ Bernoulli$(\Phi(\beta))$ with pmf

$$p_\beta(y_i) = \Phi(\beta)^{y_i}[1 - \Phi(\beta)]^{1-y_i}$$

- e.g., `pnorm(c(-3, -2, -1.5, 1.0, 0.5, 0))`

Likelihood: A General recipe

Set up distribution that we assume has generated the data in question

$$Pr(y_i) = f(y_i)$$

Write down likelihood function – the joint probability of observing all events under the assumed distribution

$$L(\theta|y_i) = f(y_i|\theta)$$
$$L(\theta|\mathbf{y}) = \prod f(\theta|y_i)$$

Likelihood: A Recipe (cont'd)

Refactor so that we can take the logs more easily;

Take the logs so we have the log-likelihood function:

$$\ell(\theta|\mathbf{y}) = \log(L(\theta|\mathbf{y}))$$

Find parameters that maximise log-likelihood:

$$\frac{\partial \ell(\theta|\mathbf{y})}{\partial \theta_1} = 0 \rightarrow \theta_1^*$$

$$\frac{\partial \ell(\theta|\mathbf{y})}{\partial \theta_2} = 0 \rightarrow \theta_2^*$$

Derive Fisher information to calculate variance of MLE estimate:

$$I_n(\theta) = -\mathbf{H}(L(\theta^*))$$

Likelihood: An Example

- **Motivation:** Suppose that we have N elections with two parties (A, B).
- A's vote share in the last five elections (`y_samp`) was:

```
library(tidyverse); library(plotly)
set.seed(42)
mu = 50; sigma = 5
(y_samp <- rnorm(5, mu, sigma))
```

```
## [1] 56.85 47.18 51.82 53.16 52.02
```

- Can we describe the underlying data-generating process?
- What's our best guess?

Normal MLE Estimation

- **Let's assume:** A's vote share is drawn i.i.d. from a normal distribution with (unknown) mean μ and (unknown) variance σ^2.
- This gives us enough structure to proceed with MLE.
- If we **knew** mean and variance, we could calculate the probability of observing any value:

$$f(x) = \text{pdf}(\mathcal{N}(\mu, \sigma^2)) \equiv \phi(x)$$

- But we **don't**. So we have to make our best guess.

Normal MLE Estimation (cont'd)

- (Step 2). We ask: what is the likelihood of observing any set combination of parameters (μ, σ^2), **given the data that we observe?**

$$L(\mu, \sigma^2|\mathbf{y}) = \prod f(y_i|\mu, \sigma^2) \tag{1}$$

$$= \prod \frac{\exp(-\frac{(y_i-\mu)^2}{2\sigma^2})}{\sqrt{2\pi\sigma^2}} \tag{2}$$

- (Step 3). Refactoring.

$$L(\mu, \sigma^2|\mathbf{y}) = \frac{\exp(-\sum \frac{(y_i-\mu)^2}{2\sigma^2})}{(2\pi)^{n/2}\sigma^{2n/2}}$$

Normal MLE Estimation (cont'd)

- (Step 4). Taking logs.

$$\ell(\mu, \sigma^2|\mathbf{y}) = -\sum \frac{(y_i - \mu)^2}{2\sigma^2} - \frac{n}{2}\log(\sigma^2) + C$$

- Now we can plug in any combination of candidate values for μ and σ^2 into this function and we get a score.
- We have a nicely defined function $\rightarrow$ now to maximise it!

Normal MLE Estimation (cont'd)

```r
likelihood_normal <- function(y, mu, sigma2){
  - sum((y - mu)^2) / (2 * sigma2) - length(y) / 2 * log(sigma2)
}

## Quick example
likelihood_normal(y_samp, 43, 2)
```

```
## [1] -119.7
```

Normal MLE Estimation (cont'd)

Let's set up some more code to plot the likelihood for every combination of μ and σ^2.

```
mu_rg <- seq(30, 70, by = 0.5) ; sigma2_rg <- seq(3, 60, by = 0.5)
viz_df <- expand.grid(mu = mu_rg, sigma2 = sigma2_rg) %>%
  as.data.frame %>% rowwise() %>%
  mutate(likelihood = likelihood_normal(y_samp, mu, sigma2))
max_row <- which.max(viz_df$likelihood)
viz_df[max_row, ]
```

mu	sigma2	likelihood
52	9.5	-8.166

grid-search is extremely inefficient

- We can also find the parameter combination that optimises the likelihood algebraically.
- Recall from Wednesday's lecture:

$$\mu^* = \frac{\sigma^2 y_i}{n} = \bar{y} \tag{3}$$

$$\sigma^{2*} = \frac{1}{n}\sum(y_i - \bar{y})^2 \tag{4}$$

Not all likelihoods have an analytic solution. Many likelihoods need to be maximised numerically.

numerical optimization algorithms

- Newton-Raphson
- Nelder-Mead
- BHHH

MLE and uncertainty (cont'd)

- What if I told you that the data were generated with:

$$\mu = 50, \sigma^2 = 25$$

- Our MLE estimate only takes the "sample" from the DGP.
- We can't make any assumptions about the parameters in the DGP: that's the thing we're trying to estimate using MLE!
- But because of the convergence in distribution, we can still infer how likely the observed MLE estimate is if we assume a true parameter θ_0.

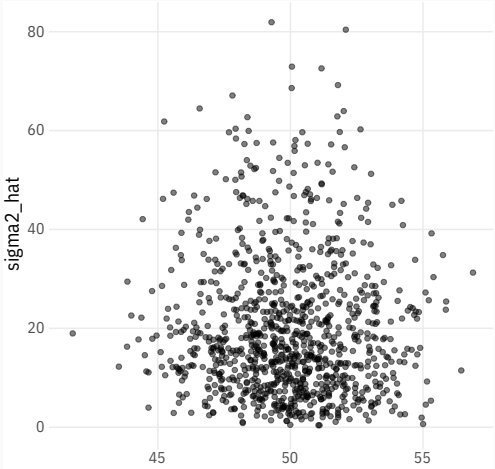
MLE and uncertainty (cont'd)

- Let's create M samples with size n from our true DGP.
- For each of these samples, we calculate the MLE estimate and its variance.

```
get_mean_variance <- function(n){
  y_samp <- rnorm(n, 50, 5)
  y_mean <- mean(y_samp)
  y_sigma <- 1/n * sum((y_samp - y_mean)^2)
  return(c(y_mean, y_sigma))
}
n <- 5 ; m <- 1000
rep_vec <- 1:m ; names(rep_vec) <- 1:m
samp_df <- map_dfr(rep_vec, ~ get_mean_variance(n)) %>% t %>% as.data.frame
```

MLE and uncertainty (cont'd)

```
ggplot(samp_df, aes(V1, V2)) + geom_point(alpha = .5) +
  labs(x = "mu_hat", y = "sigma2_hat")
```



MLE and uncertainty (cont'd)

This is a two-dimensional distribution. We can characterise its (empirical) mean and variance.

```
map_dfr(samp_df, ~ tibble(sampling_mean = mean(.x),
                          sampling_var = var(.x))) %>%
 mutate(dim = c("mle_mean", "mle_var"))
```

sampling_mean	sampling_var	dim
49.93	5.262	mle_mean
19.98	198.426	mle_var

What do these quantities correspond to?

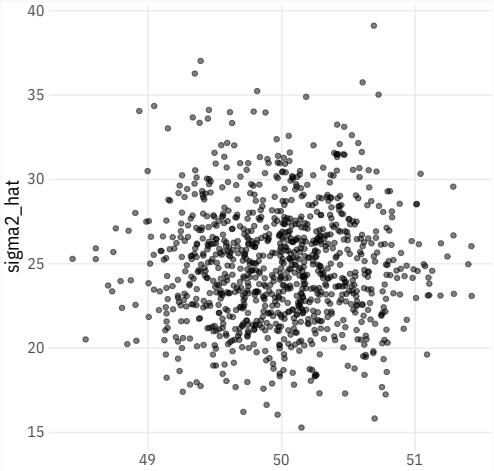
MLE and uncertainty (cont'd)

What happens if we increase the sample size?

```
n <- 100
samp_df <- map_dfr(rep_vec, ~ get_mean_variance(n)) %>% t %>%
  as.data.frame
```

MLE and uncertainty (cont'd)

```
ggplot(samp_df, aes(V1, V2)) +
  geom_point(alpha = .5) + labs(x = "mu_hat", y = "sigma2_hat")
```



MLE and uncertainty (cont'd)

```
map_dfr(samp_df, ~ tibble(sampling_mean = mean(.x),
                          sampling_var = var(.x))) %>%
  mutate(dim = c("mle_mean", "mle_var"))
```

sampling_mean	sampling_var	dim
49.98	0.2527	mle_mean
24.89	12.1670	mle_var

What do these quantities correspond to?

MLE and uncertainty (cont'd)

Recall the asymptotic property of MLE estimators as $n \to \infty$:

$$p(\hat{\mu}, \hat{\sigma}^2) \xrightarrow{d} \text{MVN}\Big((\bar{y}, \frac{1}{n}\sum(y_i - \bar{y})^2), \begin{bmatrix} \frac{\sigma^2}{n} & 0 \\ 0 & \frac{2(\sigma^2)^2}{n} \end{bmatrix}\Big)$$

Since we know the true parameters...

- We have problems when $n = 5$
- Mean ($\mu = 50$) and Variance ($\sigma^2 = 25$) parameters are correctly estimated with $n = 100$
- "Sampling" uncertainty of these parameters falls with sample size
- Variance of sampling distribution converges to $25/100 = 0.25$ and $(2 * 25^2)/100 = 12.5$, respectively

Setup

$$Y_i \sim \text{Bernoulli}(\theta)$$

$$\theta = \Lambda\left(X_i^T\beta\right) = \frac{\exp\left(X_i^T\beta\right)}{1+\exp\left(X_i^T\beta\right)} = \frac{1}{1+\exp\left(-X_i^T\beta\right)}$$

Likelihood:

$$f(Y \mid \theta) = \prod_{i=1}^{n} \Lambda\left(X_i^T\beta\right)^{y_i}\left(1-\Lambda\left(X_i^T\beta\right)\right)^{1-y_i}$$

Log likelihood:

$$\mathcal{L}(\beta) = \sum_{i=1}^{n} y_i \log\left(\Lambda\left(X_i^T\beta\right)\right) + \sum_{i=1}^{n}(1-y_i)\log\left(\Lambda\left(X_i^T\beta\right)\right)$$

Setup

$$Y_i \sim \text{Bernoulli}(\theta)$$
$$\theta = \Phi\left(X_i^T \beta\right)$$

Likelihood:

$$f(Y \mid \theta) = \prod_{i=1}^{n} \Phi\left(X_i^T \beta\right)^{y_i} \left(1 - \Phi\left(X_i^T \beta\right)\right)^{1-y_i}$$

Log likelihood:

$$\mathcal{L}(\beta) = \sum_{i=1}^{n} y_i \log\left(\Phi\left(X_i^T \beta\right)\right) + \sum_{i=1}^{n} (1 - y_i) \log\left(\Phi\left(X_i^T \beta\right)\right)$$

Summary

- Generic recipe for a how to think about likelihood.
    - Decide on model
    - Write down likelihood f'n: how likely is θ given the observed data?
    - Refactor and take the logs
    - Maximise w.r.t. θ (take first derivative)
    - Derive second derivative / Hessian for variance
        - this last step is often non-trivial for more complicated likelihoods. if all else fails, use gradient-free optimizers like BFGS / Golden Section
- Applied to normal distribution (both with algebra and with code)
- Thinking about uncertainty and inference in the context of MLE