Section I: Maximum Likelihood Estimation

450C

Stanford University

Department of Political Science

Toby Nowacki

Overview

Likelihood: An Intuition

- In previous classes, we asked:

"What is the effect of X on Y?"

"Given the (known) DGP, what is the probability of observing our sample?"

- In a clean causal inference setting, we know what the data-generating process is because we are in charge of assigning treatment (in an experiment) or assume that it is as-if random.
- But not all questions lend themselves to clean causal inference.
- What if we *don't know* the DGP?

"How can we best describe the process that generated this data?"

"Given our observed sample, what is the DGP?"

- When we don't know how X and Y are related and want to describe that relationship.

Likelihood: An Intuition (cont'd)

- Previously: given our **model**, how likely are the **results** that we observe?
- Now: given our **results**, how likely is the **model** that we assume?
- This approach yields powerful solutions to some questions
- Some methodological debates yielded dead ends (OLS v logit/probit)

Likelihood: A Recipe

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

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

2. 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)

3. Refactor so that we can take the logs more easily;
4. Take the logs so we have the log-likelihood function:

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

5. Find parameters that maximise log-likelihood:

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

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

6. Derive Fisher information to calculate variance of MLE estimate:

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

$$I_n(\theta_1) = -\frac{\partial^2 L(\theta)}{\partial \theta_1^2}(\theta^*)$$

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:

```
## [1] 51.88486 51.50774 44.50988 44.34797 36.01733
```

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

Breakout Activity I: Guessing

How might we best model A's vote share?

What is your best guess for the vote share in the next election?

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:

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

- 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?**

$$L(\mu, \sigma^2 | \mathbf{y}) = \prod f(y_i | \mu, \sigma^2)$$
$$= \prod \frac{\exp(-\frac{(y_i - \mu)^2}{2\sigma^2})}{\sqrt{2\pi\sigma^2}}$$

- (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 the 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 $\rightarrow$ time for some coding!

Normal MLE Estimation (cont'd)

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

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

```
## [1] -52.77709
```

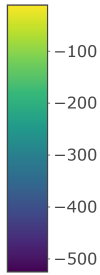
Normal MLE Estimation (cont'd)

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, ]
```

```
## Source: local data frame [1 x 3]
## Groups: <by row>
##
## # A tibble: 1 x 3
##        mu sigma2 likelihood
##     <dbl>  <dbl>      <dbl>
## 1  45.5     34      -11.3
```

```r
viz_mat <- viz_df %>%
  pivot_wider(names_from = mu, values_from = likelihood)  %>%
  dplyr::select(-sigma2) %>% as.matrix
plot_ly(x = mu_rg, y = sigma2_rg, z = viz_mat,
  type="surface")
```



Normal MLE Estimation (cont'd)

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

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

- Again, this is something that we can implement in code.

Breakout activity II

Implement the two functions for the MLE of mean and variance in R and compute the estimated mean and variance of the MLE normal for `y_samp`.

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.

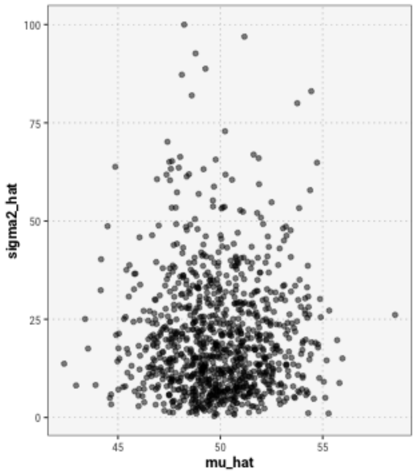
MLE and uncertainty (cont'd)

- 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.

```r
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) +
  theme_tn() +
  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"))
```

```
## # A tibble: 2 x 3
##   sampling_mean sampling_var dim
##           <dbl>        <dbl> <chr>
## 1          49.9         4.97 mle_mean
## 2          20.5       227.   mle_var
```

What do these quantities correspond to?

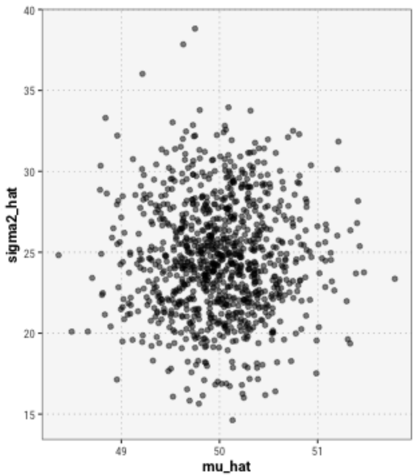
MLE and uncertainty (cont'd)

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) +
  theme_tn() +
  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"))
```

```
## # A tibble: 2 x 3
##   sampling_mean sampling_var dim
##           <dbl>        <dbl> <chr>
## 1          50.0        0.239 mle_mean
## 2          24.6       12.4   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

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
- Applied to normal distribution (both with algebra and with code)
- Thinking about uncertainty and inference in the context of MLE