Full Information Maximum Likelihood

Utrecht University Winter School: Missing Data in R

Kyle M. Lang

Department of Methodology & Statistics
Utrecht University

Outline

FIML Intuition

FIML is an ML estimation method that is robust to ignorable nonresponse.

- FIML partitions the missing information out of the likelihood function so that the model is only estimated from the observed parts of the data.

After a minor alteration to the likelihood function, FIML reduces to simple ML estimation.

- So, let's review ML estimation before moving forward.

Maximum Likelihood Estimation

ML estimation simply finds the parameter values that are “most likely” to have given rise to the observed data.

- The *likelihood* function is just a probability density (or mass) function with the data treated as fixed and the parameters treated as random variables.
- Having such a framework allows us to ask: “Given that I’ve observed these data values, what parameter values most probably describe these data?”

Maximum Likelihood Estimation

ML estimation is usually employed when there is no closed form solution for the parameters we seek.

- This is why you don't usually see ML used to fit general linear models.

After choosing a likelihood function, we iteratively optimize the function to produce the ML estimated parameters.

- In practice, we nearly always work with the natural logarithm of the likelihood function (i.e., the *loglikelihood*).

ML Intuition

Let's say we have the following $N = 10$ observations.

- We assume these data come from a normal distribution with a known variance of $\sigma^2 = 1$.
- We want to estimate the mean of this distribution, μ.

```
(y <- rnorm(n = 10, mean = 5, sd = 1))

 [1] 5.060983 3.364836 4.968344 6.696222 3.610013
 [6] 6.627266 4.165329 4.615346 4.537332 6.024850
```

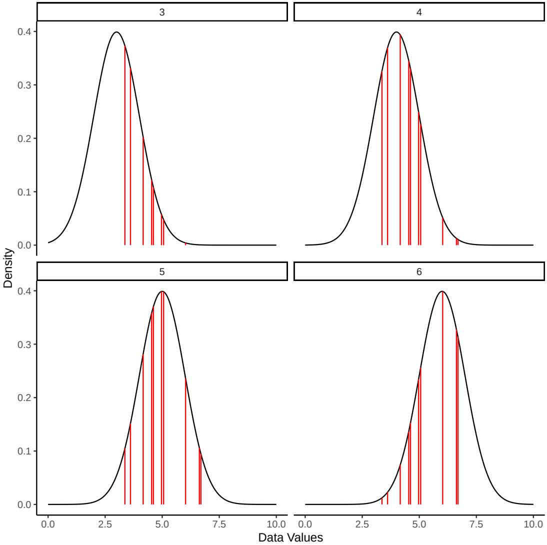
ML Intuition

In ML estimation, we would define different normal distributions.

- Every distribution would have $\sigma^2 = 1$.
- Each distribution would have a different value of μ.

We then compare the observed data to those distributions and see which distribution best fits the data.

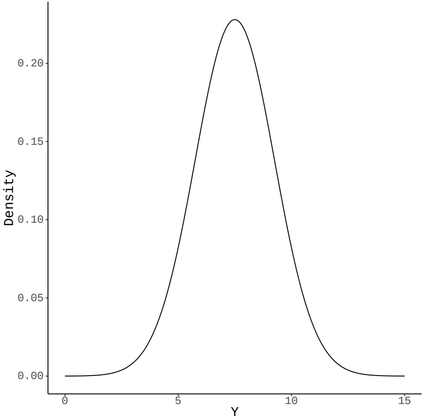
ML Intuition



Likelihoods

Suppose we have the following model:

$$Y \sim \mathrm{N}\left(\mu, \sigma^2\right).$$



Likelihoods

For a given Y_n, we have:

$$P\left(Y_n|\mu, \sigma^2\right) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(Y_n-\mu)^2}{2\sigma^2}}. \tag{1}$$

If we plug estimated parameters into Equation 1, we get the probability of observing Y_n given $\hat{\mu}$ and $\hat{\sigma}^2$:

$$P\left(Y_n|\hat{\mu}, \hat{\sigma}^2\right) = \frac{1}{\sqrt{2\pi\hat{\sigma}^2}} e^{-\frac{(Y_n-\hat{\mu})^2}{2\hat{\sigma}^2}}. \tag{2}$$

Applying Equation 2 to all N observations and multiplying the results produces a *likelihood*:

$$\hat{L}\left(\hat{\mu}, \hat{\sigma}^2\right) = \prod_{n=1}^{N} P\left(Y_n|\hat{\mu}, \hat{\sigma}^2\right).$$

Likelihoods

We generally want to work with the natural logarithm of Equation 2. Doing so gives the *loglikelihood*:

$$\begin{aligned}\hat{\mathcal{L}}\left(\hat{\mu}, \hat{\sigma}^2\right) &= \ln \prod_{n=1}^{N} P\left(Y_n | \hat{\mu}, \hat{\sigma}^2\right) \\ &= -\frac{N}{2} \ln 2\pi - N \ln \hat{\sigma} - \frac{1}{2\hat{\sigma}^2} \sum_{n=1}^{N} \left(Y_n - \hat{\mu}\right)^2\end{aligned}$$

ML tries to find the values of $\hat{\mu}$ and $\hat{\sigma}^2$ that maximize $\hat{\mathcal{L}}\left(\hat{\mu}, \hat{\sigma}^2\right)$.

- Find the values of $\hat{\mu}$ and $\hat{\sigma}^2$ that are *most likely*, given the observed values of Y.

Likelihoods

Suppose we have a linear regression model:

$$Y = \beta_0 + \beta_1 X + \varepsilon,$$

$$\varepsilon \sim \mathrm{N}\left(0, \sigma^2\right).$$

This model can be equivalently written as:

$$Y \sim \mathrm{N}\left(\beta_0 + \beta_1 X, \sigma^2\right)$$

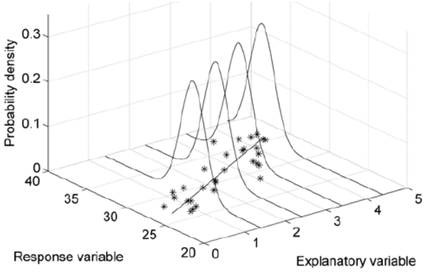


Image retrieved from:
`http://www.seaturtle.org/mtn/archives/mtn122/mtn122p1.shtml`

Likelihoods

For a given $\{Y_n, X_n\}$, we have:

$$P\left(Y_n | X_n, \beta_0, \beta_1, \sigma^2\right) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(Y_n - \beta_0 - \beta_1 X_n)^2}{2\sigma^2}}. \tag{3}$$

If we plug our estimated parameters into Equation 3, we get the probability of observing Y_n given $\hat{Y}_n = \hat{\beta}_0 + \hat{\beta}_1 X_n$ and $\hat{\sigma}^2$.

$$P\left(Y_n | X_n, \hat{\beta}_0, \hat{\beta}_1, \hat{\sigma}^2\right) = \frac{1}{\sqrt{2\pi\hat{\sigma}^2}} e^{-\frac{\left(Y_n - \hat{\beta}_0 - \hat{\beta}_1 X_n\right)^2}{2\hat{\sigma}^2}} \tag{4}$$

Likelihoods

So, our final loglikelihood function would be the following:

$$\begin{aligned}\hat{\mathcal{L}}\left(\hat{\beta}_0, \hat{\beta}_1, \hat{\sigma}^2\right) &= \ln \prod_{n=1}^{N} P\left(Y_n | X_n, \hat{\beta}_0, \hat{\beta}_1, \hat{\sigma}^2\right) \\ &= -\frac{N}{2} \ln 2\pi - N \ln \hat{\sigma} - \frac{1}{2\hat{\sigma}^2} \sum_{n=1}^{N} \left(Y_n - \hat{\beta}_0 - \hat{\beta}_1 X_n\right)^2 .\end{aligned}$$

Example

```r
## Fit a model:
out1 <- lm(ldl ~ bp + glu + bmi, data = diabetes)

## Extract the predicted values and estimated residual standard error:
yHat <- predict(out1)
s    <- summary(out1)$sigma

## Compute the row-wise probabilities:
pY <- dnorm(diabetes$ldl, mean = yHat, sd = s)

## Compute the loglikelihood, and compare to R's version:
sum(log(pY)); logLik(out1)[1]

[1] -2109.939
[1] -2109.93
```

Multivariate Normal Distribution

The PDF for the multivariate normal distribution is:

$$P(\mathbf{Y}|\boldsymbol{\mu},\Sigma) = \frac{1}{\sqrt{(2\pi)^P|\Sigma|}}e^{-\frac{1}{2}(\mathbf{Y}-\boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{Y}-\boldsymbol{\mu})}.$$

So, the multivariate normal loglikelihood is:

$$\mathcal{L}\,(\boldsymbol{\mu},\Sigma) = -\left[\frac{P}{2}\ln(2\pi) + \frac{1}{2}\ln|\Sigma| + \frac{1}{2}\right]\sum_{n=1}^{N}(\mathbf{Y}_n-\boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{Y}_n-\boldsymbol{\mu}).$$

Which can be further simplified if we multiply through by -2:

$$-2\mathcal{L}\,(\boldsymbol{\mu},\Sigma) = \left[P\ln(2\pi) + \ln|\Sigma|\right]\sum_{n=1}^{N}(\mathbf{Y}_n-\boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{Y}_n-\boldsymbol{\mu}).$$

Steps of ML

1. Choose a probability distribution, $f(Y|\theta)$, to describe the distribution of the data, Y, given the parameters, θ.
2. Choose some estimate of θ, $\hat{\theta}^{(i)}$.
3. Compute each row's contribution to the loglikelihood function by evaluating: $\ln\left[f\left(Y_n|\hat{\theta}^{(i)}\right)\right]$.
4. Sum the individual loglikelihood contributions from Step 3 to find the loglikelihood value, $\hat{\mathcal{L}}$.
5. Choose a "better" estimate of the parameters, $\hat{\theta}^{(i+1)}$, and repeat Steps 3 and 4.
6. Repeat Steps 3 – 5 until the change between $LL^{(i-1)}$ and $LL^{(i)}$ falls below some trivially small threshold.
7. Take $\hat{\theta}^{(i)}$ as your estimated parameters.

Full Information Maximum Likelihood

From ML to FIML

The nth observation's contribution to the multivariate normal loglikelihood function would be the following:

$$\mathcal{L}(\mu, \Sigma)_n = -\frac{P}{2}\ln(2\pi) - \frac{1}{2}\ln|\Sigma| - \frac{1}{2}(\mathbf{Y}_n - \mu)^T\Sigma^{-1}(\mathbf{Y}_n - \mu). \tag{5}$$

From ML to FIML

The nth observation's contribution to the multivariate normal loglikelihood function would be the following:

$$\mathcal{L}(\mu, \Sigma)_n = -\frac{P}{2}\ln(2\pi) - \frac{1}{2}\ln|\Sigma| - \frac{1}{2}(\mathbf{Y}_n - \mu)^T \Sigma^{-1}(\mathbf{Y}_n - \mu). \tag{5}$$

FIML just tweaks Equation 5 a tiny bit:

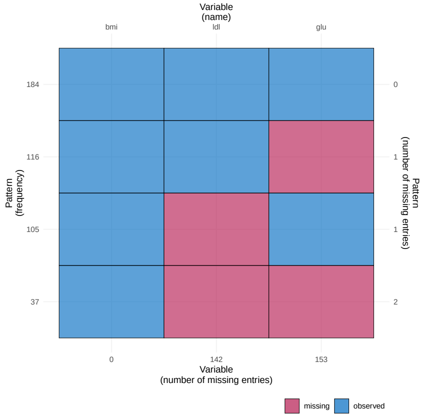
$$\mathcal{L}(\mu, \Sigma)_{fiml,n} = -\frac{P}{2}\ln(2\pi) - \frac{1}{2}\ln|\Sigma_q| - \frac{1}{2}(\mathbf{Y}_n - \mu_q)^T \Sigma_q^{-1}(\mathbf{Y}_n - \mu_q).$$

Where $q = 1, 2, \ldots, Q$ indexes response patterns.

Visualize the Response Patterns

These data contain 4 unique response patterns.

- We'd define 4 different version of μ and Σ.
- We'd calculate each individual loglikelihood contributions using the appropriate flavor of μ and Σ.



FIML Example

We most often apply FIML when estimating latent variable models.

- We can eastily apply FIML to latent variable models in **lavaan**.

```
library(lavaan)

## Read in some data:
bfi <- readRDS(paste0(dataDir, "bfi_datasets.rds"))$incomplete

## Specify the measurement model:
cfaMod <- '
agree =~ A1 + A2 + A3 + A4 + A5
open  =~ O1 + O2 + O3 + O4 + O5
'

## Estimate the CFA using FIML:
fimlOut <- cfa(cfaMod, data = bfi, std.lv = TRUE, missing = "fiml")
```

FIML Example

```
partSummary(fimlOut, 1:4)

lavaan 0.6-12 ended normally after 41 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                        31

  Number of observations                          2800
  Number of missing patterns                        32

Model Test User Model:

  Test statistic                               360.865
  Degrees of freedom                                34
  P-value (Chi-square)                           0.000
```

FIML Example

```
partSummary(fimlOut, 7, fmi = TRUE)

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)      FMI
  agree =~
    A1                0.538    0.038   14.109    0.000    0.399
    A2               -0.764    0.030  -25.530    0.000    0.417
    A3               -0.989    0.033  -29.722    0.000    0.434
    A4               -0.693    0.039  -17.953    0.000    0.407
    A5               -0.838    0.031  -26.622    0.000    0.403
  open =~
    O1                0.635    0.025   24.968    0.000    0.003
    O2               -0.640    0.036  -17.615    0.000    0.004
    O3                0.831    0.029   28.880    0.000    0.008
    O4                0.345    0.028   12.333    0.000    0.002
    O5               -0.647    0.031  -20.890    0.000    0.005
```

FIML Example

```
partSummary(fimlOut, 8, fmi = TRUE)

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)      FMI
  agree ~~
    open             -0.290    0.028  -10.530    0.000    0.147
```

FIML Example

```
partSummary(fimlOut, 9, fmi = TRUE)

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)      FMI
   .A1                2.422    0.031   77.128    0.000    0.281
   .A2                4.814    0.025  189.726    0.000    0.252
   .A3                4.616    0.028  163.443    0.000    0.232
   .A4                4.735    0.032  146.063    0.000    0.275
   .A5                4.573    0.027  167.561    0.000    0.243
   .O1                4.816    0.021  225.180    0.000   -0.000
   .O2                2.713    0.030   91.745    0.000   -0.000
   .O3                4.441    0.023  192.731    0.000   -0.000
   .O4                4.894    0.023  212.316    0.000   -0.000
   .O5                2.490    0.025   99.119    0.000   -0.000
    agree             0.000
    open              0.000
```

FIML Example

```
partSummary(fimlOut, 10, fmi = TRUE)

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)      FMI
   .A1                1.693    0.059   28.649    0.000    0.352
   .A2                0.765    0.037   20.491    0.000    0.463
   .A3                0.736    0.047   15.632    0.000    0.489
   .A4                1.654    0.061   27.307    0.000    0.354
   .A5                0.876    0.043   20.256    0.000    0.484
   .O1                0.878    0.031   28.191    0.000    0.003
   .O2                2.039    0.062   32.972    0.000    0.005
   .O3                0.797    0.040   19.848    0.000    0.006
   .O4                1.369    0.038   35.732    0.000    0.002
   .O5                1.349    0.044   30.369    0.000    0.005
    agree             1.000
    open              1.000
```

FMI with FIML

As you saw above, we can also estimate the FMI when using FIML.

- The FMI is calculated using the method described by Savalei and Rhemtulla (2012).

Savalei and Rhemtulla (2012) take an information-theoretic approach to defining the FMI.

- Based on the *Missing Information Principle* of Orchard and Woodbury (1972)
- Their FMI estimates the ratio of missing to complete information for each parameter.

You can use this method to compute the FMI for sufficient statistics via the **semTools**::`fmi()` function.

AUXILIARY VARIABLES

Satisfying the MAR Assumption

Like MI, FIML also requires MAR data.

- Parameters will be biased when data are MNAR.

Unlike MI, FIML directly treats the missing data while estimating the analysis model.

- The MAR predictors must be included in the analysis model.
- Otherwise, FIML reduces to pairwise deletion.

When the MAR predictors are not substantively interesting variables, naively included them in the analysis model can change the model's meaning.

Saturated Correlates Technique

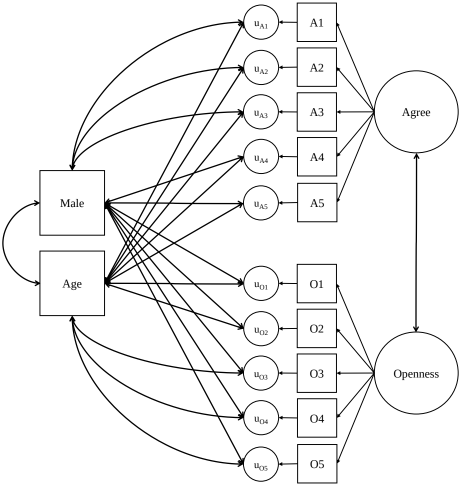
Graham (2003) developed the *saturated correlates* approach to meet two desiderata:

1. Satisfy the MAR assumption by incorporating MAR predictors into the analysis model.
2. Maintain the fit and substantive meaning of the analysis model.

The approach entails incorporating the MAR predictors via a fully-saturated covariance structure:

1. Allow every MAR predictor to covary with all other MAR predictors.
2. Allow every MAR predictor to covary with all observed variables in the analysis model (or their residuals).

Saturated Correlates Diagram



Saturated Correlates Example

We can use the `lavaan.auxiliary()` function from **semTools** (or one of its wrappers) to streamline the analysis.

```
library(semTools)

## Estimate the CFA from above with auxiliary variables:
fimlOut2 <- bfi %>%
    mutate(male = as.numeric(sex == "male")) %>%
    cfa.auxiliary(cfaMod,
                  data = .,
                  aux = c("age", "male"),
                  std.lv = TRUE)
```

Saturated Correlates Example

The `cfa.auxiliary()` function has automatically added the following paths to our CFA model.

```
age ~~ age
age ~~ male
age ~~ A1
age ~~ A2
age ~~ A3
age ~~ A4
age ~~ A5
age ~~ O1
age ~~ O2
age ~~ O3
age ~~ O4
age ~~ O5
```

```
male ~~ male
male ~~ A1
male ~~ A2
male ~~ A3
male ~~ A4
male ~~ A5
male ~~ O1
male ~~ O2
male ~~ O3
male ~~ O4
male ~~ O5
```

Saturated Correlates Example

The auxiliaries have been correlated with all other variables.

```
inspect(fimlOut2, "est")$theta[11:12, 1:6] %>% round(3)

         A1     A2     A3     A4     A5    O1
age  -2.927  1.330  1.188  2.622  1.863 0.763
male  0.119 -0.094 -0.083 -0.082 -0.073 0.052

inspect(fimlOut2, "est")$theta[11:12, 7:12] %>% round(3)

         O2    O3    O4     O5     age   male
age  -0.773 0.491 0.147 -1.484 123.998 -0.246
male -0.006 0.020 0.001 -0.015  -0.246  0.220
```

Saturated Correlates Example

The degrees of freedom have not changed, though.

```
## Naive FIML:
fitMeasures(fimlOut, "df")

df
34

## FIML w/ saturated correlates:
fitMeasures(fimlOut2, "df")

df
34
```

Saturated Correlates Example

Let's compare the effects of the various missing data treatments on the latent covariance estimates.

	Complete Data	Listwise Deletion	Multiple Imputation	Naive FIML	FIML w/ Auxiliaries
Est	-0.306	-0.254	-0.299	-0.290	-0.295
FMI	—	—	0.305	0.147	0.146

Latent Covariances

References

Graham, J. W. (2003). Adding missing-data-relevant variables to fiml-based structural equation models. *Structural Equation Modeling: A Multidisciplinary Journal*, *10*(1), 80–100. doi: 10.1207/S15328007SEM1001_4

Orchard, T., & Woodbury, M. A. (1972). A missing information principle: Theory and applications. In *Proceedings of the sixth berkeley symposium on mathematical statistics and probability, volume 1: Theory of statistics.*

Savalei, V., & Rhemtulla, M. (2012). On obtaining estimates of the fraction of missing information from full information maximum likelihood. *Structural Equation Modeling: A Multidisciplinary Journal*, *19*(3), 477–494.