Describing Bilteral Migration Data

Guy J. Abel

Multiplicative Component Model

- Rogers et al. (2002) proposed dis-aggregating origin-destination flow tables into separate components to allow for an easier examination of migration flows
  - Overall component - level of migration γ
  - Origin component - relative 'pushes' from each region α_i
  - Destination component - relative 'pulls' to each region β_j
  - Origin–Destination interaction component - physical or social distance between places not explained by the overall and main effects. δ_{ij}
- Simple calculations to estimate each component:

$$\gamma = m_{++} \qquad \alpha_i = \frac{m_{i+}}{m_{++}} \qquad \beta_j = \frac{m_{+j}}{m_{++}} \qquad \delta_{ij} = \frac{m_{ij}}{\gamma \alpha_i \beta_j}$$

- The interaction, δ_{ij}, is the ratio of observed flow to an expected flow (for the case of no interaction).

Multiplicative Component Model

- The dis-aggregation of the components is multiplicative:

$$m_{ij} = \gamma \alpha_i \beta_j \delta_{ij}$$

- Equivalent to a saturated Poisson regression model ($R^2 = 1$) where
  - γ is constant term
  - α_i is categorical term for the origin regions
  - β_j is categorical term for the destination regions
  - δ_{ij} is an interaction term between the α_i and β_j

$$\log m_{ij} = \gamma + \alpha_i ORIG_i + \beta_j DEST_j + \delta_i ORIG_i : DEST_j$$

- When data is in a tidy format with row h would be:

$$\log y_h = \beta_0 + \beta_1 ORIG_h + \beta_2 DEST_h + \beta_3 ORIG_h : DEST_h$$

- Poisson regression models such as these - where all the predictor variables are categorical - are also know as log-linear models
- Standard functions for fitting regression models, such as `glm()` in R will provide the same fitted values, but different parameter estimates
  - Use different coding system for the constraints when estimating parameters
  - Rogers' terms the parameter estimates using the equations for $\gamma, \alpha_i, \beta_j$ and δ_{ij} above the *total reference* coding system

Multiplicative Component Model

- The *migest* package contains a `multi_comp()` function to generate parameter estimates from an origin-destination flow matrix
  - Demonstrate with previous dummy data set

```
> r <- LETTERS[1:4]
> m0 <- matrix(data = c(0, 100, 30, 70,
+                       50, 0, 45, 5,
+                       60, 35, 0, 40,
+                       20, 25, 20, 0),
+              nrow = 4, ncol = 4, byrow = TRUE,
+              dimnames = list(orig = r, dest = r))
> addmargins(m0)
     dest
orig    A   B  C   D Sum
  A     0 100 30  70 200
  B    50   0 45   5 100
  C    60  35  0  40 135
  D    20  25 20   0  65
  Sum 130 160 95 115 500
```

Multiplicative Component Model

```
> library(tidyverse)
> library(migest)
> m0 %>%
+   multi_comp() %>%
+   round(3)
     dest
orig        A       B       C       D     Sum
  A     0.000   1.563   0.789   1.522   0.400
  B     1.923   0.000   2.368   0.217   0.200
  C     1.709   0.810   0.000   1.288   0.270
  D     1.183   1.202   1.619   0.000   0.130
  Sum   0.260   0.320   0.190   0.230 500.000
```

Multiplicative Component Model

- As the model is saturated, the fitted values are the same as the observed values.

```
> multi_comp(m = m0)
     dest
orig          A         B         C         D         Sum
  A   0.0000000 1.5625000 0.7894737 1.5217391   0.4000000
  B   1.9230769 0.0000000 2.3684211 0.2173913   0.2000000
  C   1.7094017 0.8101852 0.0000000 1.2882448   0.2700000
  D   1.1834320 1.2019231 1.6194332 0.0000000   0.1300000
  Sum 0.2600000 0.3200000 0.1900000 0.2300000 500.0000000
>
> # fitted value for A to B
> 500 * 0.4 * 0.32 * 1.5625
[1] 100
```

Multiplicative Component Model

- The total reference coding scheme for the parameter estimates are easier to examine than parameter estimates from a Poisson model fitted using `glm()`
  - More detail on `glm()` in next section

```
> d0 <- as.data.frame.table(x = m0, responseName = "flow")
> f0 <- glm(formula = flow ~ orig + dest + orig * dest, family = poisson(),
+           data = d0)
> f0

Call:  glm(formula = flow ~ orig + dest + orig * dest, family = poisson(),
    data = d0)

Coefficients:
(Intercept)        origB        origC        origD        destB        destC
     -24.30        28.21        28.40        27.30        28.91        27.70
      destD  origB:destB  origC:destB  origD:destB  origB:destC  origC:destC
      28.55       -57.12       -29.45       -28.68       -27.81       -56.10
origD:destC  origB:destD  origC:destD  origD:destD
     -27.70       -30.85       -28.96       -55.85

Degrees of Freedom: 15 Total (i.e. Null);  0 Residual
Null Deviance:      463.7
Residual Deviance: 2.232e-10    AIC: 96.27
```

Multiplicative Component Model

```
> # fitted and observed values are the same
> d0 %>%
+   as_tibble() %>%
+   mutate(fit = round(f0$fitted.values, digits = 5))
# A tibble: 16 x 4
   orig  dest   flow   fit
   <fct> <fct> <dbl> <dbl>
 1 A     A         0     0
 2 B     A        50    50
 3 C     A        60    60
 4 D     A        20    20
 5 A     B       100   100
 6 B     B         0     0
 7 C     B        35    35
 8 D     B        25    25
 9 A     C        30    30
10 B     C        45    45
11 C     C         0     0
12 D     C        20    20
13 A     D        70    70
14 B     D         5     5
15 C     D        40    40
16 D     D         0     0
```

Multiplicative Component Model

- Rogers' and colleagues have used the multiplicative component model to estimate migration flow tables
- Expand to multiple dimensions
- Rectify bumpy age schedules
  - Replace reported age parameters (proportions) in the multiplicative component model with proportions from a more regular schedule.
  - Multiply the new age parameters with the existing total, origin, destination and interaction parameters to obtain new estimated flows.

Multiplicative Component Model

- Italian data in *migest* package

```
> italy_area
# A tibble: 3,500 x 5
   orig       dest        year age_grp  flow
   <chr>      <chr>      <dbl> <fct>   <dbl>
 1 North-West North-West  1970 0-4         0
 2 North-East North-West  1970 0-4      2350
 3 Center     North-West  1970 0-4      1687
 4 South      North-West  1970 0-4      9697
 5 Islands    North-West  1970 0-4      5139
 6 North-West North-East  1970 0-4      2448
 7 North-East North-East  1970 0-4         0
 8 Center     North-East  1970 0-4      1063
 9 South      North-East  1970 0-4      1560
10 Islands    North-East  1970 0-4       689
# ... with 3,490 more rows
# i Use `print(n = ...)` to see more rows
```

Multiplicative Component Model

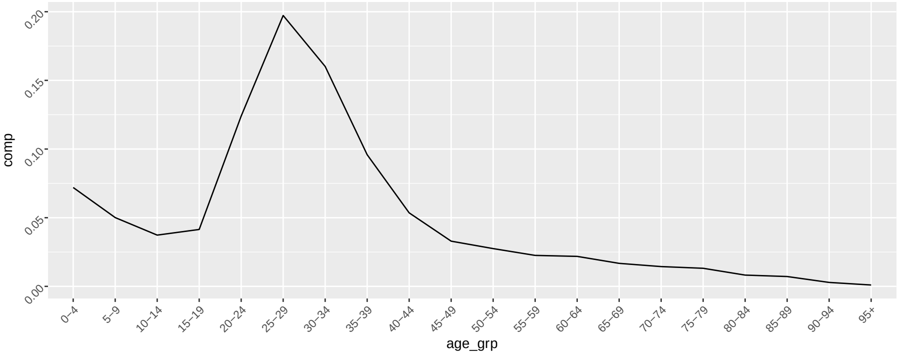
```
> # single year, multiple age groups
> c0 <- italy_area %>%
+   filter(year == 2000) %>%
+   multi_comp()
> round(c0, 3)
, , age_grp = 0-4

            dest
orig          Center   Islands North-East North-West     South       Sum
  Center       0.000     1.401      0.859      0.909     2.370     0.010
  Islands      0.970     0.000      1.181      1.513     0.681     0.012
  North-East   1.053     1.916      0.000      1.179     2.501     0.010
  North-West   0.877     2.490      0.887      0.000     2.023     0.014
  South        1.409     0.531      1.184      1.102     0.000     0.025
  Sum          0.016     0.007      0.017      0.018     0.014     0.072

, , age_grp = 5-9

            dest
orig          Center   Islands North-East North-West     South       Sum
  Center       0.000     1.589      0.779      0.762     2.243     0.007
  Islands      1.166     0.000      1.393      1.707     0.562     0.010
  North-East   0.840     1.932      0.000      0.936     2.085     0.006
  North-West   0.877     2.714      0.844      0.000     1.963     0.010
  South        1.387     0.507      1.283      1.151     0.000     0.018
```

Multiplicative Component Model

```
> # origin components (shares)
> c0 %>%
+   as.data.frame.table(responseName = "comp") %>%
+   filter(orig != "Sum", dest == "Sum", age_grp == "Sum")
        orig dest age_grp      comp
1     Center  Sum     Sum 0.1477314
2    Islands  Sum     Sum 0.1663483
3 North-East  Sum     Sum 0.1245945
4 North-West  Sum     Sum 0.2017835
5      South  Sum     Sum 0.3595424
>
> # destination components (shares)
> c0 %>%
+   as.data.frame.table(responseName = "comp") %>%
+   filter(orig == "Sum", dest != "Sum", age_grp == "Sum")
  orig       dest age_grp       comp
1  Sum     Center     Sum 0.23305555
2  Sum    Islands     Sum 0.08368777
3  Sum North-East     Sum 0.25254113
4  Sum North-West     Sum 0.26283900
5  Sum      South     Sum 0.16787656
```

Multiplicative Component Model

```
> # age components
> c0 %>%
+   as.data.frame.table(responseName = "comp") %>%
+   filter(orig == "Sum", dest == "Sum", age_grp != "Sum") %>%
+   ggplot(mapping = aes(x = age_grp, y = comp, group = 1)) +
+   geom_line() +
+   theme(axis.text = element_text(angle = 45, hjust = 1))
```



Log-Linear Models

- Rogers' and collaborators like to shorten the multiplicative form of the log-linear model to use capital letters to represent parameters

$$m_{ij} = \gamma \alpha_i \beta_j \delta_{ij} = TO_i D_j OD_{ij}$$

- When there is multiple origin-destination tables, by different age groups, sex, education level, etc,... the notation can be easily used to study different log-linear models

$$m_{ij} = TO_i D_j A_x OD_{ij} OA_{ix}$$

- When data is in a tidy format with row h would be:

$$\begin{aligned} \log y_h = &\beta_0 + \beta_1 ORIG_h + \beta_2 DEST_h + \beta_3 AGE_x + \\ &\beta_4 ORIG_h : DEST_h + \beta_5 ORIG_h : AGE_h \end{aligned}$$

Log-Linear Models

- We can fit log-linear models in R using the `glm()` function (for generalised linear models)
- Requires a `formula`, `data` and `family` argument
- The `formula` argument is similar to that in `xtabs()`, where we use the ~ symbol to separate the the response and explanatory variables
    - For example the model in the previous slide would use `formula = flow ~ orig + dest + age + orig:dest + orig:age`
    - Use `:` or `*` to denote interactions
- The `family` argument should be set to `poisson()` for a log-linear model

Log-Linear Models

- Example with age-specific migration flows between Italian regions in 1970

```
> d1 <- italy_area %>%
+   filter(orig != dest,
+          year == 1970) %>%
+   # rename so later output fits on slide
+   rename(age = age_grp)
> d1
# A tibble: 400 x 5
   orig       dest        year age    flow
   <chr>      <chr>      <dbl> <fct> <dbl>
 1 North-East North-West  1970 0-4    2350
 2 Center     North-West  1970 0-4    1687
 3 South      North-West  1970 0-4    9697
 4 Islands    North-West  1970 0-4    5139
 5 North-West North-East  1970 0-4    2448
 6 Center     North-East  1970 0-4    1063
 7 South      North-East  1970 0-4    1560
 8 Islands    North-East  1970 0-4     689
 9 North-West Center      1970 0-4    2097
10 North-East Center      1970 0-4    1183
# ... with 390 more rows
# i Use `print(n = ...)` to see more rows
```

Log-Linear Models

```
> glm(formula = flow ~ orig + dest, family = poisson(), data = d1)

Call:  glm(formula = flow ~ orig + dest, family = poisson(), data = d1)

Coefficients:
   (Intercept)     origIslands  origNorth-East  origNorth-West       origSouth
       6.39791         0.17515        -0.20852         0.99427         0.98847
    destIslands  destNorth-East  destNorth-West       destSouth
       -0.76940        -0.32536         1.08367         0.02188

Degrees of Freedom: 399 Total (i.e. Null);  391 Residual
Null Deviance:      758100
Residual Deviance: 5e+05    AIC: 503100
```

Log-Linear Model Analysis

- As we increase the number of dimensions of the data, it might become important to understand which dimensions of the data are most important
- We can use log-linear models with detailled migration data to
  - Understand the dominate dimensions, for example Imhoff et al. (1997) Rogers et al. (2002)
  - Predict origin-destination flows with partial data, for example Beer et al. (2010) Rogers, Willekens, and Raymer (2003) Raymer (2007)
  - Project detailed origin-destination flows, for example Raymer, Bonaguidi, and Valentini (2006)
- All the above examples involve fitting a number log-linear models based on different dimensions of the data frames
  - Use model fit statistics to judge the best model

Log-Linear Model Analysis

- One approach to choosing the most important dimensions is to fit all possible combinations of models - known as *dredging* the model space
- The `dredge()` function in the *MuMIn* package will fit all combinations of regression models given an upper limit, i.e. the most complex model.
  - The number of combinations grows exponentially with the number of predictors
  - Does not allow `na.action = "na.omit"` - set by default in `glm()` for handling missing values in regression models

Log-Linear Model Analysis

- Fit the most complex model using `glm()`.
  - Set `na.action = na.fail` to exclude failed models in when using the `dredge()` function later
  - Most complex model typically involves at least all two-way interactions
- The `formula` argument in `glm()` allows the use `()^2` to construct all two-way interactions, i.e. the below give the identical outputs
  - Use `()^3` for all three way interactions

```
> f1 <- glm(formula = flow ~ (orig + dest + age)^2,
+           family = poisson(), data = d1, na.action = na.fail)
> f2 <- glm(formula = flow ~ orig * dest + orig * age + dest * age,
+           family = poisson(), data = d1, na.action = na.fail)
>
> # check terms used in models
> attr(f1$terms, "term.labels")
[1] "orig"      "dest"      "age"       "orig:dest" "orig:age"  "dest:age"
> attr(f2$terms, "term.labels")
[1] "orig"      "dest"      "age"       "orig:dest" "orig:age"  "dest:age"
```

Log-Linear Model Analysis

- Models will have many estimated coefficients
  - Some will be non-determinable because no observations (e.g. diagonal terms such as `origIslands:destIslands` below) as

```
> f1 %>%
+   coef() %>%
+   length()
[1] 196
> summary(f1)

Call:
glm(formula = flow ~ (orig + dest + age)^2, family = poisson(),
    data = d1, na.action = na.fail)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-12.1125   -1.4474    0.0186    1.5870    8.3143

Coefficients: (5 not defined because of singularities)
                                Estimate Std. Error z value Pr(>|z|)
(Intercept)                    6.812e+00  2.085e-02 326.747  < 2e-16 ***
origIslands                    4.277e-01  2.236e-02  19.126  < 2e-16 ***
origNorth-East                 1.709e-01  2.390e-02   7.151 8.64e-13 ***
origNorth-West                 7.906e-01  1.789e-02  44.190  < 2e-16 ***
origSouth                      1.381e+00  2.027e-02  68.123  < 2e-16 ***
```

Log-Linear Model Analysis

- Pass the upper model to `dredge()`. Use `trace = TRUE` to monitor progress.

```
> library(MuMIn)
> mm <- dredge(global.model = f1, trace = TRUE)
Fixed term is "(Intercept)"
0 : glm(formula = flow ~ 1, family = poisson(), data = d1, na.action = na.fail)
1 : glm(formula = flow ~ age + 1, family = poisson(), data = d1,
    na.action = na.fail)
2 : glm(formula = flow ~ dest + 1, family = poisson(), data = d1,
    na.action = na.fail)
3 : glm(formula = flow ~ age + dest + 1, family = poisson(), data = d1,
    na.action = na.fail)
4 : glm(formula = flow ~ orig + 1, family = poisson(), data = d1,
    na.action = na.fail)
5 : glm(formula = flow ~ age + orig + 1, family = poisson(), data = d1,
    na.action = na.fail)
6 : glm(formula = flow ~ dest + orig + 1, family = poisson(), data = d1,
    na.action = na.fail)
7 : glm(formula = flow ~ age + dest + orig + 1, family = poisson(),
    data = d1, na.action = na.fail)
11 : glm(formula = flow ~ age + dest + age:dest + 1, family = poisson(),
    data = d1, na.action = na.fail)
15 : glm(formula = flow ~ age + dest + orig + age:dest + 1, family = poisson(),
    data = d1, na.action = na.fail)
21 : glm(formula = flow ~ age + orig + age:orig + 1, family = poisson(),
```

Log-Linear Model Analysis

```
> mm
Global model call: glm(formula = flow ~ (orig + dest + age)^2, family = poisson(),
    data = d1, na.action = na.fail)
---
Model selection table
   (Int) age dst org age:dst age:org dst:org  df      logLik    AICc     delta
64 6.515   +   +   +       +       +       + 191   -2944.992  6624.6      0.00
56 6.616   +   +   +               +       + 115   -5286.311 10896.6   4271.97
48 6.619   +   +   +       +               + 115   -7617.005 15558.0   8933.35
40 6.691   +   +   +                       +  39  -11408.460 22903.6  16278.99
32 6.865   +   +   +       +       +         180  -22817.598 46292.7  39668.13
24 6.995   +   +   +               +         104  -25545.324 51372.7  44748.08
16 6.997   +   +   +       +                 104  -27876.018 56034.1  49409.47
8  7.070   +   +   +                          28  -31667.473 63395.3  56770.72
12 7.612   +   +           +                 100  -82409.496 165086.5 158461.95
4  7.684   +   +                              24  -86200.951 172453.1 165828.50
22 7.250   +       +               +         100 -114058.016 228383.6 221758.99
6  7.325   +       +                          24 -120180.165 240411.5 233786.93
2  7.734   +                                  20 -160715.461 321473.1 314848.54
39 6.019       +   +                       +  20 -231284.060 462610.3 455985.74
7  6.398       +   +                           9 -251543.073 503104.6 496480.01
3  7.012       +                               5 -306076.551 612163.3 605538.65
5  6.653           +                           5 -340055.765 680121.7 673497.08
1  7.062                                       1 -380591.061 761184.1 754559.53
   weight
```

Log-Linear Model Analysis

- Model comparison based on model statistics measuring the goodness of fit.
  - AIC measures a goodness of fit with a penalty for the number of predictor variables.
  - AICc has a bias correction term for small samples
- Typically the origin-destination interaction term is very important for accurately predicting the age-specific origin-destination migration flows
- The time to conduct a dredging analysis increase exponentially as the number of dimensions increases.

Exercise (ex6.R)

```
# 0.  a) Load the KOSTAT2022.Rproj file.
#     Run the getwd() below. It should print the directory where the
#     KOSTAT2022.Rproj file is located.
getwd()
#     b) Load the packages used in this exercise
library(tidyverse)
library(migest)
library(MuMIn)
##
##
##
# 1. Run the code below to read in the migration flow data for flows within the
#    USA, decomposed by move type, from 6 censuses between 1940 and 2000.
us <- read_csv("./data/us_area_1940_2000.csv")
us
# 2. Show the multiplicative components, rounded to 3 digits, for the flows from
#    the 2000 census
us %>%
  filter(year == 2000) %>%
  #####() %>%
  round(digits = #####)
# 3. Fit a log-linear model to the entire data set using all two-way
#    interactions between the four dimensions (orig, dest, period and move_type)
f <- glm(formula = flow ~ (##### + dest + ##### + move_type) ^#####,
         family = #####(), data = us, na.action = na.fail)
```

References I

Beer, Joop de, James Raymer, Rob van der Erf, and Leo van Wissen. 2010. "Overcoming the Problems of Inconsistent International Migration data: A New Method Applied to Flows in Europe." *European Journal of Population / Revue Européenne de Démographie* 26 (4): 459–81. https://doi.org/10.1007/s10680-010-9220-z.

Imhoff, Evert van, Nicole van der Gaag, Leo van Wissen, and Philip H. Rees. 1997. "The selection of internal migration models for European regions." *International Journal of Population Geography IJPG* 3 (2): 137–59. https://doi.org/10.1002/(SICI)1099-1220(199706)3:2%3C137::AID-IJPG63%3E3.0.CO;2-R.

Raymer, James. 2007. "The estimation of international migration flows: a general technique focused on the origin–destination association structure." *Environment and Planning A* 39 (4): 985–95. https://doi.org/10.1068/a38264.

Raymer, James, Alberto Bonaguidi, and Alessandro Valentini. 2006. "Describing and projecting the age and spatial structures of interregional migration in Italy." *Population, Space and Place* 12 (5): 371–88. https://doi.org/10.1002/psp.414.

Rogers, Andrei, Frans Willekens, Jani Little, and James Raymer. 2002. "Describing migration spatial structure." *Papers in Regional Science* 81 (1): 29–48. https://doi.org/10.1007/s101100100090.

Rogers, Andrei, Frans Willekens, and James Raymer. 2003. "Imposing Age and Spatial Structures on Inadequate Migration-Flow Datasets." *The Professional Geographer* 55 (1): 56–69. https://doi.org/10.1111/0033-0124.01052.