Modelling spatial spread II

Andrea Parisi

Main contact for this notebook: Andrea Parisi (andrea.parisi@warwick.ac.uk)

Course context

Purpose and scope of the course

This material has been developed as part of the *GeMVi* project: *NIHR Global Health Research Group on the Application of Genomics and Modelling to the Control of Virus Pathogens* in East Africa and the University of Warwick.

In these workshops you will be introduced to some common techniques used in infectious disease modelling. The topics covered will include the implementation of deterministic and stochastic compartmental models, the use of maximum likelihood estimation to analyse super-spreading behaviour in novel disease outbreaks, modelling of contact patterns, and optimisation techniques for fitting epidemic models to real data.

Contributors

The contributors are all part of the GeMVi project:

- Prof. James Nokes (JNokes@kemri-wellcome.org)
- Prof. Matt Keeling (m.j.keeling@warwick.ac.uk)
- Dr. Joe Hilton (j.hilton@warwick.ac.uk)
- Dr. Rabia Aziza (rabia.aziza@warwick.ac.uk)
- Dr. Samuel Brand (s.brand@warwick.ac.uk)
- Dr. Andrea Parisi (andrea.parisi@warwick.ac.uk)

Helpful references for the course

- Anderson, R. M., & May, R. M. (1992). Infectious Diseases of Humans: Dynamics and Control.
- Bjørnstad, O. N. (2018). Epidemics, models and data using R. https://doi.org/10.1007/978-3-319-97487-3
- Diekmann, O., & Heesterbeek, J. A. P. (2000). Mathematical epidemiology of infectious diseases: model building, analysis and interpretation. 104: John Wiley and Sons.
- Keeling, Matt J., & Pejman Rohani. Modeling Infectious Diseases in Humans and Animals. Princeton University Press, 2008.
- Martcheva, M., 2015. An Introduction to Mathematical Epidemiology, Texts in Applied Mathematics. Springer US, Boston, MA. https://doi.org/10.1007/978-1-4899-7612-3
- Vynnycky, Emilia, & White, Richard G. An Introduction to Infectious Disease Modelling

Requirements

The following packages are required for this tutorial. Please have them installed before the start.

1. pracma - *Practical Numerical Math Routines* implements a number of mathematical functions, a couple of which are used in this tutorial.
2. deSolve - *Differential Equation solver* implements methods to integrate initial value differential equations
3. viridis - *Viridis colour palette* implements the viridis colour palettes, a set of colour palettes that provide perceptually uniform colours, are appropriate for color blindness and render properly in black and white. Particularly useful for visualising maps.
4. sf - *Simple Features* is a package useful to handle GIS (Geographic Information Systems) datasets.
5. lwgeom [optional] - This is an optional package and is not required for this tutorial. However, it may be needed to handle some vector maps.

The packages may be installed using (uncomment as needed and copy-paste into the console):

```
#install.packages("pracma", "deSolve", "viridis", "sf")
#install.packages("lwgeom")
```

Introduction

In the previous workshop we learned how to build a simple metapopulation model to explore the spatial spread of an infectious disease. The key to perform simulations of spatial spread is the matrix ρ_{ij} which describes how individual move between the sub populations. In this workshop we will explore alternative formulations of metapopulation models which may offer an improved description. We will then focus once again on the ρ_{ij} matrix and describe methods that aim at producing estimates for its entries when data on human mobility is only partially available. Common situations are when the total number of commuters leaving and entering each sub population is known, but the number of individuals moving between each pair of sub populations is not known, or when data is known for a subset of sub populations. Models for human mobility may be used to estimate commuting fluxes by optimizing a set of parameters to the known data. In the worst case scenario no data is available, yet it is still possible to use some model of human mobility to build fluxes that can be collectively tuned to explore the dependence of disease spread on human mobility.

We start by loading the required packages:

```
library(pracma)
library(sf)
```

```
## Linking to GEOS 3.6.2, GDAL 2.2.3, PROJ 4.9.3
```

```
#library(lwgeom)  # Required by some shapefiles
library(viridis)
```

```
## Loading required package: viridisLite
```

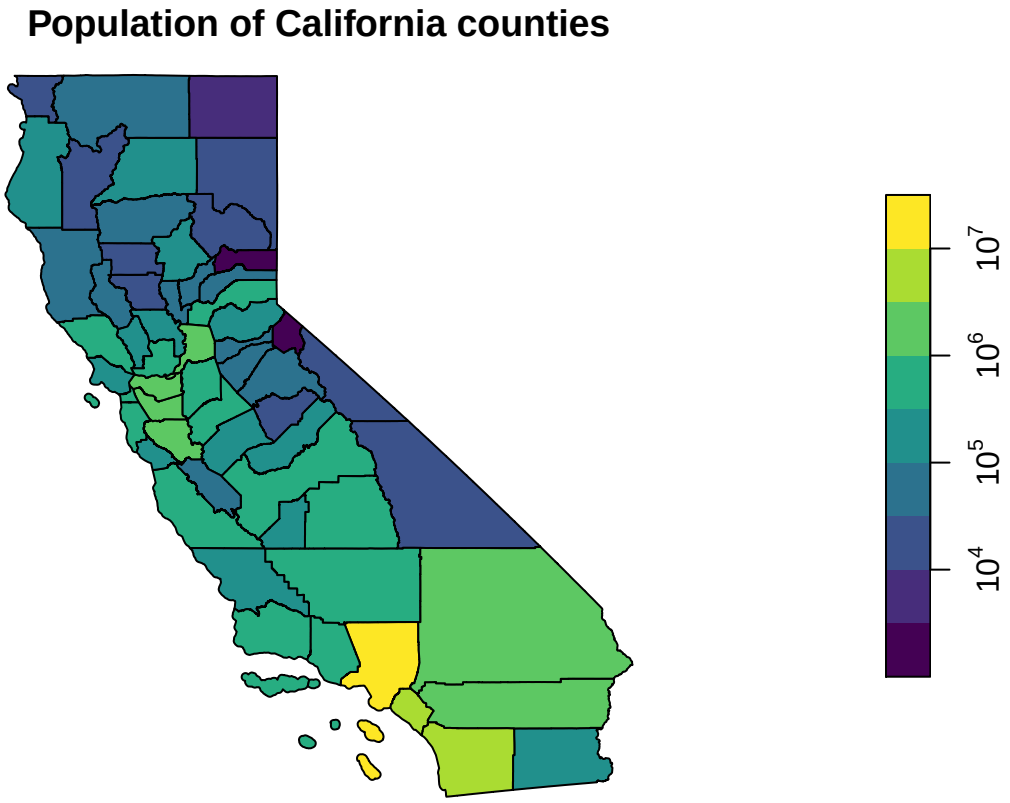
We then read the data.frames that were produced in the previous workshop. The data is organised as follows:

1. *counties.wgs84* This is the main database, storing polygons of each county, population size, and other information that we do not use. It uses a coordinate reference system based on latitude/longitude based on the World Geodetic System standard (WGS84).
2. *counties.pop* This contains population sizes for each Californian county
3. *coordinates* List of coordinates of centroids of each county
4. *counties.mapper* This is a look-up dictionary of County name and Id
5. *comm* Commuting data: each row stores the number of commuters travelling from location `src` to location `dst`

```
# Polygons of California counties + other infos
counties.wgs84 <- read_sf(dsn="Data/Local/map.shp")
# Population size of each county
counties.pop <- read.table(file="Data/Local/population.dat", header=TRUE)
# Latitude & longitude of each county
coordinates <- as.matrix( read.table(file="Data/Local/coordinates.dat",
                                     header=TRUE) )
# Mapping of county names and county id
counties.mapper <- read.table(file="Data/Local/mapper.dat", header=TRUE)
patches <- nrow(counties.pop)
# Commuting data
comm <- read.table(file="Data/Local/comm.dat", header=TRUE)
```

Let us get back into practice by re-plotting the map of California

```
plot(counties.wgs84['pop'], pal=viridis(9), logz=TRUE,
     main="Population of California counties")
```



Other metapopulation models for spatial spread

In the previous workshop we studied a simple metapopulation model described by the following set of equations:

$$\begin{aligned} \frac{dS_i(t)}{dt} &= \mu(N_i - S_i(t)) - \beta_i \sum_j \rho_{ij} \frac{S_i(t) I_j(t)}{N_i} \\ \frac{dI_i(t)}{dt} &= \beta_i \sum_j \rho_{ij} \frac{S_i(t) I_j(t)}{N_i} - (\gamma + \mu) I_i(t). \end{aligned} \tag{1}$$

Such model is based on the assumption that the disease is spread mostly by the movement of infectious individuals who leave their sub population spreading the disease elsewhere. The transmission term has the form:

$$\beta_i \sum_j \rho_{ij} \frac{S_i(t) I_j(t)}{N_i}$$

the term describes the fact that susceptible individuals in county i may be infected from infectious individuals coming from (and usually resident in) county j. The tranmission matrix ρ_{ij} represents

the fraction of the population in j visiting county i, scaled by an appropriate factor that takes into account the duration of stay in i. The normalizing factor N_i represents the population of county i and is an indication that we are assuming that contagion occurs in county i. Another way to interpret this is that the chance for infectious individuals coming from j to encounter susceptibles from i is proportional to the density of susceptibles in i, or $S_i(t)/N_i$.

Transmission however does not necessarily occur because someone infected decides to travel. Susceptible individuals may travel, get the disease while travelling and bring back the disease when they return home. In this case transmission occur outside the home county. A model that describe this process has the form:

$$\begin{aligned} \frac{dS_i(t)}{dt} &= \mu(N_i - S_i(t)) - \beta_i \sum_j \rho_{ji} \frac{S_i(t) I_j(t)}{N_j} \\ \frac{dI_i(t)}{dt} &= \beta_i \sum_j \rho_{ji} \frac{S_i(t) I_j(t)}{N_j} - (\gamma + \mu) I_i(t). \end{aligned} \tag{2}$$

It is straightforward to rewrite the above equations as:

$$\begin{aligned} \frac{d\vec{S}(t)}{dt} &= \mu(\vec{N} - \vec{S}(t)) - \vec{\lambda'} \otimes \vec{S}(t) \\ \frac{d\vec{I}(t)}{dt} &= \vec{\lambda'} \otimes \vec{S}(t) - (\gamma + \mu)\vec{I}(t). \end{aligned} \tag{3}$$

with $\vec{\lambda'} = \vec{\beta} \otimes \rho^{\mathsf{T}} \cdot (\vec{I} \oslash \vec{N})$. Here we use the transpose of the transmission matrix built for the previous case. If ρ_{ij} represents the fraction of individuals in j visiting i, $\rho_{ji} = (\rho^{\mathsf{T}})_{ij}$ is the fraction of individuals from i visiting j.

The approaches detailed above consider one of the two transmission modes: either transmission led from infectives moving out of their home county, or individuals acquiring infection elsewhere and returning to their home counties. In the above descriptions, the contact between the individuals between different subpopulations is modelled, but individuals do not truly move as the population remain the same. The model described below takes instead a slightly different approach and models the change in population due to the movement of individuals. Here we assume that contagion occurs in a given county, but the populations of infectives, susceptibles, etc are increased or decreased according to a *transport operator* $\Omega(\{X_i\})$:

$$\begin{aligned} \frac{dS_i(t)}{dt} &= \mu(N_i - S_i(t)) - \beta_i \frac{S_i(t) I_i(t)}{N_i} + \Omega(\{S_i(t)\}) \\ \frac{dI_i(t)}{dt} &= \beta_i \frac{S_i(t) I_i(t)}{N_i} - (\gamma + \mu) I_i(t) + \Omega(\{I_i(t)\}). \end{aligned} \tag{4}$$

The $\Omega\{X_i\}$ operator describes how individuals move between counties:

$$\Omega(\{X_i\}) = \sum_j \rho_{ij} X_j - \sum_j \rho_{ji} X_i$$

Note however, that in this approach the commuting of individuals (moving out and returning back) leads to a clear mixing within each subpopulation: there is no guarantee that those who went from i to j will actually move back. In this sense, movement of individuals is not properly described. This however does not mean it is not a useful approach: for instance it was used to make predictions on flu pandemics (model description in Colizza et al. 2007 and Colizza et al. 2006, where a stochastic version is implemented, but the principle is the same).

A more complete and complex approach is to divide the subpopulation into sets of individual staying or travelling. Given a subpopulation N_j, we identify N_{ij} with the set of individual from subpopulation j currently visiting i, with the understanding that N_{jj} is the set of individuals from j currently staying in j. Thus:

$$N_j = \sum_i N_{ij}(t)$$

is the total number of individual resident in location j. Summing over the second index however, gives the instantaneous population (residents and non residents) in sub population i:

$$N_i^*(t) = \sum_j N_{ij}(t)$$

These new variables verify the following set of equations that describes the mobility of individuals.

$$\begin{aligned}
\frac{dN_{ii}(t)}{dt} &= -\sum_{j\neq i} \bar{\rho}_{ji} N_{ii}(t) + \sum_{j\neq i} \tau N_{ji}(t) \\
\\
\frac{dN_{ij}(t)}{dt} &= \bar{\rho}_{ij} N_{jj}(t) - \tau N_{ij}(t)
\end{aligned}$$

The matrix $\bar{\rho}_{ij}$ represents as the fraction of individuals from j who visit i (for commuting data that would be the number of daily commuters divided by their population size of origin), whilst τ is the rate of return to the home location. τ may also be a matrix, but it is often simplified to a common rate representing the inverse of the time spent at the destination. The relationship between our definitions of contact matrix is $\bar{\rho}_{ij} \times 1/\tau = \rho_{ij}$

Similarly to the above equations, we can write $I_j = \sum_i I_{ij}$ and $S_j = \sum_i S_{ij}$ and write the corresponding set of equations as follows:

$$\begin{aligned}
\frac{dS_{ii}(t)}{dt} &= \mu(N_{ii}(t) - S_{ii}(t)) - \beta_i S_{ii}(t) \frac{\sum_j I_{ij}(t)}{\sum_j N_{ij}(t)} - \sum_{j\neq i} \bar{\rho}_{ji} S_{ii}(t) + \sum_{j\neq i} \tau S_{ji}(t) \\
\\
\frac{dS_{ij}(t)}{dt} &= \mu(N_{ij}(t) - S_{ij}(t)) - \beta_i S_{ij}(t) \frac{\sum_j I_{ij}(t)}{\sum_j N_{ij}(t)} + \bar{\rho}_{ij} S_{jj}(t) - \tau S_{ij}(t) \\
\\
\frac{dI_{ii}(t)}{dt} &= \beta_i S_{ii}(t) \frac{\sum_j I_{ij}(t)}{\sum_j N_{ij}(t)} - (\gamma + \mu) I_{ii}(t) - \sum_{j\neq i} \bar{\rho}_{ji} I_{ii}(t) + \sum_{j\neq i} \tau I_{ji}(t) \\
\\
\frac{dI_{ij}(t)}{dt} &= \beta_i S_{ij}(t) \frac{\sum_j I_{ij}(t)}{\sum_j N_{ij}(t)} - (\gamma + \mu) I_{ij}(t) + \bar{\rho}_{ij} I_{jj}(t) - \tau I_{ij}(t)
\end{aligned}$$

The model above, consisting of $3n^2$ differential equations (n being the number of sub populations), is amenable to simplifications depending on the values of $\bar{\rho}_{ij}$ and τ. Further details may be found in the following:

- Sattenspiel, L., and Dietz, K. (1995). A structured epidemic model incorporating geographic mobility among regions. Mathematical Biosciences 128, 71–91.
- Keeling, M.J., and Rohani, P. (2002). Estimating spatial coupling in epidemiological systems: a mechanistic approach. Ecology Letters 5, 20–29.
- Keeling, Matt J., & Pejman Rohani. Modeling Infectious Diseases in Humans and Animals. Princeton University Press, 2008.

Models for human mobility

In order to describe the spatial spread of an infectious disease, knowledge of the contact matrix $\bar{\rho}_{ij}$ is essential. However, often this is not readily available: in this case, one would like to have a formula that can express $\bar{\rho}_{ij}$ as a function of a tunable set of parameters and other characteristics related to the population. Our approach to $\bar{\rho}_{ij}$ has been to identify it with the fraction of individuals from j that travels to i, that is $\bar{\rho}_{ij} = T_{ji}/N_j$ where T_{ij} represents the actual number of individuals travelling from i to j. Note that $\bar{\rho}_{ij}$ represents the number of individuals moving from j to i, whereas T_{ij} represents the number of individuals travelling from i to j. The different notation reflects the conventions used in these two distinct domains. Models for human mobility typically attempt to provide an estimate for T_{ij}. Several models have been proposed in recent years to model human mobility. Among these, two have been predominant: the Gravity model and the Radiation model.

Gravity model

The Gravity model was proposed multiple times within the field of Economics, with the earliest reference dating back to 1924 in an attempt to describe a migration law based on the Gravity Law in Physics. The Gavity Law states that the force between two masses m_1 and m_2, is proportional to the product of the two, divided by their distance squared: $F_{12} = -Gm_1m_2/r_{12}^2$. The original formulation mimicked this form, however in its current formulation the model predicts that the number of individuals travelling from location i to location j is given by:

$$T_{ij} = C\frac{n_i^\alpha n_j^\beta}{f(r_{ij})}$$

where n_i and n_j are the populations at locations i and j, whilst α, β and C are model parameters. The above expression is valid for all $i \neq j$. The function $f(r_{ij})$ is an increasing function of the distance r_{ij} between i and j, so that the far away places have a small flux. A typical form for the distance function is $f(r_{ij}) = r_{ij}^\gamma$, with γ an additional model parameter. The fluxes T_{ij} thus

depend on four parameters: C, α, β and γ. We can easily estimate these parameters so that they describe the commuting fluxes in California. Using logarithms we get:

$$\log T_{ij} = \log C + \alpha \log n_i + \beta \log n_j - \gamma \log r_{ij}$$

Therefore we only need to build a data.frame with the above data and use a linear fitting method.

```
comm.model <- data.frame( src=as.integer(), dst=as.integer(),
            pop.src=as.integer(), pop.dst=as.integer(),
            flux=as.numeric(), dist=as.numeric() )

for (src in 1:patches)  {
    for (dst in 1:patches)  {
        if (src == dst)  {
            next
        }

        hasentry <- nrow(comm[ comm$dst == dst & comm$src == src, ])
        if (hasentry)  {
            flux <- comm[ comm$dst == dst & comm$src == src, ]$Total
        } else {
            flux <- 0
        }
        popsrc <- as.numeric(counties.pop[ counties.pop$id == src, ]$pop)
        popdst <- as.numeric(counties.pop[ counties.pop$id == dst, ]$pop)
        dist <- haversine( coordinates[src,], coordinates[dst,] )

        comm.model <- rbind( comm.model, data.frame(src=src, dst=dst,
                pop.src=popsrc, pop.dst=popdst, flux=flux, dist=dist) )
    }
}
#counties.pop
nrow(comm.model)
```

```
## [1] 3306
```

```
grav_model <- lm( log(flux) ~ log(pop.src) + log(pop.dst) + log(dist),
                  data=comm.model[comm.model$flux > 0,], na.action = "na.omit" )
summary(grav_model)
```

```
##
## Call:
## lm(formula = log(flux) ~ log(pop.src) + log(pop.dst) + log(dist),
##     data = comm.model[comm.model$flux > 0, ], na.action = "na.omit")
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -3.8461 -0.7965  0.0090  0.7795  5.9652
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.19848    0.38031  -0.522    0.602
## log(pop.src)  0.61829    0.02116  29.217   <2e-16 ***
## log(pop.dst)  0.62865    0.02100  29.932   <2e-16 ***
## log(dist)    -2.10654    0.04414 -47.726   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.261 on 1610 degrees of freedom
## Multiple R-squared:   0.64,  Adjusted R-squared:  0.6394
## F-statistic: 954.2 on 3 and 1610 DF,  p-value: < 2.2e-16
```

```
comm.model$pred_grav_model <- exp( predict(grav_model, comm.model) )
head(comm.model, n=10)
```

```
##     src dst pop.src pop.dst  flux      dist pred_grav_model
## V1    1   2 1650950    1047     0 214.31701        5.573908
## V11   1   3 1650950   37429     0 141.38348      126.817422
## V12   1   4 1650950  226231     0 226.09612      146.169523
## V13   1   5 1650950   45322     0 132.69689      163.466968
## V14   1   6 1650950   21496     0 172.53515       58.829747
## V15   1   7 1650950 1137268 41235  30.47676    27485.971636
## V16   1   8 1650950   27382     0 489.28625        7.622034
## V17   1   9 1650950  185976    25 174.69039      222.507953
## V18   1  10 1650950  976830    65 222.88595      377.832027
## V19   1  11 1650950   27840     0 221.54461       40.875713
```

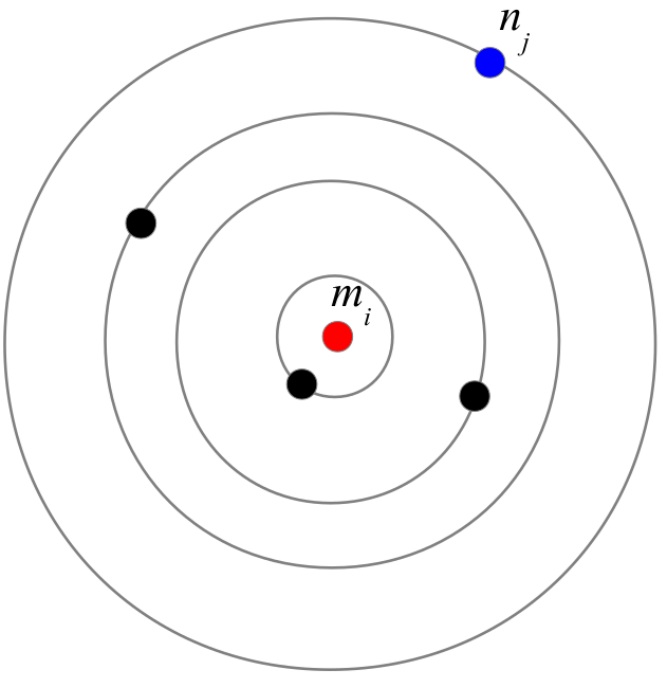
Radiation model

The Radiation model has the form:

$$T_{ij} = T_i \frac{m_i n_j}{(m_i + s_{ij})(m_i + n_j + s_{ij})} \qquad (1)$$

where m_i and n_j represent the number of individuals in locations i and j respectively. The terms s_{ij} represent the number of individuals leaving in a circle of radius r_{ij} centered in i, excluding the populations in i and j. Note that the model uses two distinct variable names (`m` and `n`) for the source and destination populations: this is linked to the different meaning the two have in the model construction, however for most purposes the two can be considered equivalent and correspond simply to population sizes. The term $T_i = m_i \frac{N_c}{N}$ represents the total flux out of location i and is expressed as a fraction N_c/N of the population in i, where N is the total

population *in the map* under consideration, and N_c is the number of individuals participating in commuting (thus N_c/N represents the fraction of the population that leaves their home counties to work, for instance, in another county). In theory, the number N_c can be calculated by summing all the fluxes appearing in the data, which provides the total number of individuals participating in commuting between counties. However, the fraction N_c/N could be treated as the only unknown of the model and can be fitted to the data.



The terms m_i and n_j refer to the source location (in red) and destination (in blue). The term s_{ij} corresponds to the population of the locations in-between, represented in black.

The tricky part in dealing with this model is to build the terms s_{ij}. To do that, for each county we must order the other counties by distance and then calculate the partial cumulative sums as follows:

```
# Order for each county and then by distance
comm.model <- comm.model[ order(comm.model$src, comm.model$dist), ]
# For each src, build cumulative sums of pop.dst
# Note that we are exluding the population at the source
sij_list <- tapply( comm.model$pop.dst, comm.model$src, function(x) if (length(x) > 1)
# Output from tapply is a list. Thus unlist.
comm.model$sij <- as.numeric( unlist( sij_list ) )
```

We now evaluate the factor that multiplies N_c/N in the equation of the radiation model (1). That is, we can write $T_{ij} = \frac{N_c}{N} f_{ij}$ and thus calculate

$$f_{ij} = m_i \frac{m_i n_j}{(m_i + s_{ij})(m_i + n_j + s_{ij})}$$

```
# Evaluate the factor that multiplies $N_c/N$ in the radiation model
comm.model$fact_radModel <- comm.model$pop.src *
    comm.model$pop.src*comm.model$pop.dst /
    ((comm.model$pop.src+comm.model$sij)*
        (comm.model$pop.src+comm.model$sij+comm.model$pop.dst))
head(comm.model)
```

```
##      src dst pop.src pop.dst  flux     dist pred_grav_model     sij
## V15    1   7 1650950 1137268 41235 30.47676       27485.972       0
## V139   1  41 1650950  767906 37275 48.02949        8236.965 1137268
## V141   1  43 1650950 1928368 73555 51.29696       12792.029 1905174
## V137   1  39 1650950  732809  2460 64.72650        4266.198 3833542
## V146   1  48 1650950  438858  1780 68.99147        2701.984 4566351
## V142   1  44 1650950  274396   405 71.06505        1889.647 5005209
##      fact_radModel
## V15      673395.20
## V139     211092.25
## V141     269491.12
## V137      58576.08
## V146      28904.55
## V142      16212.65
```

As $T_{ij} = f_{ij}\ N_c/N$, it is straightforward to estimate the optimal value of the ratio N_c/N that fits the model predictions to data.

```
# Now fit to data. The term '-1' sets the intercept to zero
rad_model <- lm( flux ~ fact_radModel - 1, comm.model )
# The fitted parameter is N_c/N
summary(rad_model)
```

```
##
## Call:
## lm(formula = flux ~ fact_radModel - 1, data = comm.model)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -81169     -9      0      4 112626
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## fact_radModel 0.064255   0.001039   61.84   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5144 on 3305 degrees of freedom
## Multiple R-squared:  0.5364, Adjusted R-squared:  0.5363
```

```
## F-statistic:  3825 on 1 and 3305 DF,  p-value: < 2.2e-16
```

The result of the fitting tells us that less than 6.5% of the population commutes between different counties. Modelled fluxes can now be easily obtained:

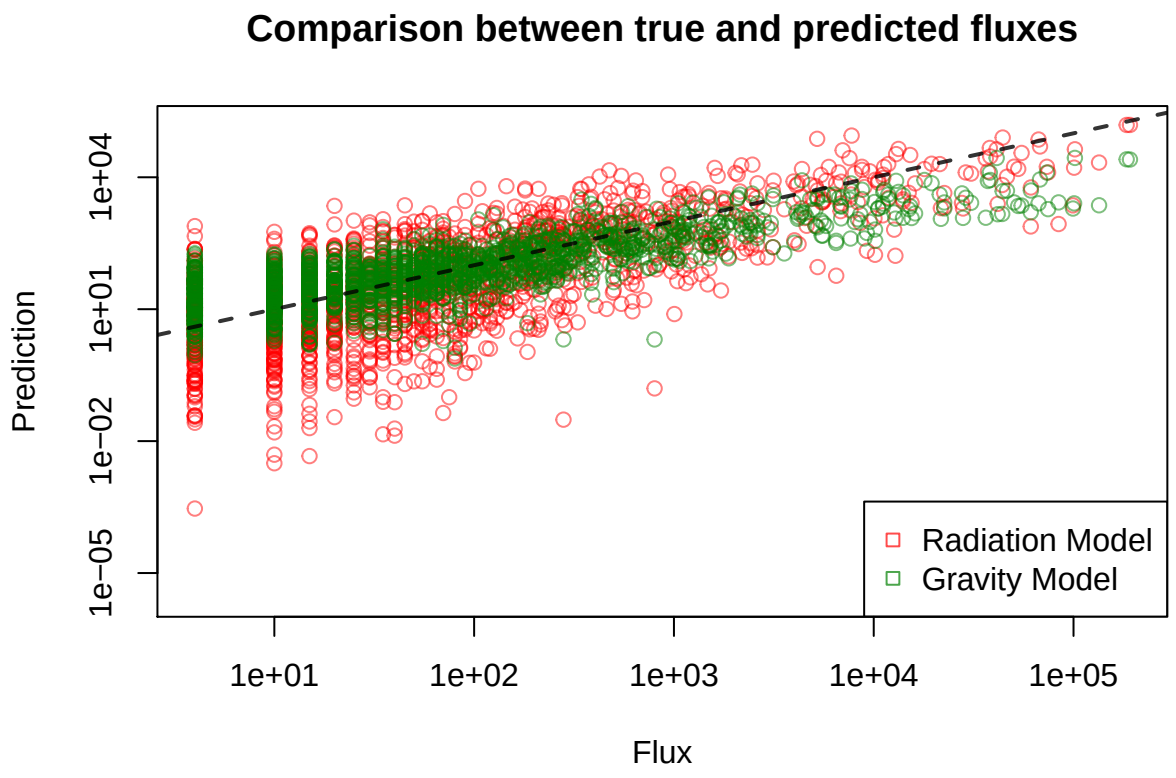
```
# Evaluate predicted fluxes
comm.model$pred_rad_model <- predict(rad_model, data=comm.model)
```

We can now verify how good these predictions are. We can do this by plotting the predicted vs true fluxes.

```
plot( comm.model$flux, comm.model$pred_rad_model, col=rgb(1,0,0,0.5),
      ty='p', log="xy", xlab="Flux", ylab="Prediction",
      main="Comparison between true and predicted fluxes" )
```

```
## Warning in xy.coords(x, y, xlabel, ylabel, log): 1692 x values <= 0 omitted
## from logarithmic plot
```

```
lines( comm.model$flux, comm.model$pred_grav_model, col=rgb(0,0.5,0,0.5), ty='p' )
abline( a=0, b=1, col=rgb(0,0,0,0.8), lw=2, lty='dashed')
legend( "bottomright", col=c(rgb(1,0,0,0.7), rgb(0,0.5,0,0.7)),
        legend=c("Radiation Model", "Gravity Model"), bty="o", pch=22)
```



If the models were able to reproduce the true fluxes, all point would fall on the diagonal (the

black dashed line). Unfortunately no existing model currently is able to exactly reproduce the fluxes observed in the real world. Nevertheless, predictions show some degree of correlation with the true fluxes.

Exercises

- *Exercise 1* - To fit the Gravity model we had to exclude entries with zero flux. To fit the Radiation model we did not exclude these entries. Does it matter?

Common Part of Commuters

In order to quantify how well a model is able to reproduce the true fluxes, a measure of similarity called the *Common Part of Commuters* is often used. This is defined as:

$$CPC(\{T_{ij}\},\{T^*_{ij}\}) = \frac{2\sum_{ij}\min(T_{ij},T^*_{ij})}{\sum_{ij}T_{ij}+\sum_{ij}T^*_{ij}}$$

where T^*_{ij} are the predicted fluxes and T_{ij} are the true fluxes. The CPC index reaches value 1 when all predicted fluxes match the data, and falls to zero when the discrepancy between the two is large. Let us define a function that implements the above formula:

```
CPC <- function( data, colname )  {
    numerator <- 2*sum( apply( data, 1, function(x) {min(x['flux'], x[colname])}) )
    denominator <- sum(data['flux']) + sum(data[colname])
    return(numerator/denominator)
}
```

We can now use the formula to evaluate the goodness of the fluxes predicted by the radiation model and the gravity model:

```
print(CPC(comm.model, 'pred_rad_model'))
```

```
## [1] 0.5490848
```

```
print(CPC(comm.model, 'pred_grav_model'))
```

```
## [1] 0.3128933
```

How good are these predictions? Typical values for the CPC that are obtained for these model do not exceed much 0.5. Thus the radiation model in this case seems to do a better job. However, this is not always the case, and the Gravity model often leads to better predictions.

Exercises

- *Exercise 2* - Take 5 distinct random subsets of commuting data and estimate the parameters of the radiation model for each subset. How do they compare with the parameters measured from the full set?
- *Exercise 3* - Order the commuting data in terms of fluxes from the largest to the smallest; take the first p fluxes and estimate the radiation model parameter. How does the estimate change with p? What does this tell us about parameter estimation for scarce data?
- *Exercise 4* - Order the commuting data in terms of size of the source population from the largest to the smallest; take the first p fluxes and estimate the radiation model parameter. How does the estimate change with p? What does this tell us about parameter estimation for scarce data?
- *Exercise 5* - Compare the spread of an SIR model where the spatial contact matrix is obtained from true data with that predicted with the radiation model. You can use the build.contact.matrix function introduced in the previous workshop. Can you see any difference in the outcomes?

```
write.table(comm.model, file="Data/Local/commframe.dat",
            col.names=TRUE, row.names=FALSE)
```