Max Planck Institute for Demographic Research
MP116 Microsimulation with Socsim

Carl Mason
Emilio Zagheni
UC Berkeley, Demography

June 14, 2010

1 Project 3: Calibrating Socsim Vital Rates with Bayesian Melding

In this project we will use Socsim to answer the question: "What demographic rates could have caused *that* to happen?" For purposes of this exercise, we will pretend that archaeologists have recently discovered fragmentary information about an ancient and long forgotten society which we will call Imaginatia. Our challenge is to take that bit of information – and find a set of vital rates that are consistent with it in a maximum likelihood sort of way.

Although grossly oversimplified for this exercise, the procedure we illustrate could in theory be used to find the set vital rates that are most likely to produce any particular population attribute that is calculable from a complete census including kinship.

In practice, we propose that this method be used to "tune" Socsim rates. That is to modify rates which are not known with certainty in order to better match census or survey data on the population being simulated – a goal much more modest than that of discovering an entire set of vital rates for a long extinct society.

The goals of this adventure are:

- To learn how to run Socsim from within R
- To gain a cocktail party understanding of *Bayesian Melding*

- To ponder an important question about generational depth as measured by "Josh's G", which as we'll see, is the expected number of living maternal ancestors (mother, grandmother etc) at birth.

In Section 3 we describe the procedure step by step. After reading that section it is perfectly safe to begin doing the exercise all of whose parts are in the `Calibration.tgz` tar file. For those who prefer a deeper understanding, Emilio would be happy to send you a copy of his excellent dissertation, (emilioz@demog.berkeley.edu).

1.1 Setup

In addition to the computing environment described in the Cousin Diversity exercise, **You will also need** to have the `demogR` package installed in R – type `library(demogR)` to see if it is installed. If not try `install.packages (``demogR'')` – before seeking professional help.

The bits that are specific to this exercise are in `http://lab.demog.berkeley.edu/socsim/Calibration.tgz` which contains the `Calibration` directory. The layout of the `Calibration` directory is shown in figure 1.

As usual the `Calibration` directory contains the files required to run Socsim and the `Rcode` directory contains some code to get you started analyzing the results. But as discussed below, this exercise is different in that you will not run Socsim directly. Instead R will be used to write socsim rate files and to control the execution of Socsim. The files `fert_rates_rescaled` and `mort_rates_rescaled` are referenced by the `calibration_exercise.sup` file but are written by R code in `calibration.r`.

Since in this exercise, **Socsim is controlled by R**, after looking at the various .sup and rate files, you'll want to start getting serious with the `Rcode/calibration.r` file.

```
|-- README
|-- Rcode
|   |-- CDmort.rsave
|   |-- calibration.r
|   |-- like100.rsave
|   |-- pointsGrid.rsave
|   `-- pointsGrid400.rsave
|-- SimResults
|   |-- pop_output.omar
|   |-- pop_output.opop
|   |-- pop_output.opox
|   `-- pop_output.pyr
|-- calibration_exercise.sup
|-- fert_rates_rescaled
|-- marr_rates
|-- mort_rates_rescaled
|-- pop_starting.omar
|-- pop_starting.opop
`-- random_number
```

Figure 1: Calibration directory contents

2 Background

Little is known for certain about Imaginatia, but experts tell us that Imaginatians celebrated widely in art and literature, the fact that at the time of birth the average child had two living maternal ancestors – that is $G = P(motheralive)+P(grandmotheralive)+P(greatgrandmotheralive)+.. = 2$ This quantity is called "Josh's G". It is the expectation of the number of maternal ancestors alive at the time of the typical child's birth.

We know pretty much nothing else about the place except that:

- Its marital and non marital fertility rates are equal.
- Its marital and non marital mortality rates are equal.
- Its vital rates were constant for long periods of time.
- its population at time t-100 years looked a lot like the one in `Calibration/pop_starting.opop` and `Calibration/pop_starting.omar` .

Our task is to find a set of fertility and mortality rates that, when applied to the starting population for about 100 years, result in a population with a $G = 2.0$

3 step-by-step

At the highest level of generality, here are the steps involved:

1. **Parametrize the problem.** No matter how cool your computer is, you cannot search over the entire space of all rates. It is therefore both necessary and smart to reduce the dimensionality of the problem. Since the mortality and fertility schedules of human populations have more or less common shapes, it is reasonable to parametrize mortality and fertility rates so as to maintain the "shape" while varying the scale... sort of.
2. run Socsim several times with rates corresponding to the points on a grid of the parameters devised in step 1. Calculate and store the quantity of interest (G) and the likelihood of producing the target quantity of interest (2.0) with the given rates, for each run.

3. Identify the rates that produce the highest likelihood of producing the target and draw some cool graphs to learn about the general behavior of G over the space of rate sets investigated.

   (this is generally enough for most projects but...)

4. Using the SIR algorithm, and a prior belief about the distribution of the parameters devised in step 1, we can also calculate the *posterior probability distribution* of the parameters devised in step 1. This part is presented as an extension in the `calibration.r` file.

3.1 Parametrize the problem

To make the problem manageable, we will use two rescaling parameters, one for age-specific fertility rates and one for age-specific mortality rates. We then construct a 10X10 grid of values of these parameters spanning an interesting range. Each element of the grid corresponds to a set of fertility and mortality rates.

For the problem at hand, it is convenient to parametrize the mortality rates by e_0 and the fertility rates by the mean age at child bearing, T. But there is no deep theory in that. Other parametrization schemes are just as valid.

3.2 Run a bunch of simulations

Having constructed a grid of reasonable rate sets, we run Socsim 10 or 100 times with each set. Obviously we'll use a different random seed each time so we get some stochastic variation.

After each trial, we observe G for all individuals born in the last month of the simulation. We'll use the mean and variance of G to calculate the likelihood that Socsim, with the given rates, could have produced a G of 2.0. To do this, we waive our hands to invoke the Central Limit Theorem, and assume that the likelihood is normally distributed around the observed mean G from all trials.

This step can take a long time since it involves running a lot of simulations. In `calibration.r` there are two places where you are invited to read in pre-cooked results in order to save time.

3.3 Find maximum likelihood rates

With a matrix of likelihood values calculated over the grid of rates, it's pretty simple to just choose the best one. We'll draw some pictures as well

because pictures are cool, but also because we cannot guarantee that the likelihood surface is convex. We might want/need to look more closely at one or several regions of parameter space.

3.4 Find the posterior distribution

For most purposes, the maximum likelihood rates are what we are looking for and in most cases our prior beliefs will be only some rough bounds. This is equivalent to a uniform prior distribution over a reasonable range of scale factors ($e0$ and mean age of child bearing, T). It is not out of the question, however, that some information might lead to a more specific prior.

Either way –with a fancy prior or not, it can be useful to estimate a posterior probability. With a posterior probability distribution of the scaling parameters, one can improve one's research project by introducing an additional stochasticity into Socsim. In addition to the stochasticity inherent in the processes that Socsim simulates – that is the variability among simulation runs with identical rates but different random seeds– it is sometimes desirable to also have stochasticity that follows from our *uncertainty about the input rates*.

With a posterior distribution of the rate scaling parameters, we can run Socsim –not just with the maximum likelihood rates – but also with a set of rates corresponding to a sample of scaling parameters drawn from the posterior distribution.

We won't actually do this today, but keep it mind. Someday it might be useful.

At this point, you may wish to turn your attention to the files in the `Calibration` directory and run a "real life" example. Those with a "strong prior" that Bayesian Melding is interesting or an indiscriminate burning passion for knowledge of all kinds, will surely wish to read Emilio's excellent dissertation. Email him at emilioz@demog.berkeley.edu for a complimentary copy.