Evaluating the performance of Iterative Proportional Fitting for spatial microsimulation: new tests for an established technique

Robin Lovelace

June 5, 2014

Abstract

Iterative Proportional Fitting (IPF), also known as biproportional fitting or the RAS algorithm, is an established technique in Regional Science and has been used for a variety of purposes in the field. Primary amongst these has been the generation of small area microdata. The technique is mature, widely used and relatively straight-forward yet there have been few studies focused on evaluation of its performance. An additional research problem is the tendency of researchers to 'start from scratch', resulting in a variety ad-hoc implementations, with little evidence about the relative merits of differing approaches. Although IPF is well described mathematically, reproducible examples of the algorithm, written in modern programming languages, are rare in the academic literature. These knowledge gaps mean that answers to methodological questions must be guessed: Under what conditions can convergence be expected? How many iterations should be used for any given application? And what are the impacts of initial weights and integerisation on the final result? This paper tackles such questions, using a systematic methodology based on publicly available code and input data. The results confirm IPF's status as robust and widely applicable method for combining areal and individual-level data and demonstrate the importance of using various metrics of for determining its performance. We conclude with an agenda for future tests and offer more general guidance on how the method can be optimised to maximise its utility for future research.

Keywords: Iterative proportional fitting, spatial microsimulation, modelling
JEL Code: C

1 Notes for authors

Additonal papers to cite include:

- Harland et al. (2012)
- Smith et al. (2009)
- Cullinan (2010)
-

2 Introduction

Surveys of the composition, attitudes and behavior of individual respondents constitute a ubiquitous empirical basis of social science research. The aggregation of survey data of this type also has a long history. Generally this involves translating raw data from a 'long' format — the usual format of raw census data as it is digitised and entered into a database, where each row represents an individual and each column a separate variable — to a 'wide' data format (Van Buuren, 2012, Chapter 9). In the wide form, each row represents a geographic area or other sub-group of the total population and each column represents a variable value or bin (see Fig. 1). The wide data form is useful because it can summarise a huge amount of information in relatively little computer disk space, but has some important disadvantages. Information loss usually occurs when georeferenced individual level survey data — *spatial microdata* — is aggregated into the wide form. This problem is exacerbated when continuous variables are converted into discrete categories and when not cross-tabulations, such as age/income, are not provided. The process of aggregation is performed in three steps:

1. conversion of all variables into discrete categories (e.g. an individual with an income of £13,000 per year would be allocated to band of £10,000 to £15,000 per year)

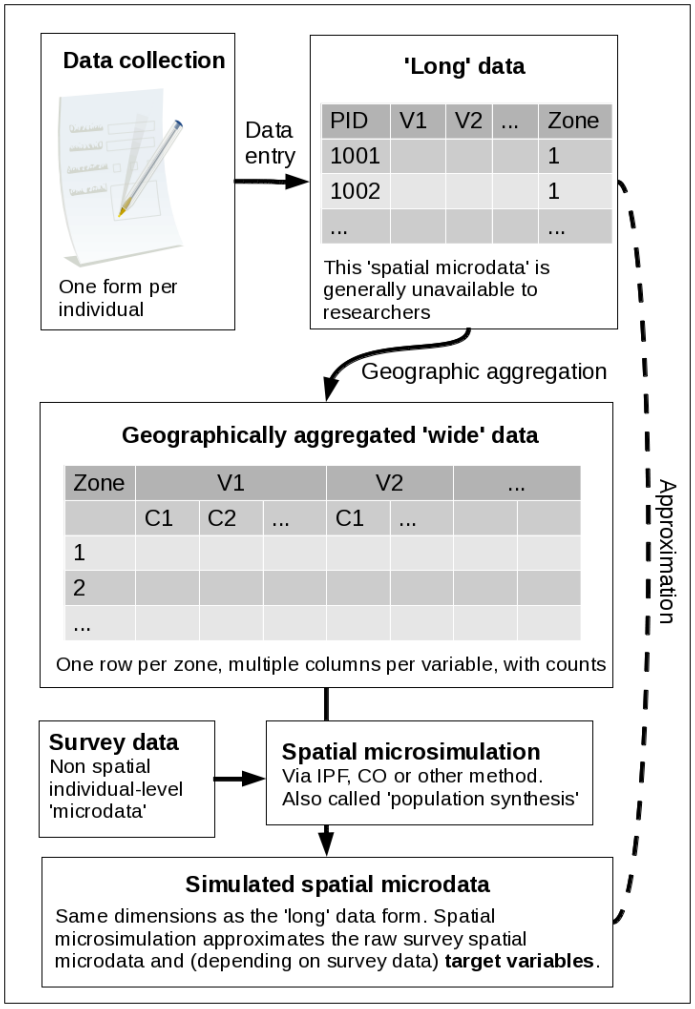

Figure 1: Schema of iterative proportional fitting (IPF) and combinatorial optimisation in the wider context of the availability of different data formats and spatial microsimulation.

2. assignment of a column to each unique category
3. assignment of integer counts to cells in each column to represent the number of people with each characteristic.

Frequently, large national social survey datasets such as censuses are *only* available in the wide, geographically aggregated form. This is problematic for the analysis of spatial phenomena (Openshaw, 1984; Lee, 2009). It is from this context of common forms of geographical data storage formats that this paper departs.

Iterative proportional fitting (IPF) can help overcome the limitations of wide, geographically aggregated data sources when used in *spatial microsimulation*. In simplistic terms spatial microsimulation can be understood as the reversal of geographic aggregation described in points 1–3 above, to approximate the original 'long' dataset (Fig. 1). Thus the need for spatial microsimulation is driven by the tendency of statistical agencies to make the raw georeferenced microdata generated by large social surveys unavailable to the research community (Lee, 2009). In fact spatial microsimulation, the term favoured by geographers and regional scientists, is generally known as *population synthesis* in transport modelling, which is an apt name: its core function is to simulate populations based on imperfect data. Spatial microsimulation can also refer to a wider approach which harnesses spatial microdata in the analysis stage (Clarke and Holm, 1987; Lovelace et al., 2014). In this paper we focus on the former meaning, which is also known as *static spatial microsimulation* (Ballas et al., 2005).

2.1 What is IPF and why use it?

In abstract terms, IPF is a procedure for assigning values to *internal cells* based on known marginal totals in a multidimensional matrix (**?**). More practically, this enables the calculation of the *maximum likelihood* estimate of the presence of given individuals individuals in specific zones. Some confusion surrounds the procedure because it has multiple names: the 'RAS algorithm', 'biproportional fitting' and to a lesser extent 'raking' have all been used to describe the procedure at different times and in different branches of academic research (Lahr and de Mesnard, 2004). 'Entropy maximisation' is a related technique, of which IPF is considered to be a subset (Johnston and Pattie, 1993; Rich and Mulalic, 2012).[1] Usage characteristics of these different terms are summarised in Table 1.

Table 1: Usage characteristics of different terms for the IPF procedure, from the Scopus database

Search term	N. citations	Commonest subject area	% Since 2010	% Mentioning "regional"
Entropy maximisation	564	Engineering	30	9
"Iterative proportional fitting"	135	Mathematics	22	16
"RAS Algorithm"	25	Comp. science	24	24
"Iterative proportional scaling"	15	Mathematics	27	0
Matrix raking	12	Engineering	50	0
Biproportional estimation	12	Social sciences	17	33

Ambiguity in terminology surrounding the procedure has existed for a long time: when Michael Bacharach began using the term 'biproportional' late 1960s, he did so with the following caveat: "The 'biproportional' terminology is not introduced in what would certainly be a futile attempt to dislodge 'RAS', but to help to abstract the mathematical characteristics from the economic associations of the model" (Bacharach, 1970). The original formulation of the procedure is generally ascribed to Deming and Stephan (1940), who presented an iterative procedure for solving the problem of how to modify internal cell counts of census data in 'wide' form, based on new marginal totals. Without repeating the rather involved mathematics of this discovery, it is worth quoting again from Bacharach (1970), who provided as clear a description as any of the procedure in plain English: "The method that Deming and Stephan suggested for computing a solution was to fit rows to the known row sums and columns to the known column sums by alternate [iterative] scalar multiplications of rows and columns of the base matrix" (p. 4). The main purpose for revisiting and further developing the technique from Bacharach's perspective was to provide forecasts of the sectoral disaggregation of economic growth models. It was not until later (e.g. Clarke and Holm, 1987) that the procedure was rediscovered for geographic research.

Thus IPF is a general purpose tool with multiple uses. In this paper we focus on its ability to *disaggregate* the geographically aggregated data typically provided by national statistical agencies to synthesise *spatial microdata*, as illustrated in Fig. 1. But why would one want do this in the first place?

To answer this question, let us briefly consider the advantages and drawbacks of the ubiquitous wide data format. The main advantages of wide datasets are that they can summarise information about very many individuals in a relatively few rows, saving storage space, easing the analysis and, critically for policy makers, ensuring the anonymity of survey respondents. But the process of aggregation and associated information loss leads to the following disadvantages:

- *Binning of continuous variables:* we do not know from a wide dataset if the aforementioned person earning £13,000 is in fact on a salary of £10,000, £15,000 or anywhere in between
- *Loss of geographical resolution:* when aggregation is done by geographical zones spatial information is lost. Instead of knowing person's precise home location, from the full dataset, we may only be able to pin them down to a relatively large administrative area.
- *Loss of cross-tabulation*: during the process of aggregation the links between different variables, often expressed as 'contingency tables', is usually lost. For example a female cyclist may be represented as additional units in the "cyclist" and "female" categories, without knowledge of their interrelation.

In the *physical sciences* both data formats are common and the original long dataset is generally available. In the social sciences, by contrast, the long format is often unavailable due to the long history of and

[1] As an aside, IPF is considered by Rich and Mulalic (2012) to be a particular efficient for of entropy maximisation: "The popularity of the IPF is therefore mainly due to the fact that it provides a solution which is equivalent to that of the ME approaches, but attained in a much more computationally efficient way" (p. 469).

confidentiality issues surrounding officially collected survey data. It is worth emphasising that National Census datasets — in which characteristics of every individual living in a state are recorded and which constitute some of the best datasets in social science — are usually only available in wide form. This poses major challenges to researchers in the sciences who interested in intra-zone variability at the local level (Whitworth, 2012; Lee, 2009).

A related long-standing research problem especially prevalent in geography and regional science is the scale of analysis. There is often a trade-off between the quality and resolution of the data, which can result in a compromise between detail and accuracy of results (Ballas and Clarke, 2003). A common scenario facing researchers investigating a spatial problem is being presented with two datasets: one long aspatial survey table where each row represents the characteristics of an individual — such as age, sex, employment status and income — and another of wide, geographically aggregated dataset containing count data for each category. IPF solves this problem when used in spatial microsimulation by proportionally assigning individuals to zones of which they are most representative.

2.2 IPF in spatial microsimulation

Spatial microsimulation tackles the aforementioned problems associated with the prevalent geographically aggregated wide data form by simulating the individuals within each administrative zone. The most detailed output of this process of spatial microsimulation is 'spatial microdata', a long table of individuals with associated attributes and weights for each zone (see Table 2 for a simple example). Because these individuals correspond to observations in the survey dataset, the spatial microdata generated in this way, of any size and complexity, can be stored with only 3 columns: person identification number (ID), zone and weight. Weights are critical to spatial microdata generated in this way, as they determine how representative each individual is of each zone. Thus, an even more compact way of storing spatial microdata taken from a single dataset is as a 'weight matrix' (table 2).

Representing the data in this form offers a number of advantages (Clarke and Holm, 1987). These include:

- The ability to investigate intra-zone variation (variation within each zone can be analysed by subsetting a single area from the data frame).
- Cross-tabulations can be created for any combination of variables (e.g. age and employment status).
- The presentation of continuous variables such as income as real numbers, rather than as categorical count data, in which there is always data loss.
- The data is in a form that is ready to be passed into an individual-level model. For example individual behaviour could simulate traffic, migration or the geographical distribution of future demand for healthcare.

These benefits have not gone unnoticed by the spatial modelling community. Spatial microsimulation is now more than a methodology, and can be considered as a research field in its own right, with a growing body of literature, methods and results (Tanton and Edwards, 2013). In tandem with these methodological developments, a growing number of applied studies has emerged to take advantage of the new possibilities opened up by such small area individual-level data, with applications as diverse as the location of health services (Tomintz et al., 2008) to the analysis of commuting (Lovelace et al., 2014). Various strategies for generating small area microdata are available, as described in recent overviews on the subject (Tanton and Edwards, 2013; Ballas et al., 2013; Hermes and Poulsen, 2012). However, it is not always clear which one is most appropriate. There has been work testing and improving (Teh and Welling, 2003) in abstract terms, but research focused on testing use the algorithm for spatial microsimulation is limited. This research gap is highlighted in the conclusions of a recent Reference Guide to spatial microsimulation (Clarke and Harding, 2013, p 270): "It would be desirable if the spatial microsimulation community were able to continue to analyse which of the various reweighting/synthetic reconstruction techniques is most accurate — or to identify whether one approach is superior for some applications while another approach is to be preferred for other applications". This paper rises to this challenge by systematically testing the IPF algorithm for spatial microsimulation. Before proceeding to the model experiments, we first provide a brief overview of the applications of IPF in regional science and outline the theory underlying the procedure based on a simple example.

2.3 Applications

Applications of IPF include using it ...

The application of greatest interest to social scientists and others working with 'ecological' data (Openshaw, 1984), however is as a method of generating spatial microdata: individual level data allocated to zones. Papers that have used IPF in this context (or a substitutable method such as simulated annealing) have explored a range of regional economic problems that purely aggregate level analyses struggle to address. These include:

- Projecting the wider impacts individual and local impacts of major job losses from individual companies (Ballas et al., 2006).
- Investigation of the local impacts of economic development in the marine energy sector in Ireland (Morrissey et al., 2013).
- Estimation of the energy impacts of freight transport at the national level, by using spatial microsimulation alongside a spatial interaction model (**?**).
- Inequalities between individuals and zones in educational opportunities and attainment (**?**).
- A model of the housing market, where outputs for a spatial microsimulation model were used as an input into an agent based model to assess the likely outcome of urban regenation interventions (**?**).
- EUROMOD, a model for analysing the impact of changes in taxation and financial laws on European citizens (**?**).
- The use of spatial microsimulation in combination with spatial interaction and agent based models to simulated the spatial distribution and movement of student populations, which are not well represented in demographic models (**?**).

3 The IPF algorithm: a worked example

In most modelling texts there is a precedence of theory over application: the latter usually flows from the former (e.g. **?**). As outlined in Section 2.1, however, the theory underlying IPF have been thoroughly demonstrated in research stemming back to the 1940s. *Practical* demonstrations, focused on making the algorithm as accessible as possible to others, are comparatively rare. For this reason this section presents a simple worked example. The steps are, to the authors knowledge, presented in more detail than any previous work, to address the theory:practice balance and ensure accessibility to understanding of the method.

That is not to say that reproducibility is completely absent in the field. Simpson and Tranmer (2005) and Norman (1999) exemplify their work with code snippets from an implementation of IPF in the proprietary software SPSS and reference to an Excel macro that is available on request, respectively. Williamson (2007) also represents good practice, with a manual for running combinatorial optimisation algorithms described in previous work. Ballas et al. (2005) demonstrate IPF with a worked example to be calculated using mental arithmetic, but do not progress to illustrate how the examples are then implemented in computer code. This section builds on and extends this work, by first outlining an example that can be completed using mental arithmetic alone before illustrating its implementation in code. A comprehensive introduction to this topic is provided in "Introducing spatial microsimulation with R: a practical" **?**. This section builds on the tutorial by providing context and a higher level description of the process.

We begin with some hypothetical data. Table 2 describes a hypothetical microdataset comprising 5 individuals, who are defined by two constraint variables, age and sex. Table 3 contains aggregated data for a hypothetical area, as it might be downloaded from census dissemination portals such as Casweb. Table 4 illustrates this table in a different form, demonstrating the unknown links between age and sex. The job of the IPF algorithm is to establish estimates for the unknown cells, which are optimized to be consistent with the known row and column marginals.

Table 5 presents the hypothetical microdata in aggregated form, that can be compared directly to Table 4. Note that the integer variable age has been converted into a categorical variable in this step, a key stage in geographical aggregation.

Using these data it is possible to readjust the weights of the hypothetical individuals, so that their sum adds up to the totals given in Table 4 (12). In particular, the weights can be readjusted by multiplying them by the marginal totals, originally taken from Table 3 and then divided by the respective marginal total in

Table 2: A hypothetical input microdata set (the original weights set to one). The bold value is used subsequently for illustrative purposes.

Individual	Sex	Age	Weight
1	Male	59	1
2	Male	54	1
3	Male	35	**1**
4	Female	73	1
5	Female	49	1

Table 3: Hypothetical small area constraints data (s).

Constraint ⇒	i		j	
Category ⇒	i_1	i_2	j_1	j_2
Area ⇓	Under-50	Over-50	Male	Female
1	8	4	6	6

5. Because the total for each small-area constraint is 12, this must be done one constraint at a time. This can be expressed, for a given area and a given constraint (i or age in this case), as follows:

$$w(n+1)_{ij} = \frac{w(n)_{ij} \times sT_i}{mT(n)_i} \tag{1}$$

where $w(n+1)_{ij}$ is the new weight for individuals with characteristics i (age, in this case), and j (sex), $w(n)_{ij}$ is the original weight for individuals with these characteristics, sT_i is element marginal total of the small area constraint, s (Table 3) and $mT(n)_i$ is the marginal total of category j of the aggregated results of the weighted microdata, m (Table 5). n represents the iteration number. Although the marginal totals of s are known, its cell values are unknown. Thus, IPF estimates the interaction (or cross-tabulation) between constraint variables. (Follow the emboldened values in the tables to see how the new weight of individual 3 is calculated for the sex constraint.) Table 6 illustrates the weights that result. Notice that the sum of the weights is equal to the total population, from the constraint variables.

After the individual level data have been re-aggregated (Table 7), the next stage is to repeat Eq. (1) for the age constraint to generate a third set of weights, by replacing the i in sT_i and $mT(n)_i$ with j and incrementing the value of n:

$$w(3)_{ij} = \frac{w(2)_{ij} \times sT_j}{mT(2)_j} \tag{2}$$

Apply Eq. (2) to the information above and that presented in Table 7 results in the following vector of new weights, for individuals 1 to 5:

$$w(3) = (\frac{6}{5}, \frac{6}{5}, \frac{18}{5}, \frac{3}{2}, \frac{9}{2}) \tag{3}$$

As before, the sum of the weights is equal to the population of the area (12). Notice also that after each iteration the fit between the marginal totals of m and s improves. The total absolute error (TAE) from $m(1)$ to $m(2)$ improves from 14 to 6 in Table 5 and Table 7 above. TAE for $m(3)$ (not shown, but calculated by aggregating $w(3)$) improves even more, to 1.3. This number would eventually converge to 0 through subsequent iterations, as there are no empty cells in the input microdataset, a defining feature of IPF (see Section 6.4 for more on the impact of empty cells).

The above process, when applied to more categories (e.g. socio-economic class) and repeated iteratively until a satisfactory convergence occurs, results in a series of weighted microdatasets, one for each of the

Table 4: Small area constraints expressed as marginal totals, and the cell values to be estimated.

Marginal totals	j			
	Age/sex	Male	Female	T
i	Under-50	**?**	?	8
	Over-50	?	?	4
	T	6	6	12

Table 5: The aggregated results of the weighted microdata set ($m(1)$). Note, these values depend on the weights allocated in Table 2 and therefore change after each iteration

Marginal totals		j		
	Age/sex	Male	Female	T
i	Under-50	**1**	1	2
	Over-50	2	1	3
	T	3	2	5

Table 6: Reweighting the hypothetical microdataset in order to fit Table 3.

Individual	Sex	age-group	Weight	New weight, w(2)
1	Male	Over-50	1	$1 \times 4/3 = \frac{4}{3}$
2	Male	Over-50	1	$1 \times 4/3 = \frac{4}{3}$
3	Male	Under-50	1	$\mathbf{1} \times \mathbf{8}/\mathbf{2} = 4$
4	Female	Over-50	1	$1 \times 4/3 = \frac{4}{3}$
5	Female	Under-50	1	$1 \times 8/2 = 4$

small areas being simulated. This allows for the estimation of variables whose values are not known at the local level (e.g. income) (Ballas et al., 2005). An issue with the results of IPF (absent from combinatorial optimisation methods), however, is that it results in non-integer weights: fractions of individuals appear in simulated areas. This is not ideal for certain applications, leading to the development of strategies to 'integerise' the fractional weights illustrated in Table 7. Methods for performing integerisation are discussed in (Ballas et al., 2005), (Pritchard and Miller, 2012) and, most recently in (Lovelace and Ballas, 2013), who systematically benchmarked and compare different procedures. It was found that the 'truncate, replicate, sample' (TRS) and 'proportional probabilities' methods were most accurate. These methods are therefore used to test the impact of integerisation in this paper.

3.1 Implementing IPF in R

The above example is best undertaken by hand, probably with a pen and paper to gain an understanding of IPF, before the process is automated for larger datasets. This section explains how the IPF algorithm described above can be implemented in R.

4 Evaluation techniques

To verify the integrity of any model, it is necessary to compare its outputs with empirical observations or prior expectations gleaned from theory. The same principles apply to spatial microsimulation, which can be evaluated using both *internal* and *external* validation methods (Edwards and Clarke, 2009). *Internal validation* is the process whereby the aggregate-level outputs of spatial microsimulation are compared with the input constraint variables.

Internal validation typically tackle such issues as "do the simulated populations in each area microdata match the total populations implied by the constrain variables?" and "what is the level of correspondence between the cell values of different attributes in the aggregate input data and the aggregated results of spatial microsimulation?" A related concept from the agent based modelling literature is *verification*, which asks the more general question, "does the model do what we think it is supposed to do?" (**?**, p. 131) *External validation* is the process whereby the variables that are being estimated are compared to data from another

Table 7: The aggregated results of the weighted microdata set after constraining for age ($m(2)$).

Marginal totals		i		
	Age/sex	Male	Female	T
j	Under-50	4	4	8
	Over-50	$\frac{8}{3}$	$\frac{4}{3}$	4
	T	$6\frac{2}{3}$	$5\frac{1}{3}$	12

source, external to the estimation process, so the output dataset is compared with another known dataset for those variables. Both validation techniques are important in modelling exercises for reproducibility and to ensure critical advice is not made based on coding errors (**?**).

This section outlines the evaluation techniques that are commonly used in the field, with a strong focus on internal validation. The options for external validation are heavily context dependent so should be decided on a case-by-case basis. External validation is thus mentioned in passing with a focus on general principles rather than specific advice. The methods are presented in ascending order of complexity and roughly descending order of frequency of use, ranging from the coefficient of determination (R^2) to entropy based measures. Before these methods are described, it is worth stating that the purpose of this section is not to find the 'best' evaluation metric: each has advantages and disadvantages, the importance of which will vary depending on the nature of the research. The use of a variety of techniques in this paper is of interest in itself. The high degree of correspondence between them (presented in Section 7), suggests that researchers need only to present one or two metrics (but should try more, for corroboration) to establish relative levels of goodness of fit between different model configurations.

The problem of internal validation (described as 'evaluating model fit') was tackled in detail by **?**. Chi-squared, Z-scores, absolute error and entropy-based methods were tested, with no clear 'winner' other than the advocacy of methods that are robust, fast and easy to understand (absolute error measures match these criteria well). The selection of evaluation methods is determined by the nature of available data: "Precisely because a full set of raw census data is not available to us, we are obliged to evaluate our simulated populations by comparing a number of aggregated tables derived from the synthetic data set with published census small-area statistics" (**?**, p. 178). The same reasoning is used to select evaluation methods here.

4.1 Descriptive statistics

Useful basic statistics to describe and compare datasets are the mean, median and standard deviation of cell counts in the simulated and known aggregate-level database. These are useful 'sanity checks': if the mean population across all zones or for a particular constraint differs greatly from expectations, there is clearly a problem. These summary statistics are dependent on the size of the dataset. To compare datasets of widely varying populations, dimensionless statistics can be used. The coefficient of variation, the ratio of the standard deviation to the mean is an example of a dimensionless statistic robust to population size.

4.2 Scatter plots

A scatter plot of cell counts for each category for the original and simulated variables is a basic but very useful preliminary diagnostic tool in spatial microsimulation (Ballas et al., 2005; Edwards and Clarke, 2009). In addition, marking the points, depending on the variable and category which they represent, can help identify which variables are particularly problematic, aiding the process of improving faulty code (sometimes referred to as 'debugging').

In these scatter plots each data point represents a single zone-category combination (e.g. 16-25 year old female in zone 001), with the x axis value corresponding to the number of individuals of that description in the input constraint table. The y value corresponds to the aggregated count of individuals in the same category in the aggregated (wide) representation of the spatial microdata output from the model. This stage can be conducted either before or after *integerisation* (more on this below). As taught in basic statistics courses, the closer the points to the 1:1 (45 degree, assuming the axes have the same scale) line, the better the fit.

4.3 The coefficient of determination

The coefficient of determination (henceforth R^2) is the square of the Pearson correlation coefficient for describing the extent to which one continuous variable explains another. R^2 is a quantitative indicator of how much deviation there is from this ideal 1:1 line. It varies from 0 to 1, and in this context reveals how closely the simulated values fit the actual (census) data. An R^2 value of 1 represents a perfect fit; an R^2 value close to zero suggests no correspondence between then constraints and simulated outputs (i.e. that the model has failed). We would expect to see very high R^2 values for internal validation (the constraint) and a strong positive correlation between target variables that have been estimated and external datasets (e.g. income).

4.4 Total Absolute and Standardized Error (TAE and SAE)

Total absolute error (TAE) is the sum of the absolute (positive) discrepancies between the simulated and known population in an area. The standardized absolute error (SAE) is simply TAE divided by the total population. An issue with SAE has been that, by using the total population as the denominator for TAE calculations on a subset of cells, the method can understate the size of the errors (Edwards and Clarke, 2009). TAE provides no level of significance, so thresholds have been used. Clarke and Madden (2001) used an error threshold of 80 per cent of the areas with less than 20 per cent error (SAE <0.20). Smith et al. (2007) work with a model that simulates with diabetes and use an error threshold of 10% (SAE <0.1) in 90 per cent of the output areas.

4.5 Z scores

Z-scores apply to individual cells (for a single constraint in a single area). and also provide a global statistic on model fit (**?**).

5 Input data

Three input datasets form the basis of the scenarios tested in this paper, 'simple', 'small-area' and 'sheffield'. Each consists of a set of aggregate constraints and corresponding individual-level survey tables. The latter are anonymous and have been 'scrambled' to enable their publication online. This enables the results to be replicated by others, without breaching the user licenses of the data. To this end, all the input data (and code) used to generate the results reported in the paper are available online on the code-sharing site GitHub. The simplest scenario consists of 6 zones which must be populated by the data presented in the previous section (see Table 2). This scenario is purely hypothetical and is used to represent the IPF procedure and the tests in the simplest possible terms. Its small size allows the entirety of the input data to be viewed in a couple of tables (see Table 2 and Table 8, the first row of which is used to exemplify the method the previous section). As with all data used in this paper, these tables can be viewed and downloaded online.

Table 8: Geographically aggregated constraint variables for the simple scenario.

	Age constraint		Sex constraint	
Zone	16–49	50+	m	f
1	8	4	6	6
2	2	8	4	6
3	7	4	3	8
4	5	4	7	2
5	7	3	6	4
6	5	3	2	6

The next smallest input dataset, and the most commonly used scenario in the paper, is 'small-area'. The area constraints consist of 24 Output Areas (the smallest administrative unit in the UK), with an average adult population of 184. Three files describe the number of hours worked, the marital status and the tenancy of these individuals (although tenancy in fact counts houses, not people): the files can be accessed from 'small-area-eg', within the 'input-data' folder of the online repository.[2] A sample of these constraint variables is presented in Table 9 below.

Table 9: Sample of constraint variables from the first 5 rows of the small-area-eg input dataset.

The individual-level table that compliments these geographically aggregated count constraints is taken from Understanding Society wave 1. These have been randomised to ensure anonymity but they are still

[2]These datasets can be found here: https://github.com/Robinlovelace/IPF-performance-testing/tree/master/input-data/small-area-eg

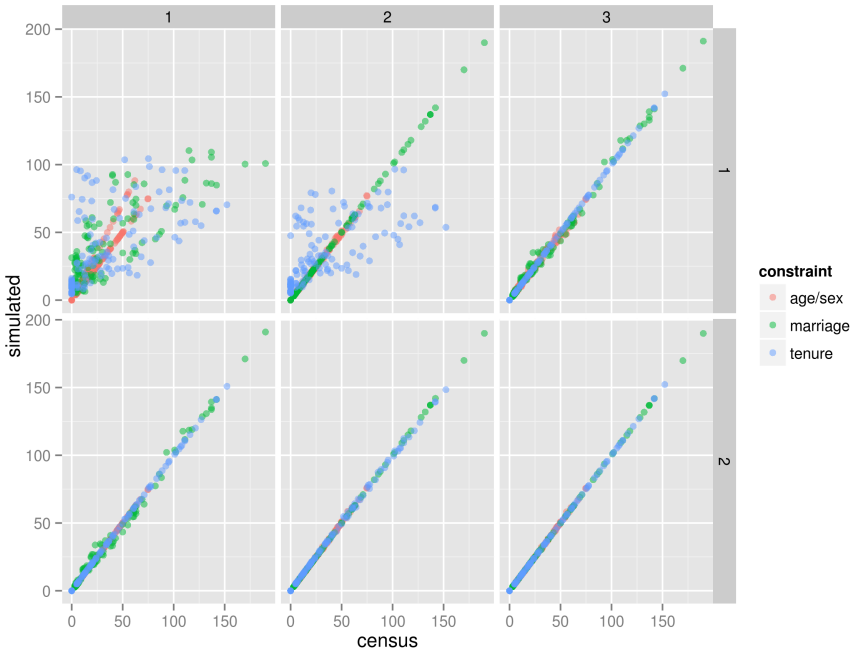

Figure 2: IPF in action: simulated vs census zone counts for the small-area baseline scenario after each constraint (columns) and iteration (rows).

roughly representative of real data. The table consists of 1768 rows of data and is stored in R's own space-efficient filetype (see "ind.RData", which must be opened in R). This individual level data consists of the same 3 constraint variables as provided in Table 9 in addition to one 'target variable': monthly pre-tax income in pounds stirling.

6 Method: model experiments

6.1 Baseline scenarios

In order to perform a wide range of model experiments, in which only one factor is altered at a time, baseline scenarios were created. These are model runs with sensible defaults, reference points against which alternative runs are compared. The baseline scenarios consist of 3 iterations. In the 'simple' scenario (with 2 constraints), there are 6 constraint-iteration permutations. Table 10 shows fit improving with each step.[3] (

Table 10: Results for the baseline 'simple' scenario. Generated through by the 'analysis' script, and available as a .csv file online.

The baseline 'small-area' scenario consists of three iterations of IPF, in which all three constraints were used. The results for all 9 constraint-iteration permutations of this baseline scenario are illustrated in Fig. 2 and in the online 'measures' table. These results show that IPF successfully brings the simulated cell counts in-line with the census after each constraint is applied and that the fit improves with each iteration (only two iterations are shown here, as the fit improves very little beyond two, and to avoid clutter).

The baseline 'Sheffield' scenario consists of the same basic set-up, but with 4 constraints. As before, the results show convergence towards better model fit, although the fit does not always improve from one constraint to the next (see the online 'measures' table).

[3]The exception is $p5$, which reaches zero in the second iteration. *pcor* continues to improve, even though the printed value, to 6 decimal places, is indistinguishable from 1 after two complete iterations.

6.2 Constraints and iterations

These were the simplest model experiments, requiring only the reporting of how the results changed with more or fewer iterations and re-arrangement of to apply constraints in a different order. 1, 3 and 10 are reported in the final results and 'convergence graphs' were created to show how model fit improved from one iteration-constraint combination to the next.

To alter the number of iterations used in each model, one simply changes the number in the following line of code (the default is 3):

```
num.its <- 3
```

The code for running the constraints in a different order is saved in 'etsim-reordered.R'.

6.3 Initial weights

The impact of initial weights was tested in two ways: first, the impact of changing a sample of one or more initial weight values by a set amount was explored. Second, the impact of gradually increasing the size of the initial weights was tested, by running the same model experiment several times, but with incrementally larger initial weights applied to a pre-determined set of individuals.

In the simplest scenario, the initial weight of a single individual was altered repeatedly, in a controlled manner. The code enabling this model experiment is contained in the file 'etsim-weights.R'. The number of different initial weight runs can be controlled by the parameter 'num.ws' which is set to 5 as an arbitrary default. Initial weights are set to k, a vector of length 'num.ws' which are by default set as a function of 'u' which is in term defined as the sequence 1:num.ws:

```
k <- u/5 + 0.5 # initial weight of sample individuals for testing
```

In this case the resultant weights are set to 0.7, 0.9, 1.1, 1.3 and 1.5 for each model experiment. For factors of one million, for example, the above would be replaced by `k <- u * 10^6 yrm`. To observe the impacts of altering the weights in this way the usual tests of model fit were conducted. In addition, the capacity of initial weights to influence an individual's final weight/probability of selection was tested by tracing its weight into the future after each successive constraint and iteration. This enabled plotting original vs simulated weights under a range of conditions (Section 7).

6.4 Empty cells

Empty cells refer to combinations individual-level attributes that are not represented by any single individual in the survey input data. An example of this would be if no female in the 16 to 49 year age bracket were present in Table 2. Formalising the definition, survey dataset that contains empty cells may be said to be *incomplete*; a survey dataset with no empty cells is *complete*.

No method for identifying whether or not empty cells exist in the individual-level dataset for spatial microsimulation has been previously published to the authors' knowledge. We therefore start from first principles. The number of different constraint variable permutations ($Nperm$) is defined by Eq. (4), where $n.cons$ is the total number of constraints and $n.cat_i$ is the number of categories within constraint i:

$$Nperm = \prod_{i=1}^{n.cons} n.cat_i \tag{4}$$

To exemplify this equation, the number of permutations of constraints in the *simple* example is 4: 2 categories in the sex variables multiplied by 2 categories in the age variable. Clearly, $Nperm$ depends on how continuous variables are binned, the number of constraints and diversity within each constraint. Once we know the number of unique individuals (in terms of the constraint variables) in the survey ($Nuniq$), the test to check a dataset for empty cells is straightforward, based on Eq. (4):

$$is.complete = \left\{ \begin{array}{ll} TRUE & \text{if } Nuniq = Nperm \\ FALSE & \text{if } Nuniq < Nperm \end{array} \right\} \tag{5}$$

Once the presence of empty cells is determined in the baseline scenarios, the next stage is to identify which individuals are missing from the individual-level input dataset (Ind) — the combination of attributes that no individual in Ind has. To do this, we created an algorithm to generate the the complete input dataset ($Ind.complete$). The 'missing' individuals, needed to be added to make Ind complete can be defined in terms of set theory by Eq. (6):

$$Ind.missing = Ind.complete \setminus Ind \tag{6}$$

This means simply that the missing cells are defined as individuals with constraint categories that are present in the complete dataset but absent from the input data. The theory to test for and identify empty cells described above is straightforward to implement in practice with R, using the commands `unique` and `%in%` respectively (see the 'empty-cells.R' script in the 'model' folder of our code repository).

To test the impact of empty cells on model results, an additional empty cell was added to the input dataset in each example.[4] Experiments in which empty cells were removed through the addition of individuals with attributes defined by Eq. (6) were also conducted, and the model fit was compared with the baseline scenarios. The baseline scenarios were run using these modified inputs.

6.5 Integerisation

This was the next variable that was altered, following (Lovelace and Ballas, 2013).

7 Results

7.1 Baseline scenarios

As expected from Fig. 2 above, fit improves over time. This improvement generally happens for each constraint and always from one iteration to the next. Through the course of the first iteration, R^2 improves from 0.67 to 0.9981. As illustrated in Fig. 3, the marginal improvement reduces from one iteration to the next: at the end of iteration 2, R^2 improved to 0.999978.

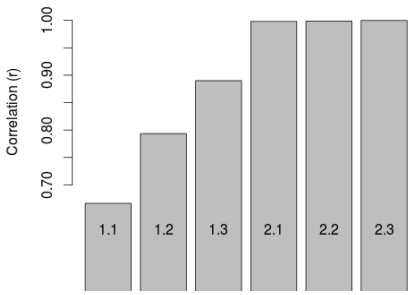

Figure 3: Improvement in goodness of fit with additional constraints and iterations, as illustrated by R^2 values.

7.2 Constraints and iterations

As illustrated above, the number of iterations required to reach a very close fit between the census and simulated data is very low, with rapid reductions in the marginal accuracy return on additional computing power. After four iterations, the result in R is indistinguishable from 1, with its default accuracy of 7 decimal places. The decay in the rate of improvement with additional iterations and constraints was found to super-exponential.

7.3 Initial weights

It was found that initial weights had little impact on the overall simulation. Taking the simplest case, a random individual (id = 228) was selected from the 'small area' data and his/her initial weight was set to 2 (instead of the default of 1). The impact of this change on the individual's simulated weight decayed rapidly, tending to zero after only 5 iterations in the small-area scenario. It was found that altered initial weights affect the simulated weights of all other individuals; this is illustrated in Fig. 4, which shows the impact on individual 228 and a randomly selected individual whose initial weight was not altered, for all areas.

Fig. 4 shows that changing initial weights for individuals has a limited impact on the results that tends to zero from one iteration to the next. After iteration one the impact of a 100% increase in the initial weight of individual 228 had reduced to 0.3%; after 3 iterations the impact was negligible. The same pattern can

[4] In the 'simple' scenario, this simply involves removing a single individual from the input data, which equates to deleting a single line of code (see 'empty-cells.R' in the 'simple' folder to see this).

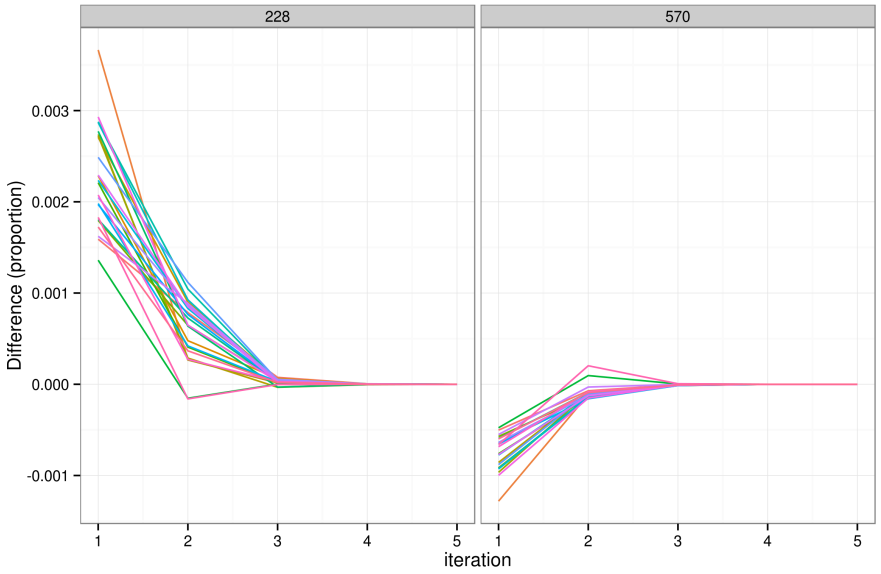

Figure 4: The impact of changing the initial weight of individual 228 on the simulated weights of two individuals after 5 iterations. Each of the 24 coloured lines represents a single zone.

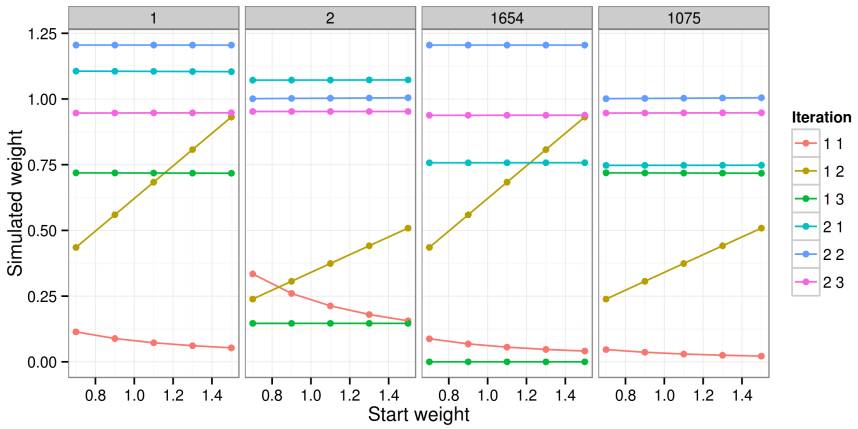

Figure 5: Initial vs simulated weight of four individuals for a single area (zone 1).

be observed in the control individual 570, although the impact is smaller and in the reverse direction. The same pattern can be observed for other individuals, by altering the 'analysis.min'.

To analyse the impacts of changing the initial weights of multiple individuals, the same experiment was conducted, but this time the initial weights were applied to the first five individuals. The results are illustrated in Fig. 5, which shows the relationship between initial and final weights for the first two individuals and another two individuals (randomly selected to show the impact of altering weights on other individual weights).

Fig. 5 shows that the vast majority of the impact of altered starting weights occurs in iteration-constraint combinations 1.1 and 1.2, before a complete iteration has even occurred. After that, the initial weights continue to have an influence, of a magnitude that cannot be seen. For individual 1, the impact of a 53% difference in initial weights (from 0.7 to 1.5) shrinks to 2.0% by iteration 1.3 and to 1.1% by iteration 2.3.

These results suggest that initial weights have a limited impact on the results of IPF, that declines rapidly with each iteration.

7.4 Empty cells

There were found to be 0, 62 and 8,105 empty cells in the simple, small-area and sheffield examples respectively. Only the smallest input dataset was *complete*. Due to the multiplying effect of more variables (Eq. (4)), *Nperm* was found to be much larger in the complex models, with values rising from 4 to 300 and 9504 for the increasingly complex test examples.

As expected, it was found that adding additional empty cells resulted in worse model fit, but only slightly worse fit if few empty cells (1 additional empty cell to 10% non-empty cells) were added. The addition of new rows of data with attributes corresponding to the empty cells to the input dataset allowed the small-area and sheffield models to be re-run without empty cells. This resulted in substantial improvements in model fit that because relatively more important with each iteration: model runs with large numbers of empty cells reached minimum error values quickly, beyond which there was no improvement.

7.5 Integerisation

7.6 Summary of results

The impact of each model experiment on each of the three scenarios is summarised in Table 11.

Table 11: Summary results of root mean square error (RMSE) between simulated and constraint data.

Model test	Simple	Small area	Sheffield
Baseline	0.0001	0.018	25.5
Constraints and iterations			
1 iteration	0.221	1.91	78.2
5 iterations	<0.000001	0.00001	20.9
10 iterations	<0.000001	0.000002	13.8
Reordered constraints	0.0001	0.018	14.5
Initial weights			
10% sample set to 10	0.0001	0.034	25.5
10% sample set to 1000	0.0001	0.998	25.5
Empty cells			
1 additional empty cell	0.852	0.018	25.5
10% additional empty cells	—	0.050	25.6
No empty cells	—	0.005	1.24
Integerisation			
TRS integerisation	0.577	3.68	29.9
Proportional probabilities	1.29	3.91	30.0

8 Conclusions

In terms of empty cells, we have demonstrated a method for identifying whether they are present in the input data. More important for the spatial microsimulation community, we have shown for the first time (to our knowledge) that empty cells have a large adverse effect on model fit. The removal of empty cells greatly improved model fit. This finding shows that identifying and removing empty cells, as demonstrated in this paper and accomanying code, can substantially enhance the performance of spatial microsimulation models, in cases when many complex constraint variables are used and many empty cells are present. It is striking that in the most complex example (sheffield), the improvement in model fit from removing empty cells was far greater than that resulting from additional iterations. A direct recommendation from these findings is therefore that researchers should test for missing cells, as defined in Section 6.4, before reweighting.

References