Hierarchical Population Generation in Transportation Modelling

Daniele Casati

Master Thesis **9XX**

Institute for Transport Planning and Systems **September 2014**

IVT Institut für Verkehrsplanung und Transportsysteme
Institute for Transport Planning and Systems

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Contents

List of Figures

List of Tables

Master Thesis

Hierarchical Population Generation in Transportation Modelling

Daniele Casati
IVT, ETH Zürich,
CH-8093 Zürich
daniele.casati@student.ethz.ch

Kay W. Axhausen et al.
IVT, ETH Zürich,
CH-8093 Zürich

September 2014

Abstract

Synthetic populations are necessary for agent-based travel models. However, conventional approaches based upon reweighting data from an available small sample suffer from low heterogeneity; simulation-based methods are promising, but have not so far taken into account hierarchies in the data.

The aim of this thesis is to develop a simulation-based approach for population synthesis that allows to consider hierarchies, as in the case of persons and households, and to apply it in a case study.

The current state of the art of hierarchical population generation is first reviewed, with a special attention to the theoretical foundation of each technique.

An approach belonging to the family of Markov Chain Monte Carlo methods is chosen for implementation, offering several advantages compared to the popular Iterative Proportional Fitting. MCMC has only recently been applied for this purpose in transportation modelling, even though it has a long history in other fields. It is a two-step procedure based upon fitting conditionals and Gibbs sampling, which exploits these conditionals in a Markov chain to simulate new agents.

In this thesis, MCMC is both generalised (iMCMC) and extended to handle hierarchies (hMCMC). A careful analysis of the results proves the validity of the developed approach.

Besides, some advanced data visualisation and clustering techniques are implemented to study the available demographic sample. Firstly, Multiple Correspondence Analysis, the categorical version of the well-known Principal Component Analysis, which finds a new basis system highlighting the distribution of the data points, rather than simply using the original variables. Secondly, Self-Organising Map, to produce low-dimensional views of the considered high-dimensional data which allow to easily identify the present clusters. Thirdly, Chow-Liu Tree, which approximates the joint probability distribution underlying the data sample by finding the most evident interdependencies among its variables.

These methods allow to distinguish the characteristics of the clusters into which the available data sample is divided. It is also illustrated how they can become further lines of research for the problem of population generation, by fitting a model for the joint probability distribution.

Keywords

Markov Chain Monte Carlo, Population Synthesis, Hierarchical Generation, Joint Probability Distribution, Conditional Distributions, Categorical Variables, Data Visualisation, Cluster Analysis

Preferred citation style

Casati, D. (2014) Hierarchical Population Generation in Transportation Modelling, *Master Thesis*, ETH Zurich, Zurich.

1 Introduction

The first step to create effective transportation models is a realistic population of persons who want to perform some activities and, because of them, need to travel between certain locations.

Nowadays, a large amount of personal data has continuously been assembled, and it may seem easy for researchers to have demographic data upon which to create their models. However, the so-called *big data* can be difficult to obtain for transport modellers because of many reasons, like simple inaccessibility or doubts about representativeness. Besides, the census data featuring information on the level of individuals and households which are available to academic research do not only constitute a reduced sample of the total, but, due to privacy concerns, have also usually been treated with anonymisation techniques.

Thus, there is a need for effective techniques to synthesise realistic large datasets approximating the unknown full population, given a small data sample. This task is called *population generation*.

Besides, if present in the full population, these new datasets should also be characterised by a *hierarchical structure*, which, in a demographic context, means that the simulated persons should be gathered into households. However, other kinds of individuals and hierarchies can also be considered, such as employees and firms. Hierarchies are vital for transportation modelling since one usually want to simulate joint decisions.

However, if resampling were the only goal, random draws with replacement of agents or full households from the reference dataset could have been considered. The intrinsic flaws of this generation are (1) an amplification of the biased characteristics of the reference sample, and (2) a lack of heterogeneity of the new agents, being simple copies of the original ones.

Since the aim of this thesis is not a trivial cloning of the reference sample, but to develop a methodology for synthesising new realistic hierarchical populations based upon some probabilistic model derived from these data, it was necessary to proceed differently from most of the current state of the art, which is summarised in Section 2. This review is divided into two parts, one illustrating methods for plain population generation (Section 2.1) and the other taking hierarchies into account (Section 2.2).

Section 3 then summarises advantages and disadvantages of the reviewed techniques. To allow heterogeneous new populations, a promising simulation-based method belonging to the family of *Markov Chain Monte Carlo* algorithms, only recently applied to this field, was chosen; however, this technique did not consider hierarchies in the data.

In Section 4 the methodology of this approach is carefully explored, together with the developed generalisation and extension for a hierarchical synthesis.

To prove the validity of these ideas, a case study was consequential. The characteristics of the employed dataset are explored in Section 5, based upon Singapore demographics.

Details about the implementation of plain and hierarchical population generation are then described in Sections 6.1 and 7.1, while an analysis of their results is illustrated in Sections 6.2 and 7.2.

The conclusions in Section 8 summarise the developed procedure and also describe other lines of research yet to be implemented.

An extensive appendix follows:

Appendix A gives some definitions about *categorical variables*, through which demographic datasets are usually expressed.

Appendix B provides with theoretical details on both MCMC and Iterative Proportional Fitting, the main method for population generation in the literature on transportation modelling (not considered for implementation).

Appendix C discusses the applied data visualisation and clustering techniques, which were very helpful to fix some issues of the implemented methodology and are not generally mentioned in the literature on transportation modelling.

Appendix D then briefly introduces promising lines of research in population generation as a matter of estimating a joint probability distribution on some high-dimensional data. The classic method for this purpose is first reviewed, followed by some possible extensions for population synthesis of a data mining technique explored in Appendix C.

Appendix E details an approach from the literature which did not prove successful in the discussed case study, generating hierarchies by connecting populations as a post-processing operation.

Appendix F finally contains a `Java` code implementing some key steps of the employed MCMC-based approach for population generation, also valid for the hierarchical case.

2 Literature Review

This section covers the techniques explored in the literature on transportation modelling, firstly for plain population generation (Section 2.1) and secondly to also deal with hierarchies (Section 2.2). The theoretical foundation of each technique is discussed together with its advantages and drawbacks, in order to provide an overall idea about the solutions to the problem faced by this thesis and justify why the implemented method was chosen.

2.1 Population Generation

In this section the most important methods being currently applied in the field of transportation modelling for plain population generation, i.e. without hierarchies, are described:

Iterative Proportional Fitting The milestone for population generation, although this is not the aim for which it was originally developed in 1940. Its success (Farooq et al., 2013; Barthelemy and Cornelis, 2012; Pritchard and Miller, 2012; Anderson et al., forthcoming; Müller and Axhausen, 2011a,b; Ortúzar and Willumsen, 2001) is also due to the limited popularity of other methods, like *Generalised Raking* (Deville et al., 1993). Its population generation is based upon the equivalence of the underlying joint probability distribution of the full population with the contingency table fitted by IPF. In fact, this method only works with categorical variables.

Combinatorial Optimisation It is much less used than IPF, but seems to be the only other technique for population generation extensively discussed in the literature (Farooq et al., 2013; Barthelemy and Cornelis, 2012; Pritchard and Miller, 2012; Anderson et al., forthcoming; Müller and Axhausen, 2011a,b). It is based upon random draws with replacement.

Markov Chain Monte Carlo A technique which makes use of this algorithm differently from how it is commonly employed (similarly to IPF), as it is discussed in Appendix B.2. The main reference article for this method applied for population generation is Farooq et al. (2013). It synthesises populations by exploiting conditional distributions of the attributes.

2.1.1 Iterative Proportional Fitting

Iterative Proportional Fitting is the usual method employed in transportation modelling to generate a large population of agents from a small sample, although it can only be applied to categorical data. This type of data is actually the most common among demographic samples, which are in turn the usual datasets treated with IPF in transportation modelling; however, IPF is also used in other fields, where it is known as *biproportional fitting*, *RAS algorithm*, or *matrix raking* or *scaling* (Bacharach, 1965).

A contingency table is such to contain the counts of agents present in a dataset for every possible combination of the considered attributes. The main purpose of IPF is to adapt the contingency table of a reduced data sample so that it respects some other known *control totals* of the full population; this is a more general term than marginals, referring to counts obtained when the table is collapsed along all dimensions except one (it is not necessary that they form whole distributions). In this way, both sources of information can be used.

Indeed, IPF is an iterative algorithm which proceeds in such a way that these control totals remain fixed. More details about this aspect are given in Appendix B.1, where a relevant assumption of this method is also illustrated, i.e. that the IPF fitted contingency table can be written as an outer product of two vectors.

Since this contingency table can be treated as an approximation of the joint distribution of the considered categorical variables, it is possible to sample new agents according to it, i.e. to effectively generate populations. However, it is also clear that the way this technique is used in transportation modelling, i.e. population generation, is not the main purpose for which it has been developed.

There are two main approaches to synthesise new populations from the IPF contingency table:

1. New agents can simply be copies of the coordinates with non-zero entries in the fitted contingency table, generated as many times as their corresponding weight.
2. Monte Carlo draws may sample new agents, given the IPF contingency table as equilibrium probability distribution.

The first method returns a single new population which can change size, but whose composition always remains the same; hence, it is not a simulation-based approach. Nevertheless, even the second method still provides multiple new datasets too much dependent upon the reference data (see also the paragraph below).

Thus, for population generation—which, as previously stated, is not the original aim of IPF—this method is based on two main assumptions:

- It focuses on keeping the marginal distributions of the new populations equal to the known control totals of the full population, something which limits the heterogeneity of the new populations. Besides, if there are no control totals of the full population, they should be replaced with marginals of the reference sample in IPF, therefore making its application, in a certain way, pointless, since its original purpose would completely be missing.
- It relies upon the already mentioned model of independence, something which does not emphasise the relationships among variables.

However, these conditions are actually reached only when the loop IPF finally converges, and a very large amount of steps can be required to arrive at it, or it can even be only an asymptotic limit. This is another disadvantage of IPF.

Zero-Cell Problem

A further shortcoming of this algorithm concerns null entries in the reference contingency table: combinations that were unobserved due to being impossible, called *structural zeros*, cannot be distinguished from those that were unobserved because of the limited size of the sample, referred to as *sampling zeros*.

Indeed, these zeros always produce null entries in the fitted contingency table as well, and hence they do not allow to sample any new agent with these characteristics, heavily affecting the heterogeneity of the new populations. Inconsistencies can also arise when a whole row or column of the reference contingency table is full of zeros and there are corresponding marginals of the full population which are not.

This *zero-cell problem* is usually prevented by inserting some nonzero, low random values in the cells of the sampling zeros; however, the task to determine if a zero is structural or sampling is left to the researcher's judgement. Another way to overcome this flaw is through Markov Chain Monte Carlo techniques for Bayesian inference (Appendix B.2).

Nevertheless, it is thanks to this "problem" that IPF, a very simple algorithm, does not usually suffer from the *curse of dimensionality* of becoming infeasible if the number of possible different combinations is too large; Section 6.2.1 deals with this problem in greater detail. In fact, the time complexity of IPF linearly depends upon the number of nonzero cells in the reference contingency table (since the others remain zero), thus obtaining a remarkable computational advantage. With many attributes and few null cells, IPF would not only reach a reasonable convergence in too many loop steps, but it would also become unstable: on computing machines, numerical errors would propagate and make the final result very different from the "real" IPF contingency table, obtainable only with exact arithmetic (which is usually infeasible). Besides, thanks to the null cells some clever forms to store the *sparse* contingency table can be adopted, as suggested by Pritchard and Miller (2012); Müller and Axhausen (2011b).

2.1.2 Combinatorial Optimisation

In this section another method for population generation from the literature is briefly described, being far less covered than the methods derived from the conventional IPF approach: *Combinatorial Optimisation*.

Its first step is to partition the geographical area for which the population is generated into *p* zones. Two forms of inputs are then required:

- A sample from the whole population, regardless of the zone partition (like IPF).
- Over each of the *p* zones, a contingency table for a subset of the agent attributes. However, a zone-by-zone fit can also be performed under IPF (or any other method that estimates weights matching exogenous controls).

The method proceeds in the following way:

1. For each zone, agents are randomly selected from the sample, so that the population size of the current zone (which can be derived from the corresponding contingency table) is matched. A statistic measure is computed to compare the known contingency table of certain attributes in that zone and the corresponding variables of the generated sub-population.
2. One of the generated agents is then switched randomly (with replacement) with another one from the sample, and the statistic measure is computed again. If the fit of the new sub-population is better than the original one, then the switch is maintained, otherwise, the original one is preserved. This process is repeated until the goodness of the fit arrives at a threshold value, or a number of iterations is reached.

The second step is not so different from the random draws with replacement mentioned in the introduction. As a result, the same already described flaws persist, and no model for the unknown full population is derived.

2.1.3 Markov Chain Monte Carlo

Markov Chain Monte Carlo is based upon two steps: (1) fit of *conditional distributions* on the available data sample and (2) *Gibbs sampling*. The latter constitutes the actual Markov chain of this approach, and exploits the fitted conditionals to sample new agents.

This simulation-based technique is carefully discussed in the section about the methodology (Section 4), being the selected approach to be developed. However, here its advantages are mentioned, in particular with respect to the most common IPF.

First, a considerable benefit of MCMC is that it is also valid for continuous attributes.

However, the main advantages of this method is that it makes the new population not too dependent upon the reference sample, allowing heterogeneity. Indeed, not as many assumptions as IPF are imposed, like that the contingency table of the full population should be decomposable into an outer product, apart from considering the conditional probabilities as smooth functions (being usually fitted under a parametric model, as in this case).

MCMC does not also suffer from the problem of IPF null cells, even though it is shown in the obtained results that ignoring very unlikely agents is not always a flaw.

So far, MCMC seems to only have advantages compared to IPF. However, it can be unfeasible to sample the whole (and, in this frame, not directly known) joint distribution of attributes with a Gibbs-based Markov chain because of the very low transition weights it may encounter, which would not allow it to pass between some separate clusters of recurrent combinations of attributes (even if both these clusters have high probability under the joint distribution). This can be due to certain characteristics of the reference sample or, more generally, to the curse of dimensionality. IPF, being based upon directly sampling from an estimation of the joint probability distribution, does not have these problems.

2.2 Hierarchical Generation

While generating a population of agents, or after this process, it can be necessary to create hierarchies among the agents, i.e., in the usual cases of transportation modelling, to group them into households. This problem is called *hierarchical population generation* and it requires to consider two *levels* of data:

On the agent level Also said on the *individual level*, or simply agent variables, they are attributes clearly referring to an individual, like AGE.

On the household level More generally on the *group level*, these household variables, like INCOME OF HOUSEHOLD (sum of the INCOMES of the agents living in a same household), remain constant for all the agents living together.

In Section 2.1 it has been observed that the literature about plain population generation mostly focuses on IPF, a simple algorithm which has even been rediscovered many times. Regarding hierarchical generation, a similar lack of variety was met in the methods proposed by the sources consulted.

In Section 2.2.1 it is shown that some techniques based upon IPF have been developed to handle hierarchies, by fitting a contingency table on the household level with some characteristics of their population of agents.

However, there are not many research papers about creating associations in the frame of MCMC, since this method has only recently been applied for population generation. The only available work is a graph-theoretic solution which is discussed in Section 2.2.2 (from Anderson et al., forthcoming). Nevertheless, it did not prove very promising, as discussed in Appendix E.

In fact, all the works found in the literature seem to deal with households populated by very few people (apart from Barthelemy and Cornelis (2012), always no more than two), and every agent of the considered datasets is characterised by a manifest variable corresponding to its *type*, i.e. its role in the household (like OWNER or SPOUSE). This heavily limits the complexity of the problem.

Thus, it proved necessary to develop a new extension of MCMC which is discussed later (Section 4.3). The only methods which, a posteriori, seemed similar to it are treated in Section 2.2.3. However, they are still based upon IPF.

2.2.1 Group-Level Generation with Signatures

Given known marginals on both the agent and household level, IPF can be modified to fit a contingency table taking into account part of this information. Several solutions have been developed for this purpose, which are based upon *household signatures*. This term refers to aggregated data about the inhabitants of the households, e.g. the sum of their AGES, which is considered as an additional variable when fitting a contingency table on the household level. Hence, these households are also characterised by some information on their inhabitants, which would help to create a hierarchical population in a later step.

Some examples of this kind of algorithms are (1) *Iterative Proportional Updating* (Ye et al., 2009), (2) an approach directly based upon *entropy maximisation* (see Appendix B.1.1 and Bar-Gera et al. (2009)), and (3) *Hierarchical IPF*, which proved to be the best among these three (Müller and Axhausen, 2011a). All these algorithms are described in Müller and Axhausen (2012).

Using these methods to generate new agents and their hierarchies, only a cloning approach (taking agents proportionally to the weights of the contingency table) can be pursued, which however does not produce very heterogeneous results. Indeed, using s Monte Carlo synthesis, based upon random draws, all the agents belonging to a household are not bounded to respect some aggregated characteristics, which in this case are present in the fitted contingency table.

Instead, if there were control totals referring e.g. to the inhabitant with the highest AGE in its household, there would be the possibility to employ IPF or GR to consider the attributes of that agent as additional "household" variables. However, stability can become an issue the higher the number of considered attributes is.

These methods are generally more complicated than standard IPF, and they may also have to deal with possible inconsistencies of the available marginals on the agent and household level (see Pritchard and Miller, 2012, p. 11). Besides, if there are no marginals at all and those of the sample have to be used for IPF, the kind of approaches described in the next section is more appropriate.

2.2.2 Deterministic Graph between Populations

The idea of this approach is to consider distinct sub-samples of the reference dataset and to generate separate populations from them: the hierarchical generation is performed afterwards. This implies that this method is valid for any kind of population generation, even based upon MCMC; hence, it was possible to apply it to the MCMC populations generated in this thesis (Appendix E).

This kind of hierarchical generation has been proposed in Müller and Axhausen (2011b); however, the main reference article for the developed implementation was Anderson et al. (forthcoming). The demographic sample synthesised in this latter paper was constituted of two agent types: the OWNER of a household and its SPOUSE. This categorisation was provided as an attribute of the reference data.

Firstly, three populations were generated: households, OWNERS, and SPOUSES. Secondly, OWNERS were connected to households, and then SPOUSES to the households which were not yet complete, by building *bipartite graphs* through the *Hungarian algorithm*.

A bipartite graph connects two groups such that there are no edges between elements of the same group. In the reference paper it is also imposed that the bipartite graph to be found is a *matching*, i.e. no element of both groups is linked by more than one edge (as expected, given the nature of the agent types). Besides, this matching must be maximum, i.e. all elements of the smaller group must be connected by an edge; at the same time, some elements of the other, larger group will remain unconnected. In other words, if more OWNERS than households were generated, a bipartite graph between them should connect every household to one OWNER and be injective, i.e. no OWNER can be connected to more than one household (every OWNER can own at most one household).

This bipartite graph is found through the Hungarian algorithm, whose aim is to find the best matching between two groups such that an objective function is minimised. In the explored case, this function is the sum of all the weights corresponding to the edges of the matching; these weights are reported in an *edge matrix*, whose dimensions are the sizes of the two groups, and the entries are the weights of all possible edges between the two groups. The usual Hungarian algorithm connects groups of the same size, but it can be easily modified for different sizes by adding copies of rows or columns (those which are fewer) to make the matrix square. In the final result, their connections are ignored, leaving some elements of the larger group unmatched.

Thus, the main difficulty of this approach lies in how to define the edge matrix.

Besides, the assumption underlying Anderson et al. (forthcoming) is to avoid considering any direct link between OWNERS and SPOUSES. Indeed, these agents are connected to their households with two separate runs of the Hungarian algorithm: the edge matrix is computed with different coefficients, but, when connecting the SPOUSES, without considering the OWNERS who have already been linked. This can lead to strange clusters of agents if large households with more than two inhabitants are considered, as it was noticed during its implementation (Appendix E).

2.2.3 IPF-Based Conditional Monte Carlo

Unlikely the previous sections, here a hierarchical generation based upon conditionals—rather than a joint probability distribution—is considered (hence, more akin to MCMC). Three approaches from different articles are considered, which are analogue to a conditional Monte Carlo method.

Barthelemy and Cornelis (2012) deals with households populated by few agents, but which can be more than two (it is the only cited paper considering this possibility), with the agent types explicitly specified as an attribute of their reference dataset. After having fitted two separate IPF contingency tables, one on the agent level (with also the agent type as a variable) and the other for the households, two new populations of agents and households are sampled according to the weights of these tables. Then, some agents are gathered into a household by randomly drawing them, each time varying the type considered: OWNER, SPOUSE, CHILD, etc.

More specifically, given a certain household, its inhabitants are generated as follows:

1. An OWNER is first drawn, without replacement, from the population of agents.
2. Depending upon the number of people living in this household (which is one of its attributes), a SPOUSE, some CHILDREN, and additional ADULTS are also drawn from the pool of

agents. Some individual attributes can directly be derived from the chosen household (e.g., the OWNER and SPOUSE's SEX) or randomly drawn accordingly to some known distributions (e.g., the SPOUSE's AGE level can be extracted from a distribution conditioned by the OWNER's AGE).

This method was presented with no specific example in the reference paper (which is a review of various methods). Nevertheless, from its description it does not seem a *"black-box" method*, i.e. which can blindly be used, without modifying it for different input demographic samples; on the contrary, this method should be adapted according to the available variables to support Item 2 of its procedure.

Another approach which makes a more extensive use of conditionals is discussed in Müller and Axhausen (2011b) about a program called `PopSynWin`. It uses a sophisticated formula for computing conditionals, which favours the formation of households with agents characterised by categories still under-represented in the population.

Finally, a method fully based upon conditionals is described in Pritchard and Miller (2012). It creates households from agents generated by a Monte Carlo sampling from the IPF contingency table. This table, which is the joint distribution estimated under IPF, is converted to conditionals linking the agent types HUSBANDS and WIVES (only these two types of agents are considered, hence households populated by no more than two people). In other words, this method (1) fits an IPF contingency table with specific attributes for the different agent types, and then (2) derives the conditional weights by dividing the obtained entries of the contingency table by some appropriate marginals (depending upon the conditional distribution looked for).

E.g., suppose one would like to determine the probability of having a WIFE of AGE 24, given a HUSBAND of AGE 29 (AGE 24 and AGE 29 are two levels of the finite set of discrete categories the attribute AGE can assume). Then, one will take the ratio between the entry of the contingency table at coordinates for $\text{AGE}_{\text{WIFE}} \equiv \text{AGE } 24$ and $\text{AGE}_{\text{HUSBAND}} \equiv \text{AGE } 29$ (AGE_{WIFE} and $\text{AGE}_{\text{HUSBAND}}$ are two separate attributes), and the sum of the weights of any combination with $\text{AGE}_{\text{HUSBAND}} \equiv \text{AGE } 29$ (i.e. its marginal) as denominator.

In the original research explored in this thesis, the basic idea of the successfully developed approach for hierarchical generation afterwards proved to be in a way similar to the method just described.

3 Strategy

Having explored the current state of the art of population generation, with or without hierarchies, the followed strategy to tackle the issue of hierarchical population generation is summarised here and extensively treated in the following sections.

Since there was the will to develop something new in the context of simulation-based methods, offering more heterogeneity in the generated populations, IPF was not an option; neither it was Combinatorial Optimisation, being practically a reweighting. Instead, MCMC, the only simulation-based alternative in transportation modelling, was chosen, given also its considerable advantages (Section 2.1.3).

However, it was shown that generating hierarchies is an open problem in the literature, which mainly deals with (1) predefined *agent types*, which state the role of an agent in its family, (2) households sparsely populated (up to 2 inhabitants), and (3) no analysis about the validity of the synthesized composition of households.

Besides, while some hierarchical methods have been developed in the context of IPF, no definitive solution was found about MCMC, given also its very recent application for population generation. One of the few methods from the literature which should have been able to assign hierarchies to MCMC populations was actually tried but proved unsuccessful, as explained in Appendix E.

Thus, a more complex problem than the usual cases discussed in the literature had to be faced, and it was necessary to develop two extensions of MCMC, called *iMCMC* and *hMCMC*, on the individual and hierarchical level. They are discussed in Section 4, together with a more detailed explanation of the MCMC algorithm.

4 Methodology

The technique implemented for generating new populations is a simulation-based approach belonging to the family of *Markov Chain Monte Carlo* methods. The keyword MCMC identifies a class of methods simulating draws, slightly dependent upon each other, from a joint probability distribution which is difficult to directly access.

However, in this thesis the term MCMC is used to refer to the single MCMC-based technique discussed here, which does not either require an explicit knowledge of the underlying joint distribution. This is very good for high-dimensional problems like population synthesis, where a large number of attributes must usually be handled.

Nevertheless, while this method is very common in other contexts, only recently it has been applied for population generation (Farooq et al., 2013). In this methodological section, this MCMC algorithm is first described in great detail in its plain version for population synthesis (Section 4.1), followed by the developed extensions, aiming at generalising it (Section 4.2) and dealing with hierarchies (Section 4.3).

Appendix B.2 finally provides with a theoretical framework about MCMC, in particular referring to its most usual purpose.

4.1 Markov Chain Monte Carlo

Working as a two-step procedure, MCMC first aims at exploiting the conditional probability distributions of each attribute with respect to all the others, called *full conditional distributions*. However, partial conditionals may also be used, as done in the reference paper Farooq et al. (2013).

The second step involves the actual Markov chain of this method through a process called *Gibbs sampling*, which exploits the fitted conditionals.

The next section describes how this fitting is performed for categorical variables, although other ways are still possible (and continuous variables may also be considered).

4.1.1 Multinomial Linear Logistic Regression

The required conditionals for MCMC are computed a priori from the available demographic sample through parametric models, like *Multinomial Linear Logistic Regressions*. Other models are possible (e.g., discrete choice models, with also variables referring to the "non-chosen" alternatives), but MLLR is similar to the one used in the reference paper about MCMC for population synthesis (Farooq et al., 2013). Furthermore, and more importantly, it is one of the most general models, it proved feasible with the available Singapore sample, and, above all, being parametric, it proved optimal for the chosen implementation.

In the frame of MLLR, given e.g. a categorical variable X_i which can take M_i different categories, the so-called *response variable*, the logarithm of its conditional probability distribution (conditioned by $X_1, ..., X_{i-1}, X_{i+1}, ..., X_N$, the *predictor variables*) is modelled by:

$$\log\left(\frac{\mathrm{P}\left(X_i = x_{ij} \mid X_1 = x_1, ..., X_{i-1} = x_{i-1}, X_{i+1} = x_{i+1}, ..., X_N = x_N\right)}{\mathrm{P}\left(X_i = x_{iM_i} \mid X_1 = x_1, ..., X_{i-1} = x_{i-1}, X_{i+1} = x_{i+1}, ..., X_N = x_N\right)}\right) = \beta_{ij0} + \sum_{\substack{n=1 \\ n \neq i}}^{N} \beta_{ijn} \cdot x_n$$

$$j = 1, 2, ..., M_i - 1 \tag{1}$$

A system of $M_i - 1$ equations for M_i different conditional probability distributions $\mathrm{P}\left(X_i = x_{ij} \mid x_1, ..., x_{i-1}, x_{i+1}, ..., x_N\right)$ is then obtained, each corresponding to one category. This is usually called a logistic model although the *logit* function should be defined as $\frac{\mathrm{P}(X_i = x_{ij})}{1 - \mathrm{P}(X_i = x_{ij})}$ (for the j-th category), while in the employed fractions $\frac{\mathrm{P}(X_i = x_{ij})}{\mathrm{P}(X_i = x_{iM_i})}$, called *odd ratios*, the denominator is the probability of a fixed category, usually called *pivot category*. The M-th equation to complete the system comes from the normalisation condition:

$$\sum_{j=1}^{M_i} \mathrm{P}\left(X_i = x_{ij} \mid X_1 = x_1, ..., X_{i-1} = x_{i-1}, X_{i+1} = x_{i+1}, ..., X_N = x_N\right) = 1 \tag{2}$$

In the right-hand side of Eq. (1), β_{ij0} is an intercept, and the model is actually linear with respect to the predictors $X_1, ..., X_{i-1}, X_{i+1}, ..., X_N$. While the response variable must be categorical, these predictors can even be continuous.

If they are still categorical, they should be expanded into *dummy variables*: given e.g. a predictor X_n which can assume M_n possible categories, in the right-hand side of Eq. (1) it should appear as $M_n - 1$ sub-predictors $X_{n1}, X_{n2}, ..., X_{n,M_n-1}$ which can assume values 0 or 1. At most only one of them can be equal to 1: if $X_n = x_{nm}$, only $X_{nm} = 1$, while the others are 0. There are only $M_n - 1$ sub-predictors because, if all of them are equal to 0, this clearly means that $X_n = x_{nM_n}$. However, other *codings* for categorical variables are possible, like *contrast coding*, but the resulting fitted models do not change.

The previous paragraph implies that the parameter N used in other formulas of this thesis (like Eq. (9)) is different from how N is used in Eqs. (1) and (2), where it is the total number of variables, also considering the dummy variables into which the categorical attributes were expanded. In other words, here $N = (M_1 - 1) \cdot (M_2 - 1) \cdot (M_{N^*} - 1)$, with N^* being the "real" number of attributes and subtracting 1s as discussed.

At any rate, the coefficients β are usually computed through optimisation techniques aimed at minimising the negative logarithmic likelihood of the available data sample. This likelihood is defined as:

$$\mathcal{L}\left(\beta_1,\beta_2,...,\beta_N\right) = \prod_{k=1}^{K} \mathrm{P}\left(x_1^{(k)} \mid x_2^{(k)},...,x_N^{(k)}\right)\cdot \mathrm{P}\left(x_2^{(k)} \mid x_1^{(k)},x_3^{(k)},...,x_N^{(k)}\right)\cdot...\cdot \mathrm{P}\left(x_N^{(k)} \mid x_1^{(k)},...,x_{N-1}^{(k)}\right) \tag{3}$$

K is the total number of agents in the reference sample, the values $x_i^{(k)}$ are the observed attribute per agent, while $\beta_1,\beta_2,...,\beta_N$ are matrices of coefficients of the different attributes ($M_i - 1$ vectors per attribute i, where M_i is the number of its possible categories).

Taking the logarithm, a monotonic function, of the likelihood allows to directly insert the β_{ijn}s without changing the stationary point of this functional. Indeed, the probabilities are clearly dependent upon these parameters through Eqs. (1) and (2), and they can be expressed as:

$$\mathrm{P}\left(X_i = x_{ij} \mid x_1,...,x_{i-1},x_{i+1},...,x_N\right) = \frac{\exp\left(\beta_{ij0} + \sum_{\substack{n=1\\n\neq i}}^{N} \beta_{ijn}\cdot x_n\right)}{1+\sum_{j^*=1}^{M-1}\exp\left(\beta_{ij^*0} + \sum_{\substack{n=1\\n\neq i}}^{N} \beta_{ij^*n}\cdot x_n\right)} \tag{4}$$
$$j = 1,...,M_i - 1$$

For the pivot category:

$$\mathrm{P}\left(X_i = x_{M_i} \mid x_1,...,x_{i-1},x_{i+1},...,x_N\right) = \frac{1}{1+\sum_{j^*=1}^{M-1}\exp\left(\beta_{ij^*0} + \sum_{\substack{n=1\\n\neq i}}^{N} \beta_{ij^*n}\cdot x_n\right)} \tag{5}$$

From Eqs. (4) and (5), it is clearly assumed that the underlying conditional probability distributions of the complete population are smooth; in particular, they belong to C^{∞}. Hence, it will be impossible for these functions to become exactly zero even for very unlikely combinations of attributes and, since their values are the transition weights of the MCMC Markov chain, this implies that some very low, but non-zero transition weights lead to these combinations during the sampling. In other words, MCMC is able to overcome the “flaw” of IPF null cells.

However, it still holds true that, if a certain category of an attribute is completely missing from the reference dataset, the fitted conditional probabilities cannot take it into account. This is because conditional regression models are fitted directly on the dataset, without assuming that some data is missing.

4.1.2 Gibbs Sampling

With the fitted conditional distributions, the second (and final) step of MCMC is to actually implement a Markov chain for the generation of agents. This is done through *Gibbs sampling*: each attribute of the new agents is drawn according to its conditional distribution, where the predictors (all the other variables) are the already generated attributes of this current agent and, where some of them have not yet been sampled, the corresponding attributes of the previous agent. In other words, given an *initial seed agent* with a vector of attributes $\boldsymbol{X}^{(\mathbf{0})} = \left(x_1^{(0)}, x_2^{(0)}, ..., x_N^{(0)}\right)$, sample the category of attribute i of agent k according to the conditional distribution $\mathrm{P}\left(X_i^{(k)} \mid x_1^{(k)}, ..., x_{i-1}^{(k)}, x_{i+1}^{(k-1)}, ..., x_N^{(k-1)}\right)$.

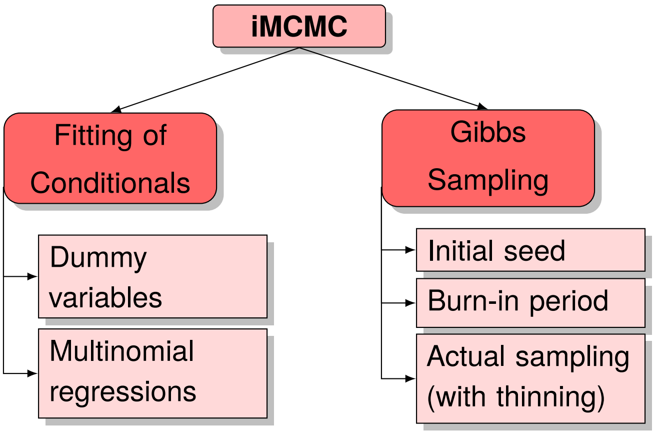
The Gibbs sampling procedure does not only make the process a Markov chain (indeed, $\mathrm{P}\left(\boldsymbol{X}^{(\boldsymbol{k})} \mid \boldsymbol{X}^{(\boldsymbol{k-1})}, \boldsymbol{X}^{(\boldsymbol{k-2})}, ...\right) = \mathrm{P}\left(\boldsymbol{X}^{(\boldsymbol{k})} \mid \boldsymbol{X}^{(\boldsymbol{k-1})}\right)$), but it also assures that the stationary probability distribution of the sampling is the joint distribution $\mathrm{P}\left(X_1, X_2, ..., X_N\right)$.

As usual with other Markov chains, in order to reset the memory of the initial seed agent and reach the stationary distribution, it is necessary to discard some sampled agents before collecting them for the new population, i.e. before the chain starts to sample at the desired equilibrium joint distribution, which can be not explicitly known with Gibbs sampling (even if it can be proven that there is a bijective relationship between joint distribution and full conditionals). This initial phase is usually called the *burn-in period*. Besides, between each two agents considered for the new population, some should be discarded in order to avoid too similar agents (or they can be retained to create other new populations), a process called *thinning*.

4.2 Individual Markov Chain Monte Carlo

As discussed in the previous section, MCMC is very interesting since it does not even try to directly deal with the underlying joint probability distribution of the attributes, a difficult task given their number.

Figure 1: Diagram of iMCMC



In the frame of the extension of MCMC explored here, which is called *Individual MCMC* (iMCMC), the aim was to be as general as possible: unlike (Farooq et al., 2013), full conditionals were always employed for all the attributes and their categories. Indeed, many more variables had to be handled and, above all, a possible “blind” usage of MCMC could be verified, i.e. if it is actually possible to apply it directly to any dataset, without having to analyse it first (working as a black-box method).

Nevertheless, in order to obtain acceptable populations, from the reference dataset the following have to be removed:

Certain categories Something equivalent was also done in Farooq et al. (2013); Anderson et al. (forthcoming). In those articles, depending upon the response variable, some categories of the predictors in the fitted conditionals were aggregated. In other words, a single coefficient β_i was considered for a dummy variable $X_i \vee X_j$, where $\vee$ is a logical disjunction.

Whole attributes Again, discarding attributes is very close to what was done in both cited papers, with partial conditionals that, given a response variable, do not consider all the other attributes as predictors.

4.3 Hierarchical Markov Chain Monte Carlo

This extension of MCMC, aimed at synthesising populations with a hierarchical structure, is based upon ordering the agents living in a same household according to their household roles (agent types). These can already be present in the reference sample as an individual variable (something usual in the data treated by the literature on transportation modelling, like Pritchard and Miller (2012)) or they must be defined.

The general formulation of hMCMC is based upon the definition of three groups of agent types, each to be generated differently:

OWNERS Synthesised under iMCMC and also characterized by variables on the household level.

Intermediate types The predictors of their conditionals take into consideration some variables of the already generated agent types, so that the model of their conditionals becomes larger the further the intermediate type is from the respective OWNER. The number of intermediate types to be defined can vary according to the complexity of the model one wants to implement.

OTHERS Their conditionals do not change and are only dependent upon their OWNERS and intermediate types, regardless of the step when they are generated.

The segmentation of the household into OWNERS, intermediate types and OTHERS is performed using a rule-based approach. The OWNER of a household can be identified e.g. by a sequential selection process. For example, first the person(s) with the highest reported INCOME are selected, and this subsample can then be further screened for other selection criteria until a single agent is identified as OWNER. Similar strategies are then also applied to classify the remaining persons of the household. Referring to a conventional nuclear family model, the intermediate types can be described as SPOUSE (second agent) and CHILD (third agent).

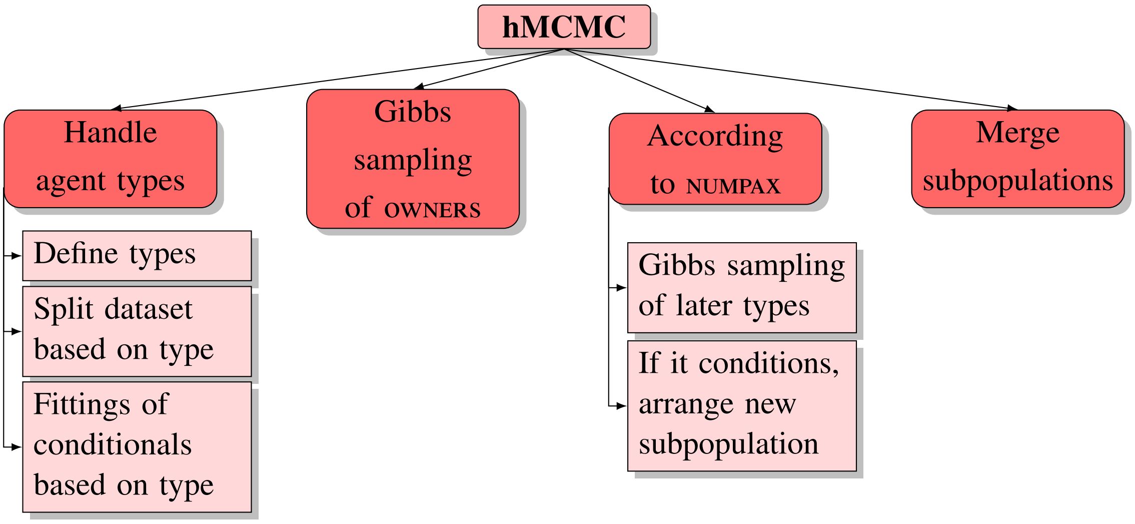
However, those descriptions should be interpreted with care as the proposed hMCMC approach does not impose a certain household composition model. Segmenting for example a single parent household will identify a CHILD as the first intermediate type, which is referred to here as SPOUSE. The accordingly fitted conditionals for the first intermediate types then ensure that the hMCMC process will generate the appropriate number of household types which not necessarily correspond to the family model employed to describe the different agent types.

Hence, to generate a household, its OWNER is sampled first according to the iMCMC approach. This agent wholly represents its household since the variables characterising it are drawn together, like the number of agents living there (NUMPAX).

Then, if the variable NUMPAX of the current OWNER is > 1, the other inhabitants of its household are also generated accordingly. In Gibbs sampling, the conditionals of these subsequent agents also depend upon some attributes of the already generated agents: e.g., when drawing the AGE of the CHILD of a household, the corresponding conditional is:

$$P\left(\text{AGE}_{\text{CHILD}} \mid \text{other variables}_{\text{CHILD}}, \text{some variables}_{\text{OWNER}}, \text{some variables}_{\text{SPOUSE}}\right) \tag{6}$$

Figure 2: Diagram of hMCMC


hMCMC
Handle agent types
Define types
Split dataset based on type
Fittings of conditionals based on type
Gibbs sampling of OWNERS
According to NUMPAX
Gibbs sampling of later types
If it conditions, arrange new subpopulation
Merge subpopulations

In case of a large NUMPAX, the attributes of type OTHER are not used to condition the generation of later agents of the same type. This was decided to keep the algorithm simple and avoid overfitting.

5 Data

This section is the first to refer to the available categorical dataset in the implemented case study. In particular, here its characteristics are discussed by both explaining its format (Section 5.1) and performing some deeper analyses about the relationships among its variables. The latter part will prove of fundamental importance when some attributes had to be ignored within the implemented MCMC approach to improve the results.

For this descriptive purpose, the methods employed are:

Multiple Correspondence Analysis Reviewed in Appendix C.1 (Le Roux and Rouanet, 2004; Greenacre and Blasius, 2006), it only takes into consideration linear relationships (Section 5.2).

Self-Organising Map Reviewed in Appendix C.3 (Kohonen, 1982), it also considers nonlinear relationships (Section 5.3).

Chow-Liu Tree As shown in Appendix D.1 (Kirshner et al., 2012), again it also considers nonlinear relationships (Section 5.4).

These techniques may also be employed to derive probabilistic models, rather than be limited to purely descriptive purposes like in this section. This possibility is discussed in Appendix D.

5.1 Demographic Sample

The demographic data sample employed in this case study was derived from the household records of the 2008 Household Interview Travel Survey of Singapore, commissioned by the country's Land Transport Authority. It consisted of 35,448 agents living in 10,640 different households. However, no marginals of the unknown full population were used because of the choice of MCMC, which does not impose them (but they would have been available from the Singapore Department of Statistics).

This sample was organised into a matrix, where every column corresponded to a variable and every row to an agent, an individual characterised by both agent and household variables; the household variables simply maintained the same categories for all the agents living together. Tables 1 and 2 list the attributes which were actually employed in this research, with their corresponding possible categories and relative marginal frequencies (with different normalisation factors if on the agent or household level).

The categories are reported in alphabetical order: this is why NO INCOME, lowest level of the ordinal variable INCOME, is not at the beginning of its column. The categories were listed in this way since their intrinsic ordering was not generally used (except when creating the agent types, see Section 7.1.1). Besides, when it was actually tried to impose the natural ordering in the codes fitting MLLRs on the data, although it was unnecessary, in order to produce more "logical" results, `R` returned fits inconsistent with the dummy coding usually employed. Section 6.1 covers this point in greater detail.

The meaning of all these variables is clear. One may argue that the attribute ETHNICITY has more sense as an agent variable, despite being marked as a household one. However, in the Singapore sample it presented an interesting property: all its households were ethnically homogeneous, meaning that all the agents living together always shared the same ethnicity. This is why it could be considered as a special household variable, and in the implemented hierarchical generations this attribute was sampled only once, for one agent per household, with the other inhabitants being assigned the same value.

The last three agent variables are further separated in another table because they were too much dependent upon INCOME; about this issue, refer to the next sections. However, the results of chi-squared tests between all variables taken two by two always discarded the hypothesis of independence, returning very low *p*-values, with an order of magnitude of 10^{-16}. This implies

Table 1: Main attributes of the Singapore demographic dataset,
with their corresponding categories and relative marginal frequencies

Agent level							
SEX	Freq.	AGE	Freq.	INCOME	Freq.	EDUCATION	Freq.
F	48.68 %	AGE 4	4.97 %	MAX 1000	4.99 %	INTERNATIONAL SCHOOL	0.30 %
M	46.36 %	AGE 9	7.96 %	MAX 1500	6.16 %	OTHER	0.08 %
N/A	4.97 %	AGE 14	7.83 %	MAX 2000	6.82 %	POLYTECHNIC	1.98 %
		AGE 19	7.85 %	MAX 2500	7.62 %	POST-SECONDARY	1.70 %
		AGE 24	5.48 %	MAX 3000	4.40 %	PRESCHOOL	3.20 %
		AGE 29	6.53 %	MAX 4000	6.53 %	PRIMARY SCHOOL	9.02 %
		AGE 34	7.45 %	MAX 5000	3.54 %	PRIVATE SCHOOL	0.54 %
		AGE 39	8.01 %	MAX 6000	2.05 %	SECONDARY SCHOOL	6.69 %
		AGE 44	9.02 %	MAX 7000	0.90 %	SPECIAL SCHOOL	0.11 %
		AGE 49	8.17 %	MAX 8000	0.61 %	UNIVERSITY	1.48 %
		AGE 54	7.69 %	NO INCOME	53.37 %	N/A	74.91 %
		AGE 59	5.61 %	OVER 8000	3.02 %		
		AGE 64	4.70 %				
		AGE 65+	8.73 %				

Household level					
ETHNICITY	Freq.	DWELL	Freq.	NUMPAX	Freq.
CHINESE	71.68 %	CONDO	13.28 %	1	10.55 %
INDIAN	12.66 %	HDB 1/2	4.45 %	2	20.23 %
MALAY	12.34 %	HDB 3	19.47 %	3	23.52 %
OTHER	3.32 %	HDB 4	29.31 %	4	26.02 %
		HDB 5	25.84 %	5	13.22 %
		LANDED PROPERTY	6.67 %	6	4.53 %
		OTHER	0.97 %	7	1.32 %
				8	0.44 %
				9	0.10 %
				10	0.03 %
				11	0.02 %

Table 2: Work attributes of the Singapore demographic dataset,
with their corresponding categories and relative marginal frequencies

Dependent upon INCOME					
OCCUPATION	Freq.	WORK BASE	Freq.	WORK HOURS	Freq.
CLEANER	2.70 %	HOME-BASED	1.42 %	FIXED	41.25 %
CLERK	4.29 %	NON-HOME-BASED	44.52 %	FLEX	4.69 %
MACHINIST	3.76 %	N/A	54.06 %	N/A	54.06 %
MANAGER	2.56 %				
OTHER	1.55 %				
CRAFTSMAN	2.09 %				
PROFESSIONAL	11.88 %				
SELLER	9.21 %				
TECHNICIAN	7.90 %				
N/A	54.06 %				

that the largest amount of possible combinations of attributes is practically impossible, since the number of characteristics which the agents can take is relatively low if compared to the total of possible combinations.

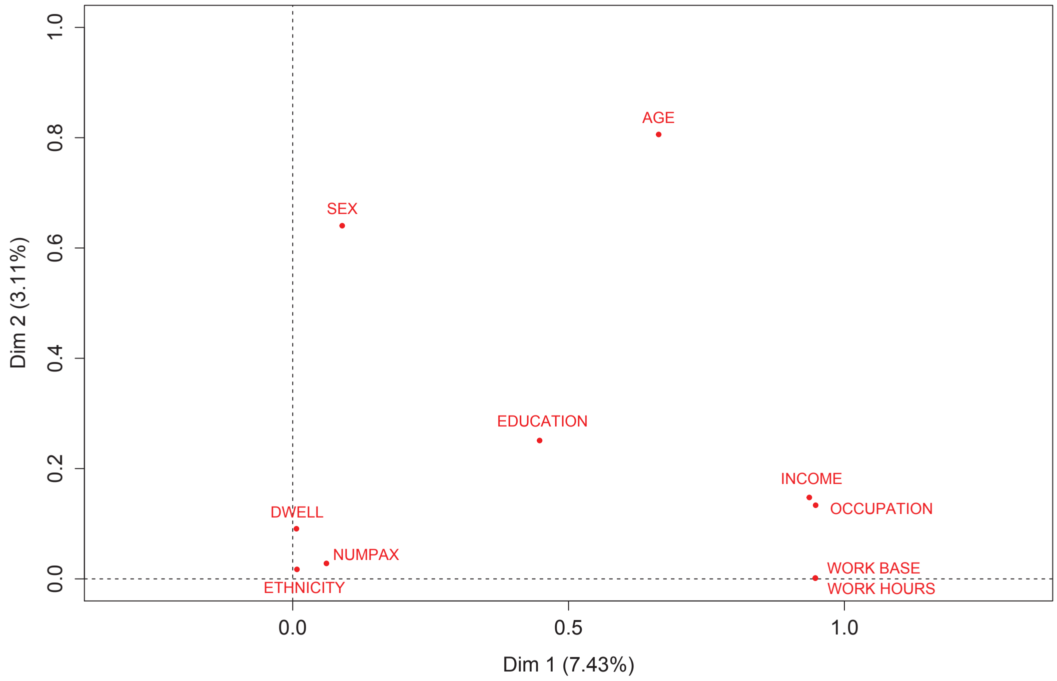
This fact can justify the choice to look for a joint distribution through a method which does not take into account an underlying model of independence, as IPF actually does, but which e.g. assigns nonzero probabilities only to certain "prototype" combinations of attributes. To this aim, refer to Appendix D.

Nevertheless, before moving on to these descriptive analysis, a final remark. In addition to all the attributes considered by Tables 1 and 2, there were also two other attributes on the household level in the sample:

INCOME OF HOUSEHOLD It was not considered in this thesis because it was the sum of the INCOMES of all the inhabitants of a household, and it would have been useless under MCMC, which cannot impose any marginals on the generated data. Hence, it could easily have been a different value from the actual sum of INCOMES of the agents per household as generated under the implemented MCMC-based hierarchical synthesis.

HHID A very important index for this research since it was specific to every household (it also contained the related postal code), thus allowing to identify agents living together (which shared the same household index).

Figure 3: Attributes of the Singapore dataset in the new MCA eigenagents' basis



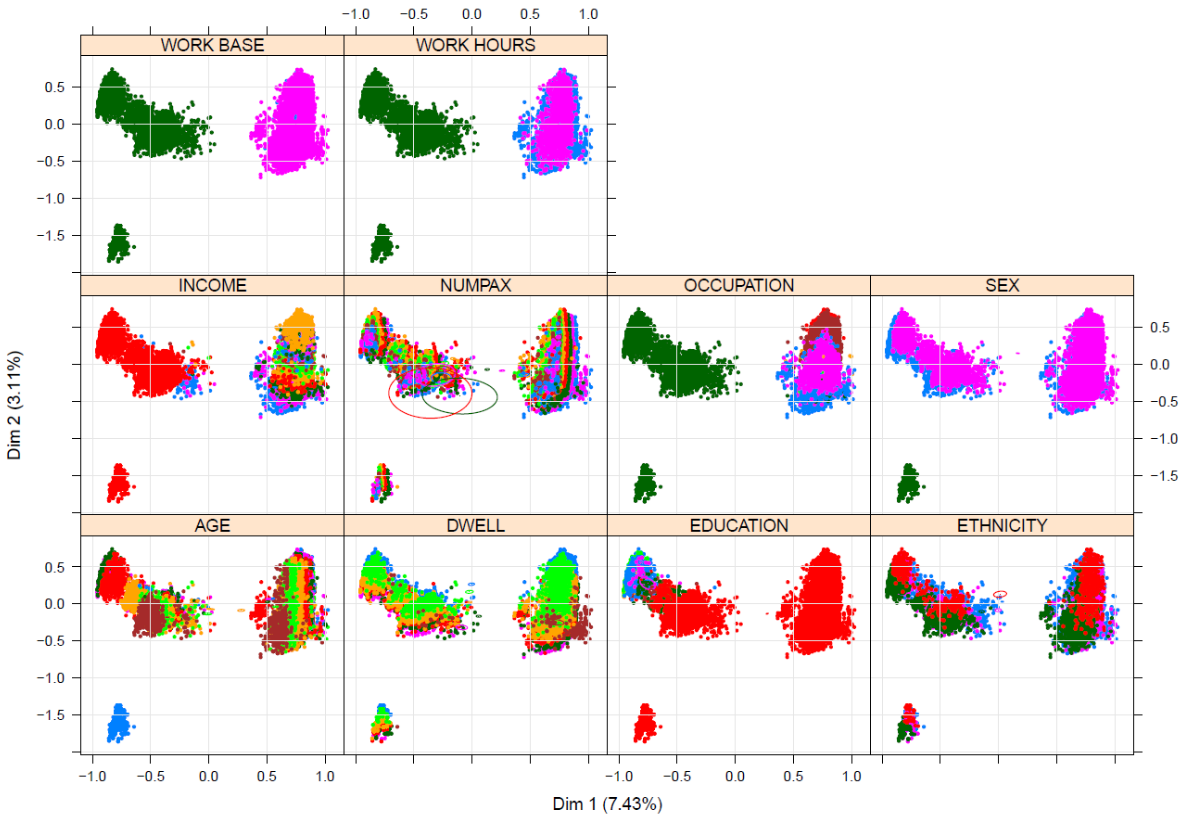
5.2 Linear Relationships among Attributes: Multiple Correspondence Analysis

With the `R` package `FactoMineR`, a *Multiple Correspondence Analysis* (Le Roux and Rouanet, 2004; Greenacre and Blasius, 2006) was performed on the purely categorical data of the Singapore sample.

MCA is the categorical version of *Principal Component Analysis* (Pearson, 1901), a linear algebra technique which, starting from a Singular Value Decomposition of the data matrix, looks for a new basis through which these data should be expressed instead of the original attributes. This basis is precisely formed by linear combinations of these original variables, which are given by the eigenvectors arising from the initial SVD (called *eigenagents' basis*). More details about the theoretical foundation of this method are provided in Appendix C.1.

Figure 3 reports the first two coordinates of each original attribute (the old basis in which the agents are reported) expressed through the new eigenagents' basis. These coordinates are the most important, since they correspond to eigenvectors having the largest eigenvalues in the frame of MCA.

Figure 4: Agents of the Singapore dataset in the new MCA eigenagents' basis, with categories (subplots refer to different attributes)



From this figure it is easy to visualise certain characteristics of the Singapore dataset. First of all, OCCUPATION, WORK BASE, and WORK HOURS, which are called *work variables*, seems to be linearly very close to each other and to INCOME.

A similar interdependency can be observed about the household variables, ETHNICITY, DWELL, and NUMPAX (about ETHNICITY being a household variable, see the previous section). However, this result is mainly due to the fact that all the agents of the dataset living together shared the same household variables, but MCA treated each agent separately, thus interpreting this fact as an evidence of a strong linear relationship among these variables. Besides, they were assigned weights with low magnitudes, not contributing much to the two largest eigenvectors.

Nevertheless, these results proved very useful to observe the attributes' "categorisation" reported in the first rows of Tables 1 and 2. They is also used in the implemented simulations to fix some issues: indeed, this kind of generations performs poorly when it deals with multipolarised data. Attributes like the work variables, which are strongly interdependent and thus almost "repeat" the same concept more than once (as with NO INCOME, which always implies N/A in all the work attributes), do not surely help breaking this entanglement.

Figure 4, which shows the agents of the dataset through their first two coordinates in the new eigenagents' basis clearly proves the existence of three main "clusters" of agents:

1. The *infants*, unique agents with the lowest AGE level AGE 4 and unspecified SEX N/A. All the work variables and EDUCATION are also unspecified, and these agents have NO INCOME.
2. The *unoccupied*, agents with N/A in all the *work variables*. Most of them have NO INCOME, and they are the only ones which can have EDUCATION categories different from N/A (the current students).
3. All the remaining *active workforce*, with unspecified education (but OCCUPATION can be different from N/A).

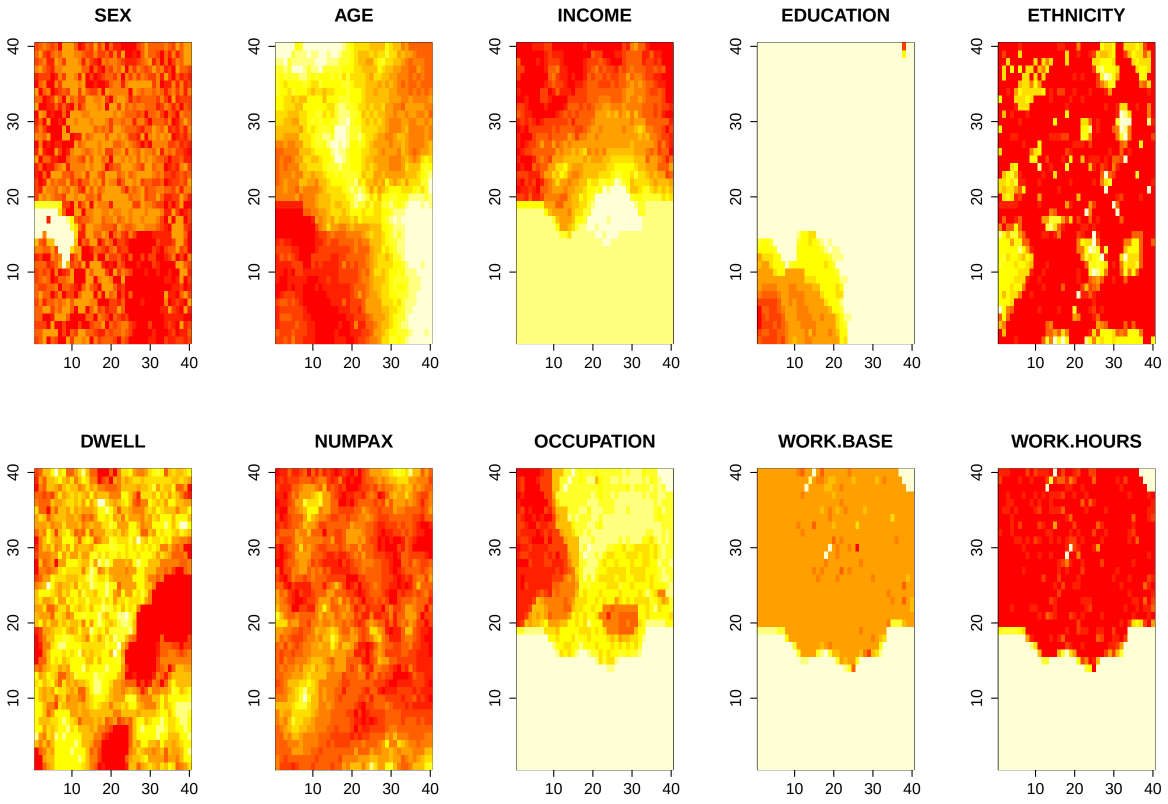
The characteristics of these clusters are deducible from the colours of Fig. 4, which refer to different categories (their labels have not been displayed for greater clarity). It should also report confidence-level ellipses around each category, although only some of NUMPAX are visible, meaning that the actual data points were much more concentrated than how it may seem from these plots. However, the household variables seem spread evenly across the different clusters, as expected since they are shared by agents with very different characteristics living in the same household.

It is shown in Section 6.2 that these observations considerably helped to produce better results in the MCMC population generation. Nevertheless, MCA was not used to define the agent types for the implemented hierarchical generations because of two motives:

- From a direct inspection, only the three clusters discussed above are clearly visible; however, more agent types would be needed.
- There are some data mining techniques, like *Hierarchical Clustering on Principal Components* from the R package FactoMineR itself (Husson et al., 2010), which would be able to detect clusters from the results of MCA (something which, in this case, is simply doable by eye). Nevertheless, there would be better clustering techniques, not based upon MCA: with the R package poLCA it would be possible to derive a latent variable from this dataset through a *Latent Class Analysis* (Appendix C.2). However, the agent types were still derived from other heuristic considerations.

The eigenagents of MCA may also have been used for population generation, similarly to what is done in computer graphics with *eigenfaces* (Sirovich and Kirby, 1987) or by assigning probability weights proportional to their eigenvalues, in order to approximate the underlying joint distribution. However, to this aim a *Self-Organising Map* would be preferable (Appendix D.2), whose descriptive results are discussed in the next section.

Figure 5: Self-Organising Map of the Singapore dataset



5.3 Nonlinear Relationships among Attributes: Self-Organising Map

Figure 5 shows the results of a *Self-Organising Map* (Kohonen, 1982) applied to the Singapore data sample, where each sub-map refers to an attribute.

Indeed, this data mining method helps to visualise high-dimensional data into a series of low dimensional (commonly 2D, like here) images, where similar colour patterns refer to clusters of data fixed in some categories for multiple attributes (while the specific shape or position of these patterns is not important). The SOM algorithm is carefully explained in Appendix C.3, given also the possibilities for future research in population synthesis which it offers (Appendices D.2 and D.3).

SOM also takes into consideration nonlinear relationships, but its results still confirm what was observed in the previous section:

- The agent variables do not seem to have clear interdependencies.
- The household variables have categories which are even more uniformly distributed

than the agent ones, as expected since agents living together share the same household attributes, regardless of the differences in the other variables.

- The work variables are strongly dependent upon INCOME.

The group of infants is also clearly recognisable in the SEX sub-map; in the AGE one it is also possible to distinguish their lowest AGE level as the darkest red spot.

However, differently from MCA (and the next Chow-Liu Tree), the SOM algorithm does not produce a univocal map: it changes according to the number of neurons and their positions, as well as their random initial attributes' vectors. In this case, a 40x40 square grid is considered, hence 1600 neurons.

Besides, to create this SOM the categorical variables were converted to integer numerical values which are then considered continuous: this is still observable in Fig. 5 since the obtained sub-maps seem "patchworks" of sharp colours (corresponding to different integer values, with a short transition made of decimal values). In particular, this is visible for the most interdependent attributes, the work ones, which are characterised by different colours only because they have different colour legends. This "patchwork" effect was also due to the low number of different categories that some attributes could take, and in fact it is less visible for AGE, NUMPAX, and, partially, INCOME, the only ordered variables of the dataset which also have many possible levels.

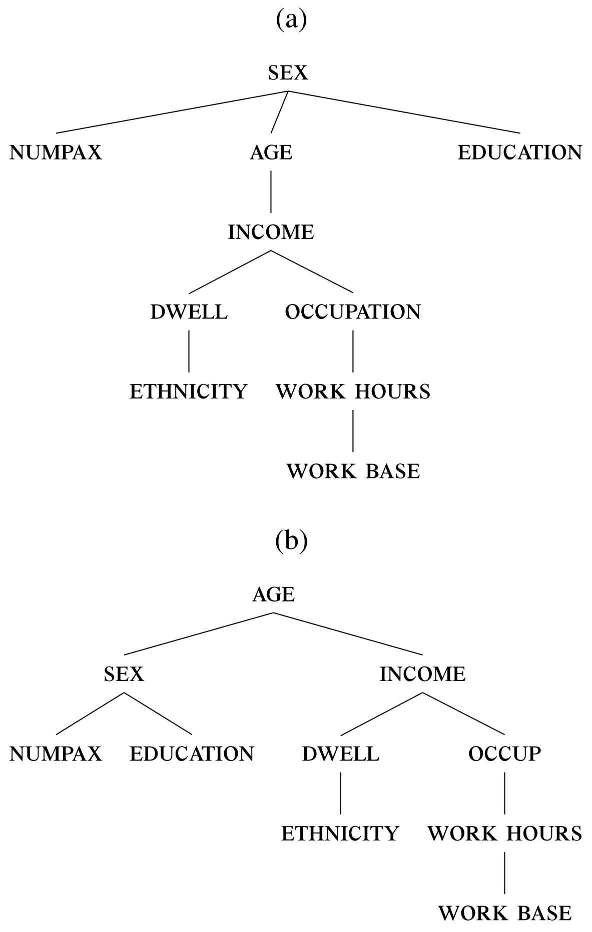
At any rate, what matters is to recognise similarity patterns for different attributes: the plot of the original agents fitted in the final SOM by finding their *Best Matching Units* (Appendix C.3) has not been reported, since it is not interesting to directly compare these agents (as it would have been, e.g., for a SOM made from data about economic parameters of different countries). However, this information may be used for the purpose of deriving a model for the joint distribution, by imposing the total of back-fitted agents per neuron as proportional to its probability weight, as it is discussed in Appendix D.2.

5.4 Nonlinear Relationships among Attributes: Chow-Liu Tree

A *Chow-Liu Tree* (Kirshner et al., 2012) was fitted on the demographic sample only to show the relationships among attributes, since every edge exists to maximise the sum of pairwise mutual informations between different couples of attributes (measuring nonlinear relationships). However, CLT was not used to approximate the joint probability distribution.

A further confirmation of the strong dependency of the work variables upon INCOME is obtained.

Figure 6: Chow-Liu Tree fitted on the Singapore dataset, shown with two different orderings



What is surprising is the position of NUMPAX, separate from the other household variables. However, as said in the previous sections, the results of the household attributes are less significant since they were, by definition, constant for all the agents living together.

Finally, the SEX attribute should not be considered as the most important only because it is shown at the top in Fig. 6(a); as discussed in Appendix D.1, being a tree a CLT is undirected, and hence it can be rewritten in any way, like in Fig. 6(b).

6 Experiment: Population Generation

Here the case study developed to prove the accuracy of iMCMC for plain population generation (Section 4.2) is illustrated. The discussion is structured into implementation (Section 6.1) and results (Section 6.2). In the latter the new populations are validated (1) by comparing the marginal distributions of some of their attributes with those of the reference sample and (2) with Standard Root Mean Square Error.

These were also methods employed in the reference article about MCMC-based population generation, Farooq et al. (2013). In particular, while SRMSE is the usual tool employed in the literature on transportation modelling to test the performance of population generation methods, in this case the marginals are equally important because MCMC does not impose marginal distributions on the fitted population, like IPF. However, if the generation is good the marginals should at least be similar to those of the reference sample.

The attributes chosen to explore the marginals were AGE and INCOME since they are the only ordinal variables of the Singapore data sample varying on the agent level (NUMPAX is ordinal but varies with the households), and they are useful to check the agent distributions per household in the frame of hierarchical generation. In fact, these validation methods are also used in the sections about these other results (Section 7.2).

Finally, Section 6.2.1 discusses an important concept which can prevent MCMC from returning acceptable results.

6.1 Individual Markov Chain Monte Carlo: Implementation

The recent application of MCMC for population generation prevented from finding already written codes exploiting it in this frame: the `R` packages `MCMC`, `MCMCpack`, or `MCMCglmm` are only aimed at Bayesian inference. Thus, the implementation of iMCMC was made from scratch, and was based upon:

1. Computing full conditionals in `R`.
2. Passing the results of the conditional fittings to `Java`, where the actual generation of agents through Gibbs sampling is performed (infeasible in `R` for performance reasons).

Item 1 only had to deal with categorical variables and, therefore, to fit models for a finite set of conditional probability weights. Three ways were explored for this:

1. After having fitted a model, it was tried to compute all the weights for every possible combination of agents in `R`, and then pass all these values to `Java`. However, due to memory shortage, `R` had problems with this operation when it had to deal with all the attributes (Fig. 7).
2. An `R` code was then opted which, having fitted the models, computed only the probability

weights needed for each response variable every time it was necessary to sample from a conditional distribution $\mathrm{P}\left(X_i^{(k)} \mid x_1^{(k)}, ..., x_{i-1}^{(k)}, x_{i+1}^{(k-1)}, ..., x_N^{(k-1)}\right)$, when the categories of the predictors are known. In other words, given a response variable X_i, this code returned the values of the MLLR probability formulas (Eqs. (4) and (5)) for all its categories.
Right after that, `R` directly called another `Java` code to do the actual drawing through the `R` package `rJava`; to this aim, `C/C++` with packages `inline` and `Rcpp` was also tried. Therefore, there was a "juggle" between codes written in different languages, something which proved computationally extremely intensive.

3. Thus, the final version of the iMCMC implementation fitted full conditionals in `R` with parametric models and passed only these parameters to `Java`, where the MLLR equations were used to obtain the desired weights for each draw. This further justified the choice of using MLLR for the fittings, being a purely parametric model (in addition to being the one used in the reference paper, Farooq et al. (2013)), and prevented from using other models to fit the conditionals, like *Multivariate Adaptive Regression Splines* or *Classification And Regression Trees*, due to their nonparametric nature and the related difficulty to pass their fits to `Java` in order to compute the probability weights there. The latter model mentioned in particular can be very successful to directly fit a joint probability distribution.
The final `R` code calls `Java` with command `system2` but, apart from a short string of arguments, like the number of agents to be generated or the numbers of categories per attribute, there is no direct interaction between them (`R` prints MLLR parameters to text files, which are then read by `Java`).

To do the actual fitting of conditionals in `R`, even the packages mentioned above aiming at Bayesian inference could have been used for the parameters of MLLR; however, an approach based upon maximising the likelihood was preferred. For this purpose, the following were tried:

`R` package `mlogit`: However, this solution proved unstable, being based upon discrete choice models and requiring a certain format for the input data sample. For further information about these issues, see Croissant (2012).

`R` package `glmnet`: Its main flaw is that, compared to the other packages, it is extremely slow, since it tries to fit lasso regularised models for multiple values of λ, so that the best can be chosen; for more information about this topic, consider Bühlmann and Mächler (2014, pp. 76–78). Besides, it was possible to make it work only for MLLRs without intercepts ($\beta_i j0$ in Eq. (1)).

`Biogeme`: This open source freeware for discrete choice models, developed by EPFL, is very flexible about the model to be fitted, but also computationally very intensive. It usually requires that the input sample is in a certain format, like `mlogit`, but it proved possible to apply it to this situation; however, it was still quite cumbersome to use.

R package nnet: Its function multinom was the best explored way to fit MLLRs, and hence the one employed in this entire thesis.

Nevertheless, a relevant difficulty was still encountered when trying to compute probability weights in Java by inserting the parameters from nnet into the MLLR formulas: when the code was initially tested, these weights were not equal to those directly computed in R, which were the right ones. This was caused by the R function ordered, used to reorganise the levels of INCOME of the dataset imported into R in a more logical way. Indeed, the default order followed by R is the alphabetical one, as shown in Table 1. However, nnet still handled INCOME in its alphabetical order and, when returning the related fitted parameters with INCOME as response, it automatically implemented contrast coding (Section 4.1.1).

Finally, the default pivot category considered by nnet is the first in alphabetical order, and not the last as hinted by the MLLR maximum likelihood given in Eq. (1).

Regarding the second step of the implementation, i.e. Gibbs sampling, Java was used to facilitate later integration into MATSim, a platform to implement agent-based transport simulations which is written in this language. Further information on this software can be collected from its website, www.matsim.org.

The generation process started with a seed agent, having randomly drawn all its attributes, and the burn-in period lasted 20,000 Gibbs steps, each of them referring to the whole sampling of an agent (the draws of all its attributes). Besides, the thinning was set to discard 20 Gibbs steps between two agents considered for the new population. Both these parameters are the same employed in Farooq et al. (2013).

In order to sample according to one of the discrete conditional distributions fitted (which were composed by a finite set of probability weights, as already mentioned), the Java code drew a uniformly distributed random number in the interval $(0,1)$. Then, given an array of these weights, it considered their cumulative sums (the first weight, this one summed to the second, this partial total summed to the third, etc.; the final total is obviously equal to 1) and compared them with the drawn value: the chosen category was the one corresponding to the smaller cumulative sum still greater than the draw.

6.2 Individual Markov Chain Monte Carlo: Results

All the plots of this subsection were obtained averaging 100 populations generated with iMCMC.

Figure 7: Marginals of dataset (left) and averaged iMCMC populations (right), considering all attributes and categories of Table 1

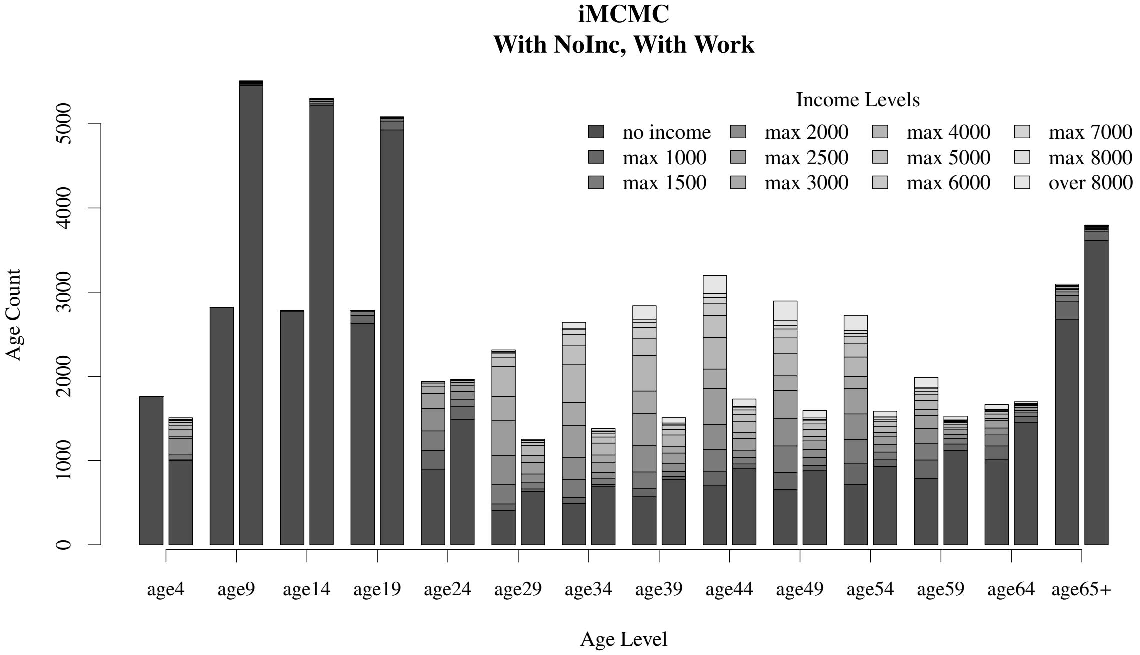
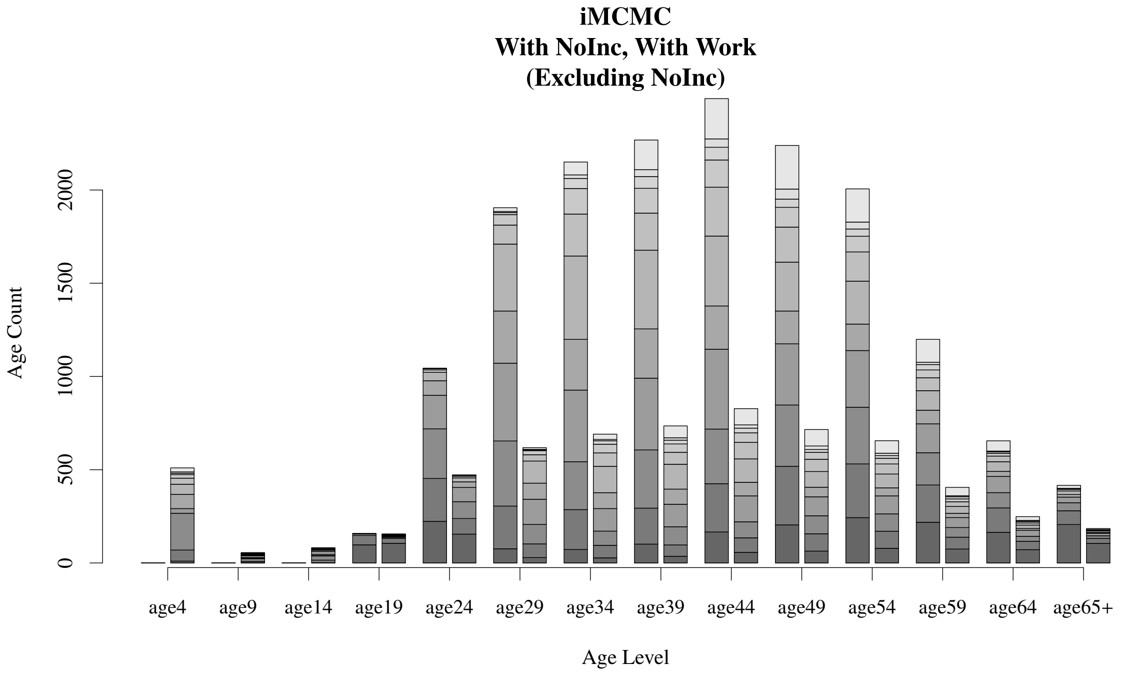


Figure 7 compares the marginal distribution of AGE—and, for each of its levels, the corresponding marginals of INCOME—belonging to the reference sample and to a new population generated by iMCMC. These distributions would have been equal with IPF (which is one of its main assumptions, although reached only when the IPF loop converges), but under MCMC they should at least be similar. However, in these plots, where the new population is generated considering all the attributes of Table 1, they are not even close.

While the AGE histogram of the reference sample is more or less uniform, it is not the case of the new populations, which are heavily skewed in AGE 9, AGE 14, AGE 19, and AGE 65+. This is due to a multipolarisation of the reference sample: almost all the agents with these characteristics have NO INCOME, and hence N/A in all their work attributes. Hence, these combinations of variables had high conditional weights between each other and very low towards other combinations. As it happens with a Markov chain for the simulation of, e.g., the *Ising model*, at certain conditions the sampling becomes "frozen" and it start considering other combinations only after a very large number of steps. For more details about this issue, see the section below.

A solution to this problem may be to aggregate some levels of INCOME (and, hence, make some agents of the reference sample undistinguishable), similarly as presented in Farooq et al. (2013). A second option, analogue to what is usually done with the null cells of a contingency table to

Figure 8: Same marginals of Fig. 7, excluding NO INCOME from both populations *a posteriori* (but it is still considered when fitting conditionals)



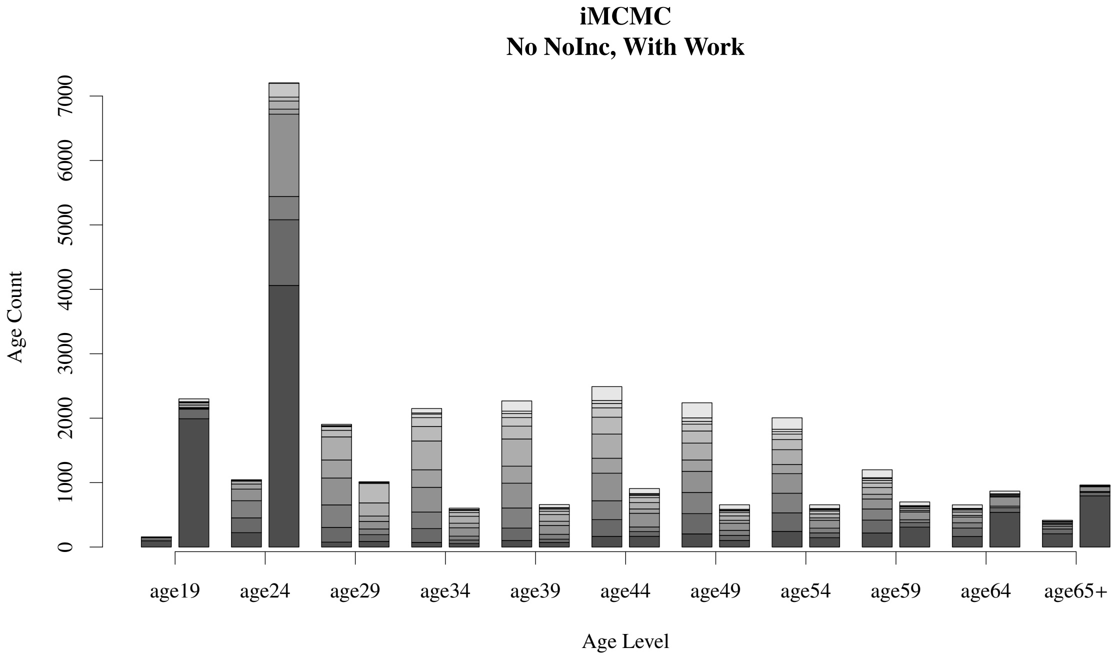
be fitted under IPF, into which small nonzero random values are inserted, might have been to introduce some fake data to make certain conditional weights larger (and, hence, the transitions they represent likelier). This can be done with MCMC for Bayesian inference, interpreting these data as missing in the reference sample (Appendix B.2). However, it may also make some unrealistic agents very possible (e.g., many infants provided with an actual INCOME), and thus was avoided.

Hence, Fig. 8 was created, which shows the same data of Fig. 7 but excluding those with NO INCOME. This passage was suggested by the results of MCA, where clusters characterised by agents having an INCOME or not were clearly distinguishable.

In this plot the new populations actually present characteristics quite close to the reference sample, at least if the columns of the generated populations corresponding to the lowest AGE level are ignored, which were completely removed from the reference data (clearly, all infants had NO INCOME).

Finally, from Figs. 9 to 11 it is possible to compare new populations generated from conditionals fitted on reduced versions of the reference sample, as described in their captions. The choice of excluding some data is similar to what was indirectly done in Farooq et al. (2013) with partial conditionals, i.e. not conditioned by all the other attributes. Indeed, there some categories

Figure 9: Marginals of dataset and iMCMC populations (averaged), excluding NO INCOME from reference sample *a priori* (not considered when fitting conditionals)



(or even whole attributes) were ignored as predictors of conditionals, which is very similar to actually discard them from the sample, as it was done here. Actually, in that paper different categories or attributes were not considered even according to the response variable. Hence, the MCMC algorithm employed there was not a black box from the beginning, while the implementation discussed in this thesis still always employs full conditionals.

In particular, Fig. 9 shows new populations generated from conditionals fitted without all NO INCOME data; however, the resulting marginals are still very different from those of the reference sample.

Hence, data characterised by NO INCOME were considered again and the work variables were instead ignored, being strongly dependent upon each other and upon INCOME. Indeed, in the same way of NO INCOME, also with work variables there is the risk to remain stuck sampling only certain combinations of variables. The plot of Fig. 10 actually shows a better match with the marginals of the reference sample, if compared to the previously explored generations.

However, the next logical step was to exclude both NO INCOME data and the work variables from the dataset on which to fit the conditionals, and the obtained results are in fact the closest to the reference sample, as it can be noticed from Fig. 11.

Figure 10: Marginals of dataset and iMCMC populations (averaged), excluding work variables from reference sample *a priori* (not considered when fitting conditionals)

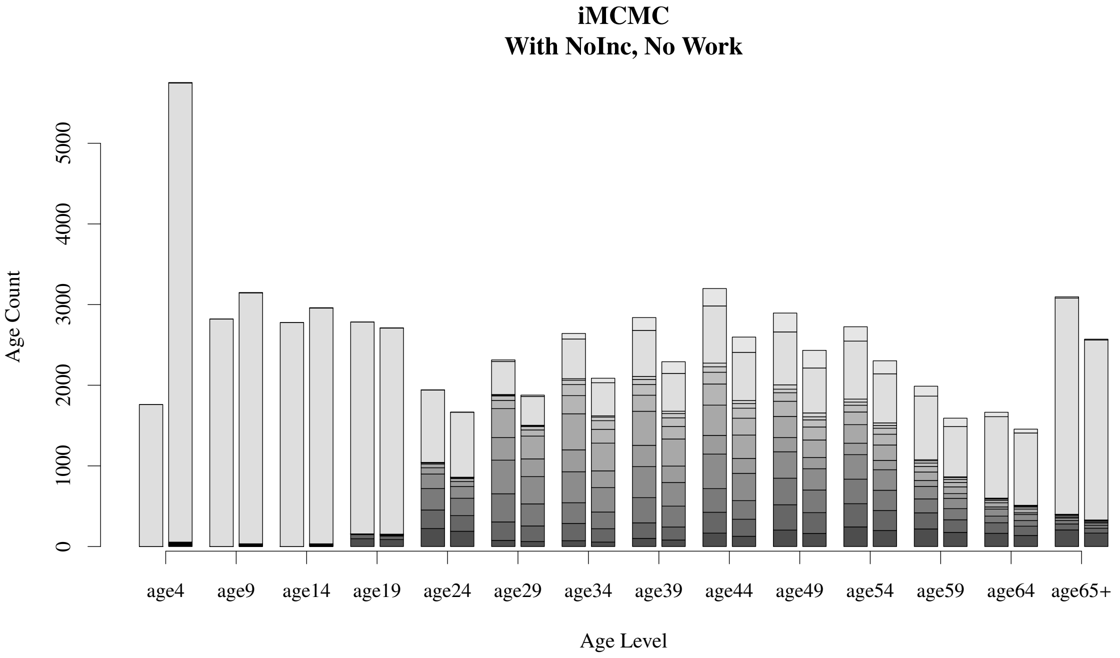


Figure 11: Marginals of dataset and iMCMC populations (averaged), excluding NO INCOME and work from reference sample *a priori* (not considered when fitting conditionals)

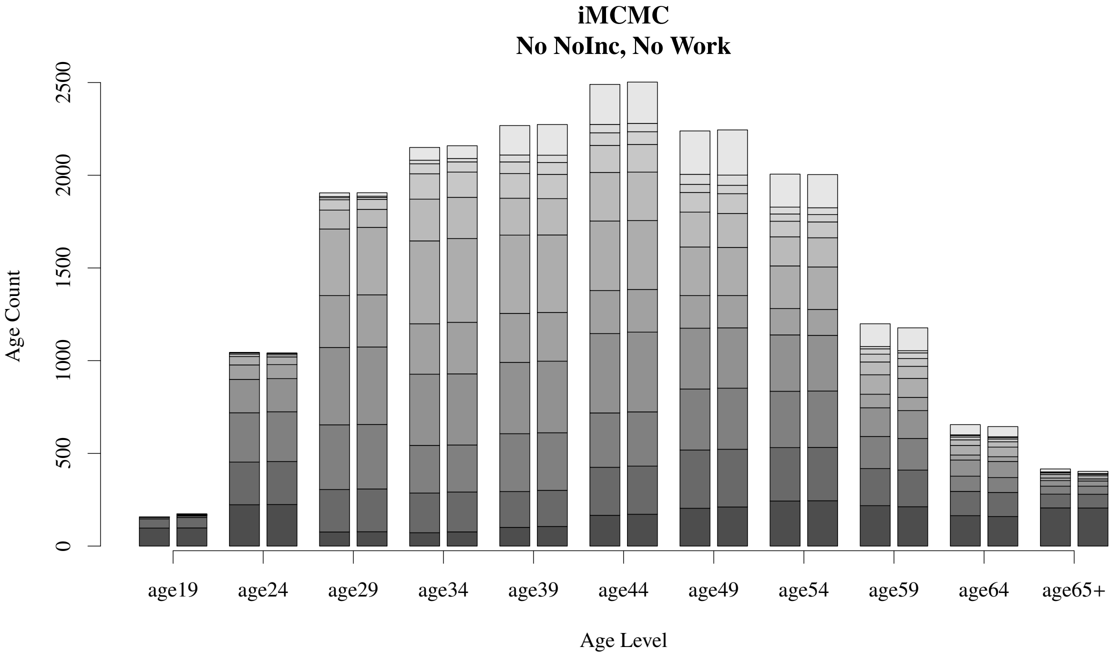


Table 3: SRMSEs of populations generated from different versions of the Singapore sample, with their standard deviations (having sampled 100 populations for each SRMSE)

Singapore dataset to fit conditionals	SRMSE	Standard deviation
Complete	178.16	3.63
Without NO INCOME	311.93	8.13
Without work variables	26.15	1.24
Without work and NO INCOME	19.74	3.68

Table 4: Totals of possible and actually observed combinations of variables, given different versions of the Singapore sample

Singapore dataset considered	Theoretical possible agents	Observed different agents
Complete	153,679,680	14,686
Without NO INCOME	53,665,920	11,342
Without work variables	1,707,552	9,736
Without work and NO INCOME	596,288	6,398

These same observations about population generations performed on different versions of the reference dataset are confirmed by an SRMSE analysis.

Like with the marginals, each SRMSE was computed averaging values obtained from 100 new populations of the same size of the sample they were derived, i.e. 35,448 or 16,529 agents, the latter if NO INCOME data were removed.

The still relatively high values of these SRMSEs if compared to the results of the reference paper of this thesis are justified by the very large numbers of considered possible agents, i.e. of different combinations of attributes, as shown in Table 4.

Indeed, SRMSE is linearly dependent upon the square root of these numbers (its formula is given in Appendix A, Eq. (9)). This is justified since the goodness of a population generation method is usually highly affected by this parameter; however, the flaws of population generations taking into account more complete versions of the Singapore dataset are not simply due to the number of combinations considered, but also to the “sparsity” of the distribution of the data, deducible from the relative small number of actually observed combinations of variables, which means that the data points occupy only a small portion of the so-called *attributes’ space*.

Nevertheless, this topic is explored in a broader context in the following section, which also tries to justify the larger SRMSE obtained when excluding NO INCOME data to fit conditionals. Besides, notice the very small standard deviations of the SRMSEs of the new populations, regardless of the Singapore dataset considered.

In conclusion, the most important result of this section is that iMCMC is a good method for population generation, preserving the characteristics of the reference dataset and, at the same time, allowing heterogeneity, but it is not a black box, i.e. it cannot be applied blindly to any kind of demographic sample and obtain acceptable results without first analysing the reference data. In particular, it performs poorly with strongly interdependent variables.

However, as also mentioned above, the fact that iMCMC is not a black box was implicitly stated in Farooq et al. (2013) through the way their conditionals were computed, where some categories or whole attributes are ignored as predictors according to the considered response variable.

What produced the little affinity of the generated populations with the reference sample were some very low conditional probability weights, which are the "transition" weights of the Markov chain of the algorithm, which prevented it from spanning the whole attributes' space. These low weights were due to the so-called *curse of dimensionality*.

6.2.1 Curse of Dimensionality

This section is devoted to a short explanation of the concept of *curse of dimensionality*, which causes what prevents the MCMC algorithm (or, more specifically, Gibbs sampling) from efficiently working with the complete Singapore sample.

The "dimension" at issue is related to the number of considered attributes, which controls the total of possible combinations of agents. Indeed, this latter comes from the product of the totals of available categories per attribute, which is bounded from below by the product of their minimum totals, i.e. 2 (otherwise, these variables would not show any actual variability); hence, it must be $> 2^N$, where N is precisely the number of considered attributes. This is why, more generally, the curse of dimensionality is so bad for many algorithms, like IPF itself (which becomes unstable): too many possibilities to be taken into account, which makes their complexity unbearable.

To explain the problems this curse causes on MCMC, the physical terminology of *phase space* can be useful, which in this context refers to the *attributes' space*: an N-dimensional space where every attribute varies along a dimension, and each agent, being a combination of variables, can be represented as a point (notice that here each variable is a dimension, while in the name of the curse this term refers to their number). With high-dimensional samples, i.e. characterised by many variables, this discrete and finite space (due to the categorical nature of these data) has too many dimensions, which leads to very long paths between data points, usually separated by some empty space of very unlikely combinations of attributes.

With the Singapore sample or, more generally, with demographic data, these effects are even more pronounced as it was visible from the MCA results and Table 4: the agents are very clustered, and only relatively few combinations of attributes are actually observable (which reflects the fact that in real life people with certain characteristics, like infants with a job or some nonzero INCOME, are impossible). This also implied the observed strong interdependency between many attributes of the Singapore sample.

A further proof of this issue was given by an MCMC generation from the Singapore dataset considering only its attributes SEX and AGE. The match of the new contingency table with the reference one was very poor: the dataset where to fit the conditionals was perfectly split into the group of infants, where information on SEX is not available and only the lowest level of AGE is present (AGE 4), and all the other agents, where the SEX could be either F or M and the AGE is always greater than AGE 4.

Thus, what breaks MCMC is not the large number of considered attributes itself, but the multipolarisation which it causes on the reference dataset where the conditionals have to be fitted, and which can also be visible at a low dimension.

Indeed, the Markov chain of MCMC, in order to walk between these clusters of data, has to pass through this empty space between them characterised by very unlikely agents, which, therefore, must be sampled (in the example above, with only SEX and AGE, they can be agents with e.g. N/A and AGE 65+). Given the absence of data points in that region (but also with just a low density of points), the fitted conditional probability weights, transition weights of the Markov chain, will be very low, their product will be even lower, and with it the probability to cover the whole path between these clusters. This prevents MCMC from freely "sweeping" the whole attributes' space: while Gibbs sampling should theoretically draw agents from the unknown joint probability distribution of their attributes, some agents will never be reached in a computationally feasible amount of steps. The larger SRMSE obtained when fitting conditionals having excluded the NO INCOME data, with respect to the more complete datasets, is probably due to the fact that these data were positioned so that they facilitated the transition between different agents.

Going back to the example discussed above, if INCOME was also considered, the closeness of the generated data to the reference ones considerably improved, in the opposite direction of the curse of dimensionality (i.e. by increasing the dimension of the problem, the number of considered attributes). This was due to the creation of a “bridge” of agents, characterised by NO INCOME, which connected the two clusters of the data where the full conditionals were fitted, making the transition weights of the Markov chain larger.

In conclusion, the results explored above reduce the usability of the MCMC approach, being based upon conditionals and thus suffering from flaws in a way analogue to a curse of dimensionality (but with exceptions, as it was shown). This section also revives methods for population generation based upon directly modelling the joint distribution, like IPF itself does. This is exactly why in Appendix D.2 another way to generate new populations was proposed, which precisely aims at exploiting the usually clustered nature of demographic data as an advantage.

7 Experiment: Hierarchical Generation

Given the recent application of MCMC to the field of transportation modelling for plain population generation, not many methods have been developed to handle its problem of hierarchical generation. Thus, hMCMC was developed.

As discussed in its methodology (Section 4.3), this method is based upon the concept of agent types, but the total of types defined in its implementation does not correspond to the maximum number of people living together. This was due to avoid overfitting; after all, only two households of the Singapore sample (out of 10,640) are actually populated by 11 agents. More details about this issue are provided in the section about its implementation (Section 7.1), in particular in Section 7.1.1, while the results in Section 7.2 prove the validity of the agent types chosen.

7.1 Hierarchical Markov Chain Monte Carlo: Implementation

Analogously to iMCMC (Section 6.1), the implementation of hMCMC used in this thesis fitted the conditionals in `R` under MLLR and passed their parameters to `Java`, where the probability weights were computed and used at each step of Gibbs sampling. Indeed, `R` alone performed poorly on a non-parallelisable process like Gibbs sampling; this justified the choice of fitting the conditionals with a parametric model, so that it was possible to only pass its parameters. MLLR as parametric model was chosen since the `R` package `nnet` through its function `multinom` proved to be very efficient.

This implementation also managed to solve a possible problem of hMCMC: the number of categories of a variable characterising an agent type can be different when this variable is involved in the generation of its agent type and when it conditions the generations of subsequent types. The developed implementation succeeds in dealing with this issue by discarding the combinations of variables which cannot be interpreted by models of later agent types (i.e. through an acceptance-rejection sampling). Like IPF, therefore, this implementation of hMCMC has a sort of zero-cell problem.

To better illustrate this prevented flaw, consider the following example. In the reference sample the youngest OWNER was the only one characterised by AGE 14, and it lived alone. However, when the attribute $\text{AGE}_{\text{OWNER}}$ was used to condition the generation of SPOUSES, this level was absent in the subsample made of SPOUSES and their OWNERS, on which the SPOUSES' conditionals had to be fitted. Any regression method, and MLLR is no exception, cannot fit completely missing categories in the input data, but there would be no problem with missing combinations (thus, solving the real IPF zero-cell problem). It follows that some OWNERS with AGE 14 and NUMPAX > 1 could actually be generated, but the conditionals to predict their SPOUSES could not have handled them. Thus, they were discarded.

Another relevant detail is that, since MCMC can be performed as long as necessary and stopped at any time, the number of considered steps of an iMCMC Markov chain is exactly equal to the size of the generated population, while this approximatively holds for hMCMC. The uncertainty is due to the fact that the controllable size of the synthetic population in hMCMC is actually the number of OWNERS, and the total of agents is then controlled by the OWNERS' NUMPAX and the total of OWNERS which had to be removed due to the flaw of hMCMC just discussed in the previous paragraph.

A number of OWNERS 10 times larger was chosen because the results of hMCMC seemed to improve the larger the synthetic population was: its Markov chains had more time to reach the more separate combinations in the attributes' space, and to properly explore the underlying joint distribution. Thus, it is possible to correctly obtain a small population from the hMCMC result only by subsampling a larger one (e.g., by increasing the thinning or through decimation). However, the fact that the size of the analysed synthetic population from hMCMC, 354,178 agents, was actually very close to 10 times the reference one (35,448) is an evidence of the validity of the developed approach.

In fact, the section about hMCMC results takes into account only one synthetic population, without averaging many of them as done in the iMCMC results (Section 6.2). This was decided to properly explore its household composition, something which is not usually performed in the literature of transportation modelling when dealing with hierarchical methods.

However, before discussing these results, the specific implemented segmentation of original households into agent types are described.

7.1.1 Agent Types

The proposed hMCMC requires agent types, and these had to be defined on the demographic sample employed. As discussed in the methodology, their role is to simply order the inhabitants of a household which are then synthesised. However, instead of names like TYPE 1, TYPE 2, etc., specific names were chosen for clarity, resembling the procedure through which these types were defined:

OWNER 1st agent of its household, with the highest INCOME. If there was a conflict, the one with the highest AGE was chosen. If they were still multiple, the OWNER was randomly identified.

Intermediate types

- **SPOUSE** 2nd agent of its household, with the minimum AGE distance with the OWNER, among those with opposite SEX. If no agent with opposite SEX was available, the one with the minimum AGE distance was chosen. If they were multiple, the SPOUSE was randomly identified.
- **CHILD** 3rd agent of its household, with the maximum AGE distance with the OWNER. If there was a conflict, the one with the maximum INCOME distance was chosen. If they were still multiple, the CHILD was randomly identified.

OTHER Remaining agents of the household.

In hMCMC these types were treated differently when fitting their conditionals. In order to be as general as possible—as it was also done for iMCMC—these were dependent upon all the other synthetic attributes and as many variables as possible from the already generated types. Hence, while OWNERS were generated under iMCMC also with the household attributes, the synthetic variables of the intermediate types were only SEX, AGE, and INCOME, conditioned by all the attributes of the already generated types living in the same household, as outlined in Table 5. However, incomplete models may also have been considered.

Out of the 35,448 individuals in the reference sample, 7,927 were identified as OTHERS. However, only in 2,092 households more than two individuals were identified as OTHERS, as these household were composed of five or more individuals. Due to this comparably small number and to avoid overfitting, the same conditionals were used for any agent of type OTHER, i.e. for NUMPAX $\geq$ 4.

Table 5: Attributes involved in conditionals fitted per agent

Role of attributes	OWNERS	SPOUSES	CHILDREN	OTHERS
Both predictor and response (synthesized)	SEX, AGE, INCOME, ETHNICITY, DWELL, NUMPAX	SEX, AGE, INCOME	SEX, AGE, INCOME	SEX, AGE, INCOME
Only predictor (conditioning)		ETHNICITY, DWELL, NUMPAX, SEX$_{\text{OWNER}}$, AGE$_{\text{OWNER}}$, INCOME$_{\text{OWNER}}$	ETHNICITY, DWELL, NUMPAX, SEX$_{\text{OWNER}}$, AGE$_{\text{OWNER}}$, INCOME$_{\text{OWNER}}$, SEX$_{\text{SPOUSE}}$, AGE$_{\text{SPOUSE}}$, INCOME$_{\text{SPOUSE}}$	ETHNICITY, DWELL, NUMPAX, SEX$_{\text{OWNER}}$, AGE$_{\text{OWNER}}$, INCOME$_{\text{OWNER}}$, SEX$_{\text{SPOUSE}}$, AGE$_{\text{SPOUSE}}$, INCOME$_{\text{SPOUSE}}$, SEX$_{\text{CHILD}}$, AGE$_{\text{CHILD}}$, INCOME$_{\text{CHILD}}$

7.2 Hierarchical Markov Chain Monte Carlo: Results

In this section one hMCMC synthetic population is compared to the reference sample. The very low standard deviations of SRMSEs from 100 iMCMC populations (Table 3) support this choice.

This synthetic population has 354,178 observations, around 10 times larger than the reference sample (in plots and tables abbreviated as RS). Besides, it was synthesised without EDUCATION: indeed, the populations produced by including this variable, while being considered acceptable by error measurement standards, showed many outliers, and have not been reported. The work variables were excluded as well, as already done in the results of iMCMC.

7.2.1 Individual Level

These results only refer to characteristics on individual level. However, the related new population was still synthesised with hMCMC, and the generated hierarchical structure was simply ignored.

Figure 12: Frequencies of combinations of agent variables

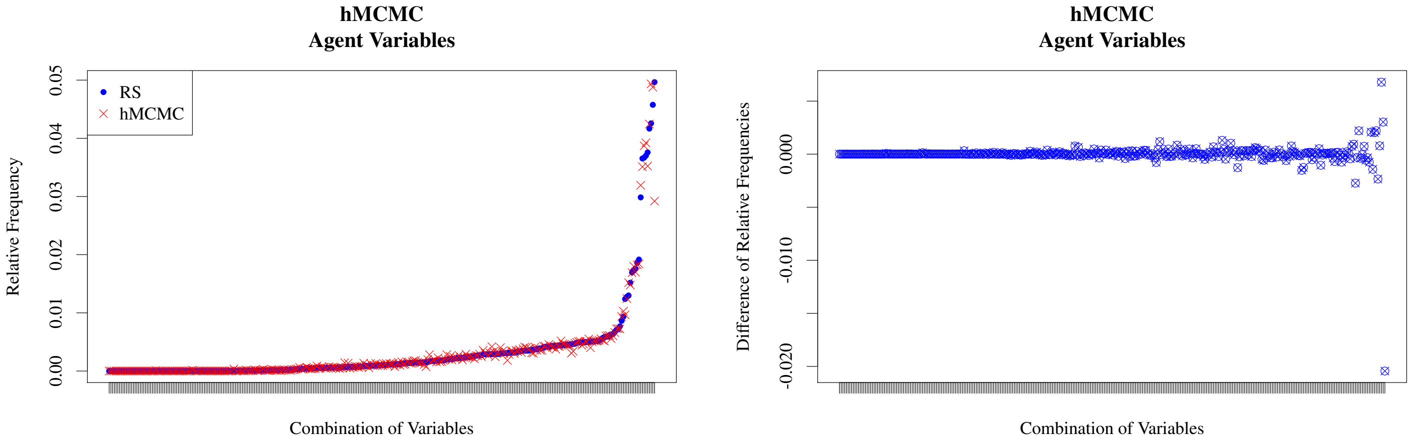
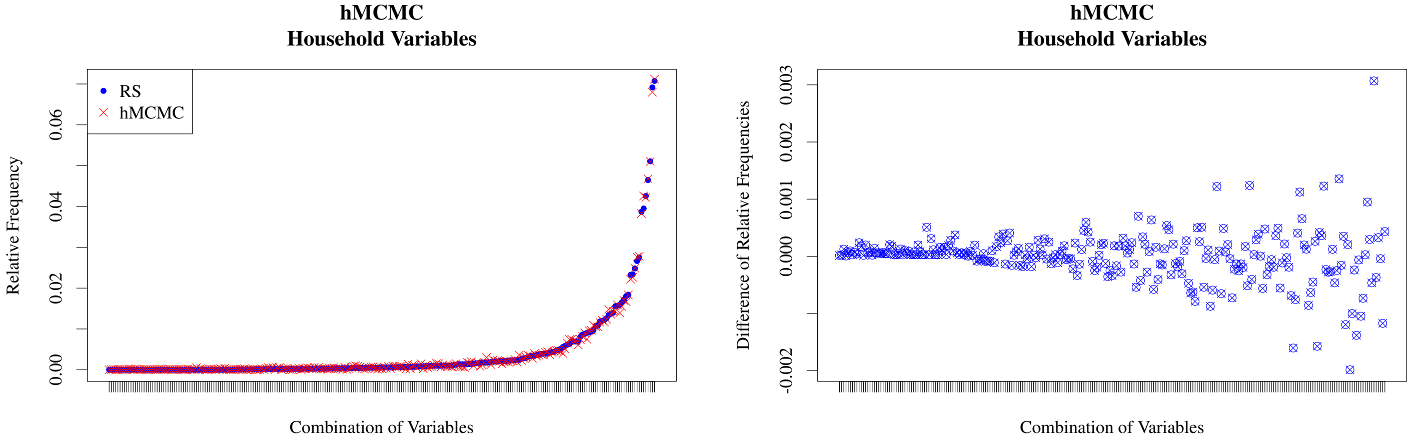


Figure 13: Frequencies of combinations of household variables



Figures 12 and 13 show plots of the relative frequencies (normalised counts) of combinations of variables, thus allowing to display results of multidimensional attributes in 2D pictures. Figure 12 shows all feasible combinations of categories for the individual variables AGE, SEX, and INCOME, ordered by their frequencies in the reference sample. Figure 13 shows the same analysis for the household variables ETHNICITY, DWELL, and NUMPAX; note that no links between the individuals in a household are analysed here. The left part of both figures shows the relative frequencies for both populations; the right part shows the absolute error of the hMCMC population compared to the reference sample.

As expected since more agents are involved, the error increases with increasing frequency in the

Table 6: Most recurrent individuals and households in the reference sample

Attributes			Frequencies	
SEX	AGE	INCOME	RS	hMCMC
N/A	AGE 4	NO INCOME	4.97 %	2.92 %
F	AGE 65+	NO INCOME	4.58 %	4.88 %
M	AGE 9	NO INCOME	4.26 %	4.94 %
M	AGE 14	NO INCOME	4.17 %	4.25 %
M	AGE 19	NO INCOME	3.76 %	3.52 %
Attributes			Frequencies	
ETHNICITY	DWELL	NUMPAX	RS	hMCMC
CHINESE	HDB 4	4	5.89 %	5.92 %
CHINESE	HDB 5	4	5.76 %	5.66 %
CHINESE	HDB 4	3	5.16 %	5.19 %
CHINESE	HDB 5	3	4.73 %	4.68 %
CHINESE	HDB 3	2	4.14 %	4.14 %

reference sample, but mostly remains below 1 ‰. Notice however the large outlier at 2 % which is due to the infants, because of the previously described flaw of Gibbs sampling (Section 6.2.1); in fact, these agents almost constituted a separate subsample of the reference sample, given their characteristics.

This can also be noticed in Table 6, which reports the five most recurrent individuals and households in the reference sample with their corresponding frequencies in the hMCMC synthetic population.

To summarise Figs. 12 and 13, SRMSEs are provided. On the individual level, the SRMSE of the hMCMC population was 0.411; on the household level, 0.117.

It is interesting to compare them with iMCMC. This algorithm, run without the EDUCATION variable to produce a population 10 times the reference one, produced a global SRMSE of 4.839 (there was no household level) and, considering only the personal variables, 1.413, a worse value than hMCMC. This is a consequence of the advantage of subdividing the data sample into agent types, which allowed the multiple hMCMC Markov chains of this algorithm to operate on more clustered data. In fact, when the reference data is grouped into a single cluster, the interdistances between likely data points are short, the fitted conditionals form high-probability "bridges" between them, and Gibbs sampling performs very well.

Table 7: Table of marginals

Individual

Attribute	Category	RS	hMCMC
SEX	F	48.68 %	49.49 %
	M	46.36 %	47.57 %
	N/A	4.97 %	2.94 %
AGE	AGE 4	4.97 %	2.95 %
	AGE 9	7.96 %	8.87 %
	AGE 14	7.83 %	8.12 %
	AGE 19	7.85 %	7.49 %
	AGE 24	5.48 %	5.38 %
	AGE 29	6.53 %	6.61 %
	AGE 34	7.45 %	7.63 %
	AGE 39	8.01 %	8.14 %
	AGE 44	9.02 %	8.98 %
	AGE 49	8.17 %	8.25 %
	AGE 54	7.69 %	7.73 %
	AGE 59	5.61 %	5.76 %
	AGE 64	4.70 %	4.75 %
	AGE 65+	8.73 %	9.33 %
INCOME	NO INCOME	53.37 %	52.79 %
	MAX 1000	4.99 %	4.96 %
	MAX 1500	6.16 %	6.29 %
	MAX 2000	6.82 %	6.96 %
	MAX 2500	7.62 %	7.64 %
	MAX 3000	4.40 %	4.50 %
	MAX 4000	6.53 %	6.55 %
	MAX 5000	3.54 %	3.62 %
	MAX 6000	2.05 %	2.10 %
	MAX 7000	0.90 %	0.94 %
	MAX 8000	0.61 %	0.61 %
	OVER 8000	3.02 %	3.02 %

Household

Attribute	Category	RS	hMCMC
ETHNICITY	CHINESE	71.68 %	71.70 %
	INDIAN	12.66 %	12.60 %
	MALAY	12.34 %	12.42 %
	OTHER	3.32 %	3.27 %
DWELL	CONDO	13.28 %	13.31 %
	HDB 1/2	4.45 %	4.58 %
	HDB 3	19.47 %	19.39 %
	HDB 4	29.31 %	29.35 %
	HDB 5	25.84 %	25.76 %
	LAND. PROP.	6.67 %	6.62 %
	OTHER	0.97 %	1.00 %
NUMPAX	1	10.55 %	10.74 %
	2	20.23 %	20.30 %
	3	23.52 %	23.31 %
	4	26.02 %	25.98 %
	5	13.22 %	13.42 %
	6	4.53 %	4.28 %
	7	1.32 %	1.26 %
	8	0.44 %	0.42 %
	9	0.10 %	0.14 %
	10	0.03 %	0.08 %
	11	0.02 %	0.07 %

Finally, marginals are reported in Table 7. MCMC does not guarantee to respect the marginals of the reference sample, although the hMCMC marginals should still be quite close to them.

7.2.2 Household Level

The plots and tables of this section actually consider the household composition, thus analysing the goodness of the generated hierarchical structure.

Figures 14 and 15 show boxplots of the distributions of per-household means and standard deviations of AGE and INCOME. AGE and INCOME are used because they are the only ordered variables of the Singapore sample varying at the individual level, and hence whose mean and standard deviation are significant because their categories can naturally be converted to numeric values.

Table 8: Most recurrent couples of OWNERS and SPOUSES in the reference sample

Attributes of OWNER			Attributes of SPOUSE			Frequencies	
SEX	AGE	INCOME	SEX	AGE	INCOME	RS	hMCMC
M	AGE 65+	NO INCOME	F	AGE 65+	NO INCOME	2.44 %	1.68 %
M	AGE 65+	NO INCOME	F	AGE 64	NO INCOME	0.66 %	0.54 %
M	AGE 44	MAX 2500	F	AGE 44	NO INCOME	0.57 %	0.44 %
M	AGE 44	MAX 4000	F	AGE 44	NO INCOME	0.45 %	0.36 %
M	AGE 49	OVER 8000	F	AGE 49	NO INCOME	0.41 %	0.37 %
M	AGE 44	OVER 8000	M	AGE 44	NO INCOME	0.40 %	0.35 %
M	AGE 49	MAX 2000	F	AGE 49	NO INCOME	0.39 %	0.32 %
M	AGE 54	MAX 2500	F	AGE 49	NO INCOME	0.39 %	0.32 %
M	AGE 54	MAX 4000	F	AGE 54	NO INCOME	0.38 %	0.29 %
M	AGE 49	MAX 4000	F	AGE 49	NO INCOME	0.37 %	0.32 %

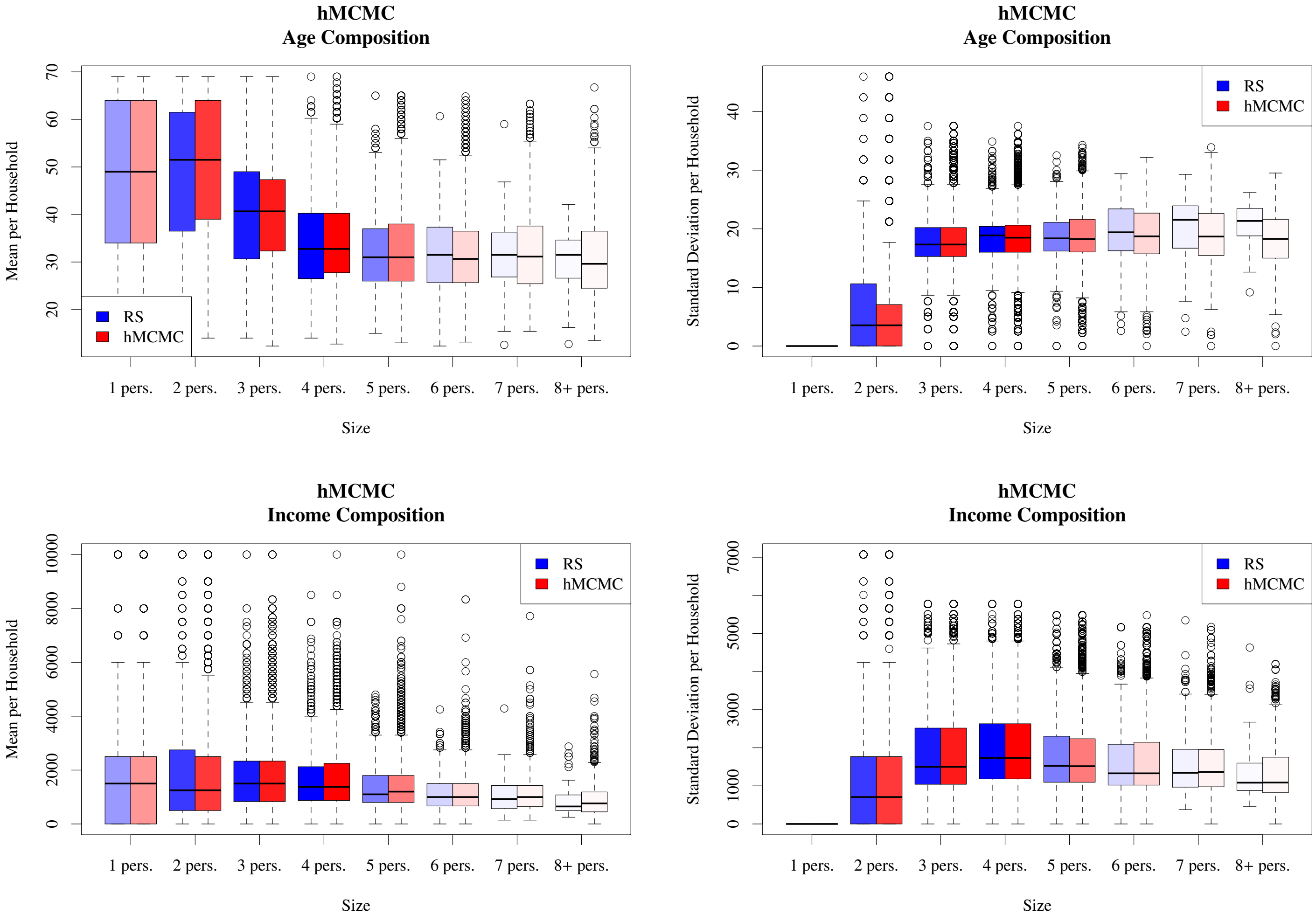
Because of this, they also played a special role when the types were defined. Their distributions are analysed for households with different SIZE (Fig. 14) and DWELL (Fig. 15). SIZE is equivalent to NUMPAX but with fewer levels.

It can be noticed that outliers in the AGE and INCOME distributions tend to be more present in the synthetic populations. This is expected because of their larger sizes and the heterogeneity of MCMC, which permits the generation of combinations absent in the reference sample.

Besides, Fig. 14, with SIZE along the x-axis, displays a better concurrence with the reference sample at lower values, which can be explained by the definition of the conditionals of OTHERS and, more generally, by having implicitly imposed a standard household composition through the definition of agent types. With higher levels of SIZE there are also higher different possibilities of populating a household, which are not fully taken into account by the more standardised compositions obtained. However, being the colours of the boxplots proportional to the number of households with that characteristic in the reference sample or hMCMC population, clearly the results for low SIZES are the only ones that matter: not only because they are more visible in the generated populations, but also since the obtained per-household distributions are more reliable.

The ten most recurrent couples of OWNERS and SPOUSES in the reference sample are illustrated in Table 8, to prove the validity of the generated matches; their frequencies in the hMCMC population are in fact quite close to those in the reference sample.

Figure 14: Distributions of intra-household indicators (SIZE along x-axis)

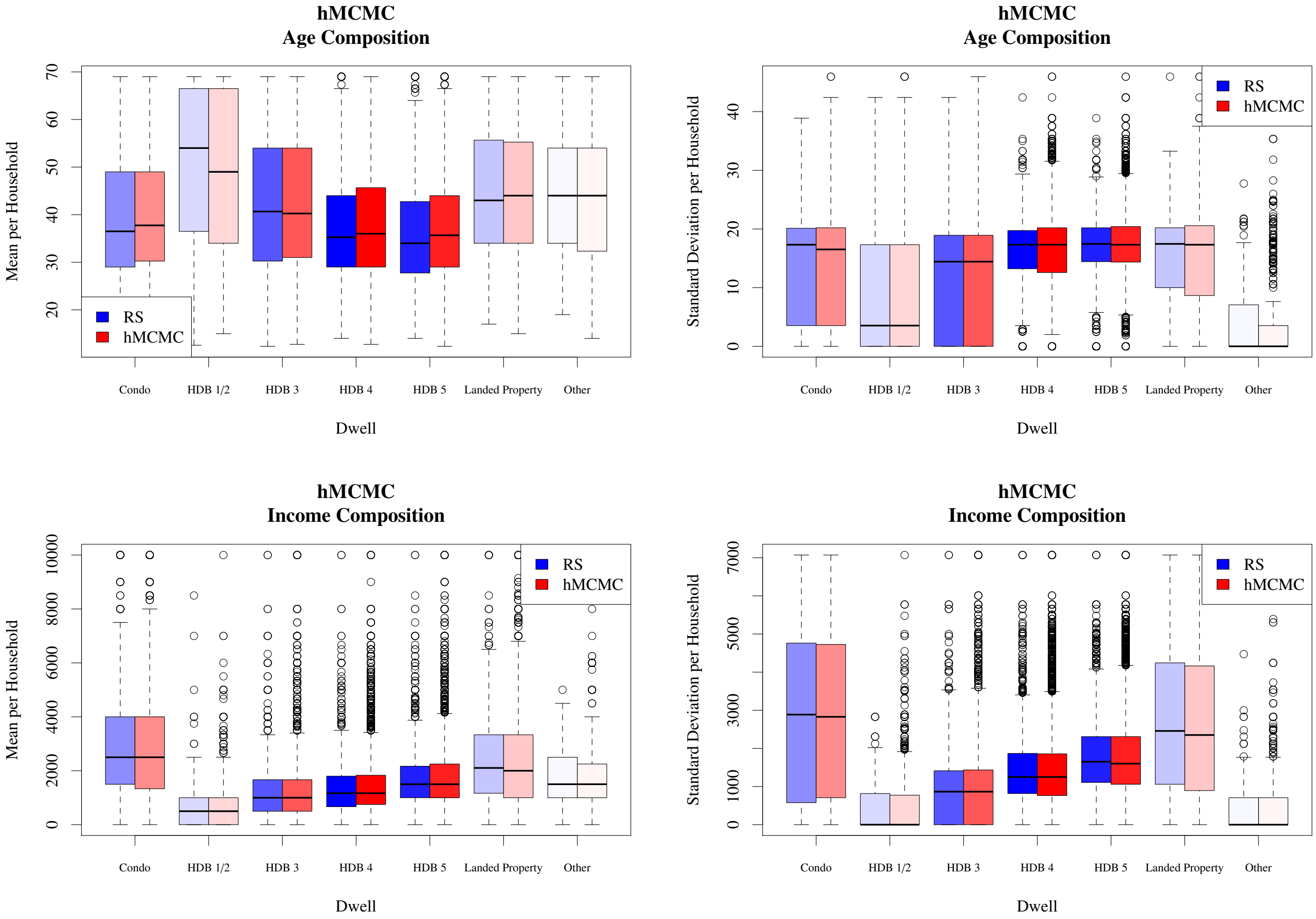


8 Conclusions

Most of the current state of the art on population generation in transportation modelling is based upon IPF, and handling hierarchies is quite an open issue. However, other techniques not suffering from IPF problems are possible and should be preferred when no control totals are available, making the application of IPF useless.

Indeed, the results demonstrate that the developed hMCMC is able to generate realistic new populations with a hierarchical structure. Being based upon MCMC and thus lacking many constraints, it produces more heterogeneous results than IPF, without its zero-cell problem. Besides, if there were some control totals to be imposed, Generalised Raking (which shares the same purpose of IPF) can be applied to fix the hMCMC populations in post-processing. This

Figure 15: Distributions of intra-household indicators (DWELL along x-axis)



was discussed in a paper derived from this thesis, Casati et al. (forthcoming).

However, MCMC-based techniques should not be applied blindly when dealing with multipolarised datasets; this is why some methods for data visualisation and clustering were illustrated, because of their ability to easily identify the main characteristics of the employed data. Nevertheless, GR post-processing can also help to overcome this flaw of MCMC, and it is still possible to later impute additional attributes or categories that needed to be dropped because of their contribution to data multipolarisation in another post-processing operation. Indeed, exactly as those variables are very polarised, they are ideally suited to be added using the set of attributes included in the synthesis as predictors, e.g. by applying statistical matching as in Müller and Axhausen (2013).

Ideally, results generated with the proposed method would also be verified against a full population, which for example would be available for the case of Switzerland. It would be particularly interesting to also test the behaviour of post-imputed attributes.

Further lines of research can deal with alternative extensions of MCMC handling hierarchies. A simple idea may be to implement iMCMC with type-based variables like $\text{AGE}_{\text{OWNER}}$, so to generate households with their fully characterized populations at once. However, the curse of dimensionality can easily break this algorithm, and one may end up handling only few possible combinations of attributes to obtain acceptable results.

Another possible development directly stems from the developed methodology of hMCMC, in which the generation of agent types living together always follows a certain order: households with an OWNER and a CHILD but no SPOUSE are not considered in this thesis, since the agent types have to be defined a posteriori. Hence, if one would have predefined types in the reference sample, it would be possible to make the generation of the other agents after OWNER not simply dependent upon NUMPAX but on some other variables, like HAS SPOUSE or NO. OF CHILDREN, which could easily be deduced from this kind of datasets. In this case, the first agent to be generated per household could actually be characterized only by household variables.

Furthermore, MCMC-based techniques can be applied for their primary purpose, Bayesian inference, to directly fit a model for the joint distribution. However, there can also be future developments involving other techniques described in this thesis outside of MCMC, particularly SOM (as detailed in Appendix D), LCA (Appendix C.2), or other data mining techniques, to formally derive agent types.

Finally, further steps would be to assign activity chains to the agents already in the population generation and to implement this in a simulation platform like `MATSim`.

Acknowledgement

I would like to express my gratitude to my supervisors for their continuous support: Mr Kirill Müller, who first believed in and made it possible to write a paper based upon this thesis (Casati et al., forthcoming), Dr Alexander Erath, for introducing me to the topic and allowing me to work at the research group on Mobility and Transportation Planning (Module 8) at the Future Cities Laboratory in Singapore, and Mr Pieter J. Fourie, for the help provided. I am thankful for their aspiring guidance, invaluably constructive criticism and friendly advice during the project work, and I hope to continue collaborating with them on other works.

I am also grateful to Professor Dr Kay W. Axhausen, Institute for Transport Planning and

Systems (IVT) at ETH Zurich, for supervising this thesis.

Besides, I would like to thank Module 8 of the Future Cities Laboratory at the Singapore-ETH Centre for Global Environmental Sustainability, where I spent three inspiring months in a truly vibrant environment.

A special thanks goes out to Mr Vahid Moosavi, PhD candidate at the Future Cities Laboratory, who advised me on the usage of SOM discussed in this thesis and whose conversations always lead to fascinating topics which I would have never discovered on my own.

I must also acknowledge Mr Paul Anderson, the Master's student at EPFL who wrote the paper about the graph-theoretic generation of hierarchies, for having helped me to implement the fitting procedure of multinomial logistic models on Biogeme, better suited for discrete choice models.

9 References

Anderson, P., B. Farooq, D. Efthymiou and M. Bierlaire (forthcoming) Association generation in synthetic population for transportation applications: Graph-theoretic solution, *Transportation Research Record*. Accepted for publication.

Bacharach, M. (1965) Estimating nonnegative matrices from marginal data, *International Economic Review*, **6** (3) 294–310.

Bar-Gera, H., K. Konduri, B. Sana, X. Ye and R. M. Pendyala (2009) Estimating survey weights with multiple constraints using entropy optimization methods, paper presented at the *88th Annual Meeting of the Transportation Research Board*, Washington, D.C., January 2009.

Barthelemy, J. and E. Cornelis (2012) Synthetic populations: review of the different approaches, *Working Paper*, **18**, CEPS/INSTEAD, April 2012.

Boriah, S., V. Chandola and V. Kumar (2008) Similarity measures for categorical data: A comparative evaluation, paper presented at the *Proceedings of the Eighth SIAM International Conference on Data Mining*, 243–254.

Bühlmann, P. and M. Mächler (2014) Computational statistics, webpage, January 2014, `https://stat.ethz.ch/education/semesters/ss2014/CompStat/sk.pdf`.

Casati, D., K. Müller, P. J. Fourie, A. Erath and K. W. Axhausen (forthcoming) Synthetic population generation by combining a hierarchical, simulation-based approach with reweighting by generalized raking, paper presented at the *Transportation Research Board 94th Annual Meeting*, Washington, D.C. Submitted for review.

Croissant, Y. (2012) *Estimation of multinomial logit models in R: The mlogit Packages*, `http://cran.r-project.org/web/packages/mlogit/vignettes/mlogit.pdf`.

Deville, J.-C., C.-E. Sarndal and O. Sautory (1993) Generalized raking procedures in survey sampling, *Journal of the American Statistical Association*, **88** (423) 1013–1020, September 1993.

Farooq, B., M. Bierlaire, R. Hurtubia and G. Flötteröd (2013) Simulation based population synthesis, *Transportation Research Part B: Methodological*, **58**, 243–263.

Greenacre, M. and J. e. Blasius (2006) *Multiple Correspondence Analysis and Related Methods*, 1st edn., Chapman and Hall/CRC, London.

Hancock, P. J. B. and C. D. Frowd (2002) Creative evolutionary systems, chap. Evolutionary Generation of Faces, 409–423, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, ISBN 1-55860-673-4.

Harman, H. H. (1976) *Modern Factor Analysis*, 3rd edn., University of Chicago Press, Chicago.

Husson, F., J. Josse and J. Pagès (2010) Principal component methods - hierarchical clustering - partitional clustering: why would we need to choose for visualizing data?, *Technical Report*, Applied Mathematics Department, Agrocampus, September 2010.

Kirshner, S. and P. Smyth (2007) Infinite mixtures of trees, paper presented at the *ICML-2007*, 417–424.

Kirshner, S., P. Smyth and A. W. Robertson (2012) Conditional Chow-Liu tree structures for modeling discrete-valued vector time series, *CoRR*, **abs/1207.4142**.

Kohonen, T. (1982) Self-Organized Formation of Topologically Correct Feature Maps, *Biological Cybernetics*, **43**, 59–69.

Lamppinen, J. and T. Kostiainen (2002) Self-organizing neural networks, chap. Generative Probability Density Model in the Self-organizing Map, 75–94, Springer-Verlag New York, Inc., New York, NY, USA, ISBN 3-7908-1417-2.

Lazarsfeld, P. F. and N. W. Henry (1968) *Latent structure analysis*, Houghton Mifflin, Boston.

Le Roux, B. and H. Rouanet (2004) *Geometric Data Analysis, From Correspondence Analysis to Structured Data Analysis*, Springer.

Mallows, C. L. (2000) Some comments on C_p, *Technometrics*, **42** (1) 87–94.

Müller, K. and K. W. Axhausen (2011a) Hierarchical IPF: Generating a synthetic population for Switzerland, paper presented at the *51st Congress of the European Regional Science Association*, Barcelona, September 2011.

Müller, K. and K. W. Axhausen (2011b) Population synthesis for microsimulation: State of the art, paper presented at the *90th Annual Meeting of the Transportation Research Board*, Washington, D.C., January 2011.

Müller, K. and K. W. Axhausen (2012) Multi-level fitting algorithms for population synthesis, *Working Paper*, **821**, Institute for Transport Planning and Systems (IVT), ETH Zurich, Zurich.

Müller, K. and K. W. Axhausen (2013) Using survey calibration and statistical matching to reweight and distribute activity schedules, *Arbeitsberichte Verkehrs- und Raumplanung*, **948**, IVT, ETH Zurich, Zurich.

Ortúzar, J. d. D. and L. G. Willumsen (2001) *Modelling Transport*, 3rd edn., John Wiley & Sons, Chichester.

Overstall, A. M. and R. King (2013) conting: An R package for bayesian analysis of complete and incomplete contingency tables, *Technical Report*, University of St Andrews.

Pearson, K. (1901) On lines and planes of closest fit to systems of points in space, *Philosophical Magazine*, **2** (6) 559–572.

Pritchard, D. R. and E. J. Miller (2012) Advances in population synthesis: fitting many attributes per agent and fitting to household and person margins simultaneously, *Transportation*, **39** (3) 685–704, May 2012.

Pukelsheim, F. and B. Simeone (2009) On the iterative proportional fitting procedure: Structure of accumulation points and l1-error analysis.

Schafer, J. L. (1997) *Analysis of Incomplete Multivariate Data*, Monographs on Statistics & Applied Probability, Chapman & Hall/CRC, Boca Raton.

Sirovich, L. and M. Kirby (1987) Low-dimensional procedure for the characterization of human faces, *Journal of the Optical Society of America A*, **4** (3) 519—-524.

Ye, X., K. C. Konduri, R. M. Pendyala, B. Sana and P. Waddell (2009) Methodology to match distributions of both household and person attributes in generation of synthetic populations, paper presented at the *Transportation Research Board 88th Annual Meeting*, Washington, D.C.

A Definitions

This section deals with the concept of categorical variables (Appendix A.1) and some of the tests which can be applied on them (Appendix A.2). Indeed, most of the demographic data in transportation modelling is expressed through this kind of variables, which is also the case of the sample considered in the case study of this thesis.

A.1 Categorical Variables

In the field of demographic studies, agents are usually represented by *categorical variables*. These variables, also called *nominal*, can assume only a finite set of distinct *categories*, which can have no intrinsic ordering. An example may be a categorical variable called SEX, with possible categories FEMALE, MALE, or N/A.

A subset of this kind of variables are the *ordinal variables*, where the categories—here also called *levels* (a term with a different meaning from Section 2.2)—present an order. An INCOME variable where the levels are not equally spaced can be an example, like MAX 1000, MAX 1500, ..., MAX 3000, MAX 4000, etc.

Finally, there is a subtype of ordinal variables, called *interval variables*, with equally spaced levels. E.g., AGE levels: AGE 4, AGE 9, AGE 14, etc.

An important concept related to categorical variables is their *contingency table* (also called *cross tabulation*). Given some categorical attributes, this table shows how many times each of their possible combinations appears in a dataset. If it considers all the categorical attributes of its reference data sample, once normalised it is the most immediate estimation of the underlying joint probability distribution, since its entries can be related to the probability weights of all possible agents (which are combinations of attributes) that can appear in the unknown full population. Storing only the contingency table of a data sample would preserve all its information.

A *marginal* of a contingency table is the sum of its entries where some variables are fixed in certain categories. By keeping constant a variable at each of its categories, hence by considering all of its marginals, the *marginal distribution* of this variable, i.e. its univariate probability distribution, would be obtained.

Finally, when some entries corresponding to different combinations of attributes are summed, e.g. to obtain the marginals or to reduce the number of categories of an attribute, it is said that the data have been *aggregated*.

A.2 Contingency Tests

From the contingency table of all the attributes and their marginal distributions, the relationships between different variables can be analysed through various techniques, like *Pearson's chi-squared test* of independence.

This test evaluates if a sample has been observed given a theoretical joint probability distribution of its variables: the so-called *null hypothesis*. In particular, it can be used to check if some variables are independent, by comparing their observed contingency table with another one respecting the theoretical distribution in which these variables are actually independent. Given the definition of independence for two random variables X and Y ($\mathrm{P}(X \cap Y) = \mathrm{P}(X) \cdot \mathrm{P}(Y)$), the contingency table of independence is formed from the outer product of their vectors of marginals.

Hence, by calling k_{ij} the non-normalised entry of the observed contingency table corresponding to the number of agents with the considered categorical variables at values i and j, the theoretical entry $\hat{k}_{ij}$ of the contingency table of independence is:

$$\hat{k}_{ij} = \left(\sum_{m_2=1}^{M_2} k_{im_2} \right) \cdot \left(\sum_{m_1=1}^{M_1} k_{m_1 j} \right) \tag{7}$$

K is the total of all entries of the observed contingency table, i.e. the total of observations. Both the contingency tables considered for this test (the observed and the theoretical) must be formed by non-normalised counts having the same total. This requirement arises from the chi-squared statistics which gives the name to this test, which is based upon squared differences between the entries of these tables, i.e.:

$$\chi^2 = \sum_{i=1}^{M_1} \sum_{j=1}^{M_2} \frac{\left(k_{ij} - \hat{k}_{ij} \right)^2}{\hat{k}_{ij}} \tag{8}$$

In the case of independent variables, χ^2 follows a certain distribution, and it is possible to compute the probability of observing the contingency table of the sample if the variables are actually independent. If this probability (the so-called *p-value*) is too low (usually lower than a threshold, like 0.05 or 0.01), the null hypothesis of independence should be discarded.

However, the chi-squared test is not limited to this kind of null hypotheses, since, as hinted above, it can check the closeness of a sample to any kind of theoretical distribution. Hence, it can also evaluate the *goodness of fit*, i.e. if two contingency tables referring to different samples have the same joint distribution. In this case, one of these tables is treated as the theoretical one: this is not just for the sake of argument, since this assumption sets how to compute the chi-squared statistics (in the denominator of Eq. (8)).

As said above, in this statistics the considered contingency tables must have the same total; therefore, if the considered samples do not have the same number of agents, one of their contingency tables should be rescaled, e.g. the one of the smaller sample, by multiplying all its entries by $\frac{K_{\text{large sample}}}{K_{\text{small sample}}}$, where these Ks are the total counts of the two tables.

If one would rather use something more precise than a chi-squared statistics, a *G-test* can be tried, of which the chi-squared test is actually an approximation. Some parameters which are related to the G-test are analysed in the appendix:

Kullback-Leibler divergence: Also called *relative entropy*, its definition is almost equal to the one of the G-test, and requires that the contingency tables of the two samples are normalised in order to treat them as joint distributions, as it can be seen in Eq. (13). The only difference between this divergence and the G-test is that the latter is defined as twice the former.

Mutual information: It is defined in Eq. (26). From its formulation it stems that mutual information is actually a particular case of Kullback-Leibler divergence, where one of the distribution to be compared assumes independence among the two variables involved. Hence, the G-test of independence can be expressed as $2 \cdot \mathrm{I}(X,Y)$, where $\mathrm{I}(X,Y)$ is the mutual information between variables X and Y, which vary respectively along the rows and columns of the contingency tables organised as matrices. Since demographic samples are usually characterised by more that two attributes, X and Y can be made of combinations of the attributes actually appearing in the samples. About this, refer also to Section 2.1.1.

Another error estimation coming from the contingency table is the *Standard Root Mean Square*

Error.

$$\text{SRMSE} = \sqrt{\sum_{m_1=1}^{M_1} \sum_{m_2=1}^{M_2} \dots \sum_{m_N=1}^{M_N} \left(f_{m_1 m_2 \dots m_N} - \hat{f}_{m_1 m_2 \dots m_N} \right)^2 \cdot (M_1 \cdot M_2 \cdot \dots \cdot M_N)} \tag{9}$$

In this equation, $f_{m_1 m_2 \dots m_N}$ and $\hat{f}_{m_1 m_2 \dots m_N}$ are the relative frequencies of a combination of attributes which appear in the reference and synthetic population, with N the number of attributes. $M_1 \cdot M_2 \cdot \dots \cdot M_N$ is the product of the numbers of categories each attribute can take, i.e. the total of possible different combinations (equivalent to the total of distinct agents).

Equation (9) is slightly different that the usual expression of SRMSE reported e.g. in Farooq et al. (2013), and it is similar to the one of Müller and Axhausen (2011a). This formulation highlights the fact that the more attributes are considered, the larger this error becomes, taking into account the intrinsic difficulty of generating more varied populations.

In this thesis SRMSE is extensively used to validate the populations generated with different methods. About the chi-squared test, instead, its application to the variables of the employed dataset as a test of independence is briefly reported in the next section, discussing the employed data for the case study.

B Theory

This section goes into details about the theory behind the main methods for population synthesis considered in this thesis: the milestone method in the literature, Iterative Proportional Fitting (Appendix B.1), and the implemented Markov Chain Monte Carlo (Appendix B.2).

B.1 Iterative Proportional Fitting

To better describe how Iterative Proportional Fitting works, the contingency table of the available data sample should be organised as a matrix (*two-way contingency table*); e.g., the rows can take into account the combinations of all variables except one, and the remaining attribute can vary along the columns. However, IPF can also be illustrated without the contingency table being in this form, and the desired totals to be imposed are not restricted to marginals, but can more generally be obtained when collapsing the table along all other dimensions. This is why the term control totals is usually preferred in this context.

The description of the IPF algorithm will make it clear why it has been such a successful technique. Indeed, it simply consists of iteratively modifying the entries of the contingency table of the reference sample under these expressions:

$$\begin{aligned} \hat{k}_{ij}^{(2\eta-1)} &= \frac{\hat{k}_{ij}^{(2\eta-2)} K_i}{\sum_{j_*=1}^{J} \hat{k}_{ij_*}^{(2\eta-2)}} \\ \hat{k}_{ij}^{(2\eta)} &= \frac{\hat{k}_{ij}^{(2\eta-1)} K_j}{\sum_{i_*=1}^{I} \hat{k}_{i_*j}^{(2\eta-1)}} \end{aligned} \tag{10}$$

η is the loop index; as previously said, $\hat{k}_{ij}^{(0)} = k_{ij}$. K_i are the control totals corresponding to the marginals of the full population; if some are unknown, marginals of the reference contingency table should be used in their place. Indeed, since all the possible marginals appear in Eq. (10), some values must always be imposed for them under IPF.

The convergence of this loop is not proven here, but there is a large literature about it, like Pukelsheim and Simeone (2009). However, its limit is discussed to provide with some hints on the theoretical foundation of IPF: indeed, the fitted contingency table is a *maximum likelihood estimation*, and the related log-likelihood is defined as:

$$\begin{aligned} \log \mathcal{L}\left(\hat{k}_{1,1}, \hat{k}_{1,2}, ..., \hat{k}_{IJ}\right) = \log \prod_{i=1}^{I} \prod_{j=1}^{J} \left(\frac{\hat{k}_{ij}}{\hat{K}}\right)^{k_{ij}} &= \sum_{i=1}^{I} \sum_{j=1}^{J} \left(k_{ij} \cdot \log \frac{\hat{k}_{ij}}{\hat{K}}\right) \\ &= \sum_{i=1}^{I} \sum_{j=1}^{J} \left(k_{ij} \cdot \log \hat{k}_{ij} - k_{ij} \cdot \log \hat{K}\right) \\ &= \sum_{i=1}^{I} \sum_{j=1}^{J} \left(k_{ij} \cdot \log \hat{k}_{ij}\right) \end{aligned} \tag{11}$$

The products and sums are along the rows and columns of the contingency table, where k_{ij} are the reference entries and $\hat{k}_{ij}$ the values to be obtained under IPF. $\hat{K}$ is the total sum of the IPF fitted entries, and this is why there are fractions $\frac{\hat{k}_{ij}}{\hat{K}}$ in the log-likelihood: these values are treated as probabilities. Indeed, a contingency table can be considered as the joint probability distribution of its categorical variables.

The last term of the log-likelihood has been removed because it does not depend upon any $\hat{k}_{ij}$ (and the log-likelihood should be maximised varying these values). However, some Lagrange multipliers should also be added to obtain Eq. (10), constraining the result of the maximisation to all control totals.

Another modification which has to be applied to derive the IPF algorithm is to substitute $\log \hat{k}_{ij}$ with $u + v_i + w_j$, where $\sum\limits_{i=1}^{I} v_i = \sum\limits_{j=1}^{J} w_j = 0$. This expression is called *model of independence*, since it does not consider direct interaction terms z_{ij} between i and j, and is the underlying fundamental assumption of IPF, which explains why the fitted contingency table can be written as the outer product of two vectors: $\mathrm{e}^u \cdot \left(\mathrm{e}^{\vec{v}} \otimes \mathrm{e}^{\vec{w}}\right)$, where e^u is just a scalar. It is interesting to compare this model of independence with Pearson's chi-squared test of independence and the related Eq. (7).

B.1.1 Generalised Raking

A similar method to IPF, which does not suffer from the zero-cell problem as well, is the so-called *Generalised Raking*. Although it shares the same aim of IPF (adapt the contingency table of a sample to some known marginals of the full population), its theoretical basis is radically different. Instead of maximising the likelihood, it aims at minimising the sum of the distances between the reference and the new weights, with some Lagrange multipliers to impose the known marginals (Deville et al., 1993). As its name suggests, it is more general than IPF because various distances can be used; if they are squared differences, a similar expression to the SRMSE (Eq. (9)) would be obtained.

Thus, GR can be applied to impose marginals on the synthetic populations as a post-processing operation. Indeed, as it is currently being explored, the fit obtained from GR is better than IPF according to the minimisation of the entropy, as defined here:

$$\mathrm{S} = \sum_{i=1}^{I} \sum_{j=1}^{J} \left(\hat{k}_{ij} \cdot \log \hat{k}_{ij} - 1\right) \tag{12}$$

Notice its similarity with the likelihood, but without the reference entries: it is a posteriori analysis, computed after having performed the algorithm, which measures the distance of the fitted contingency table from a matrix constituted only of ones (the minimum of S is indeed obtained if all $\hat{k}_{ij} = 1$). Hence, this parameter tests the possible heterogeneity of new populations generated from this fitted contingency table as joint distribution. A line of research currently carried out is to fit different GR tables, varying the considered control totals (which do not always come from reliable sources) in order to find the optimal ones for the best balance between entropy and SRMSE (similarly to what is done e.g. by Mallows's C_p, Mallows (2000)).

Nevertheless, the IPF contingency table minimises another form of entropy, the *Kullback-Leibler divergence* (or *relative entropy*) (as it can be seen in Müller and Axhausen, 2013, p. 4). It is defined as the expectation of the logarithmic difference between two probability distributions, where the expectation is considered through one of these probabilities. For discrete distributions and using $k_{i,j}/K$ as probability weights for the reference contingency table, $\hat{k}_{i,j}/\hat{K}$ for the new one, its expression is:

$$\sum_{i=1}^{I}\sum_{j=1}^{J}\left(\frac{\hat{k}_{ij}}{\hat{K}}\cdot\log\frac{\hat{k}_{ij}/\hat{K}}{k_{ij}/K}\right)=\sum_{i=1}^{I}\sum_{j=1}^{J}\left(\frac{\hat{k}_{ij}}{\hat{K}}\cdot\log\frac{\hat{k}_{ij}}{\hat{K}}-\frac{\hat{k}_{ij}}{\hat{K}}\cdot\log\frac{k_{ij}}{K}\right) \tag{13}$$

Therefore, IPF provides the closest approximation to the underlying joint probability distribution in the classical information-theoretic sense. Another method which minimises the Kullback-Leibler divergence is discussed in Appendix D.1, while in Appendix A.2 a test whose expression is almost equal to the Kullback-Leibler divergence is mentioned.

B.2 Markov Chain Monte Carlo

MCMC is a generic term for Monte Carlo methods based upon sampling from a Markov chain. In this thesis, it has been restricted to identify an algorithm for population generation employing Gibbs sampling with categorical variables, which, however, works with continuous variables as well (the conditionals would be fitted under a model which is not MLLR).

Still unlike this thesis, the usual aim of Gibbs sampling is actually to fit some probability distributions: having imposed certain parametric models, their parameters are sampled along the Markov chain, rather than new data. This approach can be the only way to fit probability models to high-dimensional data, and it is called *statistical inference*.

Its idea is the following: given some parameters $\boldsymbol{\theta}$ of the model, instead of considering a *maximum likelihood* estimate, $\hat{\boldsymbol{\theta}}_{\mathrm{ML}}(\mathbf{x}) = \arg_{\boldsymbol{\theta}} \max \mathrm{P}(\mathbf{x} \mid \boldsymbol{\theta})$ where $\mathbf{x}$ is the dataset and $\mathrm{P}(\mathbf{x} \mid \boldsymbol{\theta})$

the probability model to be fitted, a *maximum a posteriori* estimation is computed:

$$\hat{\boldsymbol{\theta}}_{\mathrm{MAP}}(\mathbf{x}) = \arg_{\boldsymbol{\theta}} \max \mathrm{P}(\boldsymbol{\theta} \mid \mathbf{x}) = \arg_{\boldsymbol{\theta}} \max \frac{\mathrm{P}(\mathbf{x} \mid \boldsymbol{\theta}) \cdot \mathrm{P}(\boldsymbol{\theta})}{\mathrm{P}(\mathbf{x})} = \arg_{\boldsymbol{\theta}} \max \mathrm{P}(\mathbf{x} \mid \boldsymbol{\theta}) \cdot \mathrm{P}(\boldsymbol{\theta}) \tag{14}$$

In the second passage the so-called *Bayes' rule* was applied, $\mathrm{P}(\boldsymbol{\theta} \mid \mathbf{x}) \cdot \mathrm{P}(\mathbf{x}) = \mathrm{P}(\mathbf{x} \mid \boldsymbol{\theta}) \cdot \mathrm{P}(\boldsymbol{\theta})$ (its derivation is given by Eq. (25)), to rewrite the so-called *posterior probability* $\mathrm{P}(\boldsymbol{\theta} \mid \mathbf{x})$.

In this frame, certain models for the so-called *prior probabilities* $\mathrm{P}(\mathbf{x} \mid \boldsymbol{\theta})$ and $\mathrm{P}(\boldsymbol{\theta})$ should be assumed (the former is the model to be fitted on the dataset, the latter a "reasonable" probabilistic distribution). Besides, $\mathrm{P}(\mathbf{x} \mid \boldsymbol{\theta}) \cdot \mathrm{P}(\boldsymbol{\theta}) \propto \mathrm{P}(\boldsymbol{\theta} \mid \mathbf{x})$ holds according to Bayes' rule. Hence, the conditional probability for each parameter θ_i can be obtained by integrating all the other parameters from $\mathrm{P}(\boldsymbol{\theta} \mid \mathbf{x})$, or, more practically, in the following way:

1. Drop everything from the estimation of $\mathrm{P}(\boldsymbol{\theta} \mid \mathbf{x})$ which does not contain θ_i.
2. Normalise the resulting equation.

Finally, given all the conditionals $\mathrm{P}(\theta_i \mid \theta_1,...,\theta_{i-1},\theta_{i+1},...,\theta_N,\mathbf{x})$, Gibbs sampling of the parameters $\boldsymbol{\theta}$ can be performed, starting from some seed parameters. After a burn-in period and thinning the results, uncorrelated parameters $\boldsymbol{\theta}^{(k)}$ are obtained with the right "joint" distribution $\mathrm{P}(\boldsymbol{\theta} \mid \mathbf{x})$ (which is the joint distribution of all parameters $\boldsymbol{\theta}$, still conditioned by the observed data $\mathbf{x}$).

The most likely value of a parameter θ_i (its mode) can be selected by choosing the sampled value that occurs most often, and this is exactly a maximum a posteriori estimation of this parameter (since it has the highest recurrence, i.e. it maximises the corresponding probability). Because the parameters are usually continuous, it is often necessary to bin the sampled ones into a finite range of possible values in order to get a meaningful estimate of the mode. More commonly, however, the expected value of the sampled parameter (its mean) is chosen, which is an estimator that takes advantage of the additional data about the entire distribution that is available from the sampling. Nevertheless, if a distribution is multimodal, the mean may not be a meaningful point, and any of the modes is typically a better choice.

In other words, a Bayesian inference returns a whole probability distribution for each parameter (a sort of "second-order" distribution) in a similar way to bootstrapping, where the aim is to obtain confidence levels on the parameters of the model (which can still be done in this frame).

However, there are cases in which Gibbs sampling does not work, since it relies on using conditionals like $\mathrm{P}(\theta_i \mid \theta_1,...,\theta_{i-1},\theta_{i+1},...,\theta_N,\mathbf{x})$. This problem is extensively discussed in Section 6.2.1. In these situations, a Markov chain with the *Metropolis-Hastings algorithm* should rather be considered, which does always work. Details about it are given in the next section, since they can clarify some general aspects of Markov chains.

B.2.1 Metropolis-Hastings Algorithm

The purpose of a Markov chain, particularly evident with the Metropolis-Hastings algorithm, is to generate draws according to a desired *equilibrium distribution* $\mathrm{P_{eq}}(\mathbf{y})$. A Markov chain converges to this equilibrium distribution (1) if its stationary distribution $\mathrm{P_{st}}(\mathbf{y})$ exists and is unique (i.e. if this Markov chain univocally converges) and (2) if this $\mathrm{P_{st}}$ is equal to $\mathrm{P_{eq}}$.

1. The condition of detailed balance $\mathrm{P_{st}}(\mathbf{y}) \cdot \mathrm{P}(\mathbf{y} \to \mathbf{y}') = \mathrm{P_{st}}(\mathbf{y}') \cdot \mathrm{P}(\mathbf{y}' \to \mathbf{y})$, where $\mathrm{P}(\mathbf{y} \to \mathbf{y}')$ is the *transition probability* of going from sample $\mathbf{y}$ to $\mathbf{y}'$ along the Markov chain, is necessary for the existence of $\mathrm{P_{st}}$, while its uniqueness is guaranteed if the chain is *ergodic*: from one draw $\mathbf{y}$, any other draw $\mathbf{y}'$ can be reached, i.e. there is always a sequence of intermediate draws $\mathbf{y}_{\text{int}}^{(\mathbf{k})}$, $k = 1,2,...,K$, so that all the transition probabilities $\mathrm{P}\left(\mathbf{y}_{\text{int}}^{(\mathbf{k})} \to \mathbf{y}_{\text{int}}^{(\mathbf{k+1})}\right)$ are nonzero (and then also their product $\mathrm{P}\left(\mathbf{y} \to \mathbf{y}_{\text{int}}^{(\mathbf{1})}\right) \cdot \mathrm{P}\left(\mathbf{y}_{\text{int}}^{(\mathbf{1})} \to \mathbf{y}_{\text{int}}^{(\mathbf{2})}\right) \cdot ... \cdot \mathrm{P}\left(\mathbf{y}_{\text{int}}^{(\mathbf{K})} \to \mathbf{y}'\right)$).
2. In the detailed balance, $\mathrm{P_{st}}$ is then substituted by $\mathrm{P_{eq}}$.

Furthermore, in the frame of Metropolis-Hastings algorithm, the transition probabilities are factorised into $\mathrm{P}(\mathbf{y} \to \mathbf{y}') = \mathrm{g}(\mathbf{y} \to \mathbf{y}') \cdot \mathrm{A}(\mathbf{y} \to \mathbf{y}')$, where $\mathrm{g}(\mathbf{y} \to \mathbf{y}')$ is the *proposal distribution*, conditional probability of proposing the change from $\mathbf{y}$ to $\mathbf{y}'$, and $\mathrm{A}(\mathbf{y} \to \mathbf{y}')$ the *acceptance distribution*, conditional probability of accepting this change. Thus, modifying the detailed balance accordingly, the following equation is obtained:

$$\frac{\mathrm{A}(\mathbf{y} \to \mathbf{y}')}{\mathrm{A}(\mathbf{y}' \to \mathbf{y})} = \frac{\mathrm{P_{eq}}(\mathbf{y}')}{\mathrm{P_{eq}}(\mathbf{y})} \frac{\mathrm{g}(\mathbf{y}' \to \mathbf{y})}{\mathrm{g}(\mathbf{y} \to \mathbf{y}')} \tag{15}$$

In order to satisfy this equation, the Metropolis choice imposes (but there are other possibilities):

$$\mathrm{A}(\mathbf{y} \to \mathbf{y}') = \min\left(1, \frac{\mathrm{P_{eq}}(\mathbf{y}')}{\mathrm{P_{eq}}(\mathbf{y})} \frac{\mathrm{g}(\mathbf{y}' \to \mathbf{y})}{\mathrm{g}(\mathbf{y} \to \mathbf{y}')}\right) \tag{16}$$

Hence, in the context of Bayesian inference, at every step of the Markov chain a new proposed set of parameters $\boldsymbol{\theta}'$ is sampled, given the current $\boldsymbol{\theta} = \boldsymbol{\theta}^{(k)}$, under the acceptance probability:

$$\mathrm{A}\left(\boldsymbol{\theta} \to \boldsymbol{\theta}' \mid \mathbf{x}\right) = \min\left(1, \frac{\mathrm{P_{eq}}\left(\boldsymbol{\theta}' \mid \mathbf{x}\right)}{\mathrm{P_{eq}}\left(\boldsymbol{\theta} \mid \mathbf{x}\right)} \frac{\mathrm{g}\left(\boldsymbol{\theta}' \to \boldsymbol{\theta} \mid \mathbf{x}\right)}{\mathrm{g}\left(\boldsymbol{\theta} \to \boldsymbol{\theta}' \mid \mathbf{x}\right)}\right) \tag{17}$$

Therefore, a random number $r \in \mathcal{U}_{(0,1)}$ is drawn: if $r < \mathrm{A}\left(\boldsymbol{\theta} \to \boldsymbol{\theta}' \mid \mathbf{x}\right)$ then $\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}'$, otherwise $\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}$.

Finally, the motivations for applying MCMC-based techniques to a contingency table, respecting their usual aim of Bayesian inference (hence, fitting a model), are listed:

- As stated before, similarly to bootstrapping, to obtain second-order distributions for the parameters of a model (and estimate them through their mode or mean).
- To check if some variables are independent by fitting a model which assumes a relationship among them: if the associated parameters result nonzero, then a dependence does exist.
- To solve the IPF zero-cell problem (Section 2.1.1) by considering the sampling zeros as missing data. Indeed, they can be predicted after having fitted a model to the observed, nonzero cells (as also suggested in Müller and Axhausen, 2011a,b). This goal and the previous one (tests of independence) can be achieved with the `R` package `conting` (Overstall and King, 2013).
- To even directly fit a contingency table, based upon some model. This has only been suggested in Farooq et al. (2013); Anderson et al. (forthcoming), citing Schafer (1997) and stating that this approach has never been tried in transportation modelling.

Some research may be undertaken in these directions, as also suggested in Appendix D.

C Data Visualisation and Clustering

In this thesis, three techniques were employed for data visualisation and clustering, i.e. to recognise recurrent patterns in the behaviour of the variables of a data sample and identifying sets of similar agents. These methods have not usually been applied to the analysis of demographic samples in transportation modelling, although they have a long history and are still popular nowadays. They are:

Multiple Categorical Analysis The categorical version of *Principal Component Analysis* (Pearson, 1901), which is a very famous technique which can highlight linear relationships among variables (Le Roux and Rouanet, 2004; Greenacre and Blasius, 2006). It can also be used for detecting clusters of data, but its primary aim is visualisation.

Latent Class Analysis The categorical version of *Factor Analysis* (Harman, 1976), another very famous technique again considering linear relationships which is, in a way, equivalent to PCA (Lazarsfeld and Henry, 1968). However, its main goal is fitting a regression to show latent variables (analogue to clustering).

Self-Organising Map Theoretically it can only be used with continuous variables, but this did not really cause problems with the categorical attributes of this thesis (Kohonen, 1982). It is better than the previous techniques since it can also detect nonlinear relationships. Its primary aim is again visualisation, with clustering as an additional implementation.

However, these technique can be used without limiting them to their original, descriptive purposes, as it is extensively discussed in Appendix D.

C.1 Multiple Correspondence Analysis

In the analysis of the employed demographic sample (Section 5), the first data visualisation technique applied to the Singapore sample was a *Multiple Correspondence Analysis* (Le Roux and Rouanet, 2004; Greenacre and Blasius, 2006). This is the "categorical" version of a *Principal Component Analysis* (Pearson, 1901); in the following, some insight is provided into PCA and how it can be extended to categorical variables.

Indeed, this technique only works with continuous variables, and it aims at reorganising high-dimensional data (with many variables) to fewer dimensions while preserving their information. Its idea is to exploit some directions as a new basis system through which to express the data, which can also identify only a subspace of the space spanned by the original variables, where every attribute varies along a certain dimension; this concept of attributes' space is also used in Section 6.2.1. If this subspace is actually proper, i.e. it does not cover the whole original space, a dimensionality reduction is obtained, without information loss.

To start, a *Singular Value Decomposition* should be performed on some dataset X, organised in a KxN matrix where the columns are the different variables and every row corresponds to an agent. From each column of X, the empirical mean of that variable has been subtracted.

$$X = U\Sigma V^{\mathrm{t}} \tag{18}$$

U and V^{t} are KxK and NxN orthogonal matrices, while Σ is a KxN diagonal matrix where the number of nonzero diagonal entries (the so-called *singular values*) equals the rank of X.

Equation (19) below shows that the singular values are square roots of the eigenvalues of matrix $X^{\mathrm{t}}X$ and that the columns of V are its eigenvectors. In fact, $X^{\mathrm{t}}X$ is symmetric, hence diagonalisable; it is the covariance matrix of the dataset X, since its means have been rescaled to zeros.

$$X^{\mathrm{t}}X = V\Sigma U^{\mathrm{t}}U\Sigma V^{\mathrm{t}} = V\Sigma^2 V^{\mathrm{t}} \tag{19}$$

Thus, X can be rewritten into the following:

$$X = TV^{\mathrm{t}} \tag{20}$$

The columns of V are the new dimensions (*principal components*) along which to represent the data; their characterising vectors $\boldsymbol{v}$ are also called *loadings* (which, in this context, is a synonym for eigenvector of $X^{\mathrm{t}}X$). The rows of the new dataset T are formed by the coordinates (*scores*) obtained from multiple projections of each original agent (row $\boldsymbol{x}^{\mathbf{t}}$) along every loading $\boldsymbol{v}$.

Being V an NxN matrix, the number of the obtained eigenvectors is equal to N, the number of original attributes (after all, V is a coordinate transformation matrix for the whole space spanned by the original variables). However, not all the eigenvectors usually have a nonzero eigenvalue: their number is equal to the rank of X, which is only at most N (indeed, this should be the smaller dimension, rather than K: usually, there are less variables than agents). Thus, there is a dimensionality reduction without information loss if rank $(X) < N$ (*lossless compression*), and the new coordinates will only be null along these directions with null eigenvalues, being all the agents $\boldsymbol{x}^{\mathbf{t}}$ orthogonal to them ($\boldsymbol{x}^{\mathbf{t}}\boldsymbol{v} = 0$).

In fact, the absolute value of each eigenvalue is proportional to the variance of the data along the corresponding eigenvector. The principal components represented by the eigenvectors with the largest eigenvalues store most of the information, and only these dimensions may be considered in the new dataset, without losing too much information (*lossy compression*).

Since PCA is actually based upon the transformation of a coordinate system, its result should be a representation of all the agents as points on an N-dimensional space, with the basis that mostly stresses their distribution in this space. However, this cannot clearly be represented graphically with $N > 3$.

Hence, the results of a PCA are usually 2D plots with the axes corresponding to the two eigenvectors with largest eigenvalues. In addition to the data points, they can also mark the positions of the attributes themselves (i.e. the rows of V, which are the old N attributes expressed in the new basis of principal components) and, if an MCA is considered, of categories as well. Besides, clusters formed by data points might become easily identifiable, so that data can be classified according to their most common characteristics (exactly the purpose of data mining). For an example of this in the context of MCA, refer to its application to the categorical Singapore sample, discussed in Section 5.2.

Indeed, there were multiple purposes to apply an MCA in this thesis:

- A "simple" graphical analysis of the considered variables, to possibly exclude some which were too much interdependent (Section 6.2). However, it is clear from the foundation of PCA, based upon linear algebra, that only linear relationships between variables are considered, which is probably the greatest flaw of this technique.
- Identify some clusters from the plots to define a latent variable representing the agent types for the hierarchical generation, although this was not precisely implemented (Section 7.1.1).
- Exploit the *eigenagents* from MCA to approximate the underlying joint distribution and simulate a population generation, which however remained a suggestion because of some intrinsic flaws of this idea (Appendix D).

To conclude, some remarks about how to actually derive MCA from the theoretical framework of PCA. All the motives to use an MCA listed above fundamentally requires to represent every agent as a point on an N-dimensional space. In other words, it is necessary to define a distance on the data, and, for the simple Euclidean distance, continuous attributes are needed. Hence, to extend PCA to categorical variables, either of these approaches can be followed:

1. Expand the categorical attributes into dummy variables, i.e. instead of considering a categorical variable with M categories, handle M variables taking values 0 or 1: the former if the considered categorical variable does not assume that category, and vice versa (this was also explained in Section 4.1.1). Then, when the empirical means are computed in order to subtract them from the data, which is the starting point of PCA, values in $[0, 1]$ are obtained, and the new data can then take continuous values. `MCA` from `R` package `FactoMineR` and `poLCA` from the package with the same name (which however implements the categorical method of the next section) have this characteristic.

2. Define a new distance of the attributes' space which is not Euclidean any more, not requiring continuity neither an intrinsic ordering of the variables involved. These distances are usually based upon the contingency table of the categorical data; for more information on this topic, refer to Boriah et al. (2008).
   A difficulty of this approach compared to Item 1 is related to the means: is it necessary to subtract them from the data if the categorical nature of the variables involved is preserved? And, if so, how to compute them? A solution may be to convert the categories to numerical values by imposing a certain ordering, so that it would actually become possible to compute these means. Since the role of subtracting them from the data in PCA is only to centre the variables at zero (which becomes their new mean value), and hence no meaning to these means is assigned, the arbitrariness of the numerical ordering imposed should not be a source of concern.

These points are also valid for the next section, which describes a better technique to identify clusters on some dataset.

C.2 Latent Class Analysis

A *latent variable* is not directly observable in a dataset, but can be inferred from it through a mathematical model. It characterises the data in a way which is not evident from the actually observed variables (also called *manifest variables* in this context), like the clusters arising from an MCA. These inferred variables can be either continuous or categorical.

For hierarchical generation without predefined agent types in the dataset, it would be useful to formally identify a categorical latent variable which specify them. To this aim, while in the actual research a different path was pursued, *Factor Analysis* (Harman, 1976), a technique closely related to PCA (so much that the existence of a distinction is even a topic of discussion), could have been employed, in particular in its doubly categorical version (for both manifest and latent variables), *Latent Class Analysis* (Lazarsfeld and Henry, 1968).

FA assumes that the latent variables which should be identified, also called *factors*, correspond to some unobserved random variables underlying the observations through the following model:

$$\boldsymbol{x_i} = t_{i,1}\boldsymbol{v_1^*} + ... + t_{i,N^*}\boldsymbol{v_{N^*}^*} + \boldsymbol{\epsilon_i}\,, \qquad i = 1,2,...,N \tag{21}$$

$\boldsymbol{x_i}$ is one of the manifest variables from which its sample mean has already been subtracted,

$t_{i,j}$ some unknown constants, $\boldsymbol{v}_{\boldsymbol{j}}^{\boldsymbol{*}}$ the latent variables which should be found, and $\boldsymbol{\epsilon_i}$ a vector of independent and normally distributed errors with null mean and finite variance, which may not be the same for all i.

Written in matrix terminology, the previous equation becomes:

$$X = TV^* + E \tag{22}$$

Notice its similarity with the previous MCA equation (Eq. (20)). The main difference with PCA is that there the error term E is absent; while the aim of PCA is to describe the available data and illustrate their linear patterns through a coordinate transformation, FA fits a regression model on them. However, both tend to dimensionality reduction (FA by assuming that the number of latent variables N^* is $< N$).

To complete this introduction on Factor Analysis, the assumptions imposed in Eq. (22) are reported:

- V^* and E are independent. Indeed, in the frame of FA, the columns of V^*, the latent variables, are random variables as well (differently from PCA).
- $\mathrm{E}\left[V^*\right] = 0$
- $\mathrm{Cov}\left[V^*\right] = I$, i.e. latent variables are uncorrelated.

About the error E, there is usually the additional assumption of its normality. This is another similarity between FA and PCA; indeed, it was not evident in its description (Appendix C.1), but this latter technique works better with normal errors: principal components are guaranteed to be independent if the variables of the dataset are jointly normally distributed.

Regardless of these details, clearly the main assumption of FA is to impose Eq. (22) on the data, thus assuming linear relationships between variables. Again, this assumption was also made by PCA, and is avoided by the data visualisation technique illustrated in the next section.

C.3 Self-Organising Map

To analyse the data and, as it is shown in Appendix D, infer its joint probability distribution, a *Self-Organising Map* (Kohonen, 1982) was created, very useful for high-dimensional data like the Singapore sample, where every agent has many attributes. However, note that with this name the technique itself is also addressed, not only its result.

SOM aims at identifying patterns of agents in a dataset like the previously explored techniques; however, unlike them, it does not assume linear relationships between variables. SOM is not a recent technique and it has already enjoyed a considerable popularity, having been applied in many fields, but never in the frame of transportation modelling.

Its idea is to dispose certain points, called *neurons*, on a 2D grid (in the past, with low computing power, 1D segments were preferred); they are generally disposed uniformly. The choices related to the number of neurons and their positions affect the final result, which is not univocal like in MCA. Each neuron corresponds to a vector whose size is equal to the number of considered attributes in the dataset, and which indeed constitutes a combination of them.

At the beginning random attributes' vectors are assigned to the neurons; to make the convergence of the SOM algorithm faster, values evenly sampled from the subspace spanned by the two PCA eigenvectors with largest eigenvalues may be used. This also prevents SOM from returning a unique map.

A loop is then started in which at every step, given an agent of the dataset, the following is done:

1. Find its *Best Matching Unit*, i.e. the neuron whose attributes' vector is closest to the attributes of the current agent, in the sense that their Euclidean distance is minimised.
2. Modify not only the attributes' vector of the BMU, but also those of the surrounding neurons in the grid.

Item 1 implies that continuous variables must be used; however, it is shown that this is not a problem for categorical variables as soon as an ordering is imposed on them—which can even be based upon the alphabetical order of their categories' names—and convert these names to their assigned numeric indices.

The size $\sigma(t)$ of the neighbourhood of neurons to be modified is usually chosen according to the formula:

$$\sigma(t) = \sigma_0 \exp(-t/\tau_\sigma) \tag{23}$$

t is the index of the current loop step. Notice that the neighbourhood size becomes smaller while the loop proceeds: this is necessary since everything started with random attributes' vectors which should be quickly adapted to the dataset; afterwards, it is only a matter of tuning them.

The attributes' vector $\boldsymbol{w}_\omega$ of the neuron ω belonging to the BMU neighbourhood Ω_{BMU} are modified according to:

$$\Delta\boldsymbol{w}_\omega = \eta_0 \exp\left(-t/\tau_\eta\right) \cdot \exp\left(-S_\omega^2/(2\sigma^2(t))\right) \cdot \left(\boldsymbol{x}^{(k)} - \boldsymbol{w}_\omega\right), \qquad \omega \in \Omega_{\mathrm{BMU}} \tag{24}$$

The first term with η is a monotonically decreasing learning coefficient (like the neighbourhood size $\sigma(t)$). The second is a Gaussian function whose numerator is the square of the Euclidean distance S_ω of neuron ω from the BMU (not a distance between their vectors, but between their coordinates in the grid). $\boldsymbol{x}^{(k)}$ is a vector of the attributes of the agent currently explored in the dataset. Clearly, this function aims at making the attributes' vector $\boldsymbol{w}_\omega$ closer to the combination of variables characterising the current agent.

At the end of this loop, the final SOM, a discrete 2D grid where each point is assigned to a multidimensional vector, is displayed by splitting it into several sub-maps. Their number is equal to the total of considered variables, and each one is coloured according to the entries of the neurons' vectors corresponding to one variable. In some sub-maps, similar coloured shapes may be noticed, identifying clusters of agents which behave similarly with respect to different attributes; in other words, many agents assuming certain categories for some variables.

The agents of the original dataset may also be plotted in the final SOM, by identifying their BMUs. The specific coordinates of these points are not important, since they depend upon the initial random attributes' vectors employed. What does matter are the relative distances between agents, which allow to identify similarities and differences in the high-dimensional data.

It is evident that the only assumption made by SOM is to deal with continuous variables, since it requires to compute Euclidean distances to find the BMUs. However, an ordering of purely categorical data can be imposed in any way, as soon as the chosen mapping is fixed. At the end, one can approximate the obtained numeric values to their closest integers, and then identify the corresponding categories. If the dataset is only categorical, the borders between shapes of different colours in the final SOM will be sharp since it will mostly contain almost integer values, with very sudden changes among them. In any case, as said above, what matters are the similar patterns arising in different sub-maps, which are present regardless of the chosen mapping.

D Joint Probability Distribution Estimation

The idea of this section is to employ techniques previously discussed for different aims to derive a model for the joint probability distribution of the attributes. This would solve the problem of population generation and, if more variables were considered, handling hierarchies as well. Indeed, a joint distribution on the household level but with variables referring to their inhabitants, like the AGE of the oldest inhabitant, would allow to automatically generate whole households with their agents included.

To this aim, Chow-Liu Tree is first discussed in Appendix D.1. This is a classic approach to estimate high-dimensional joint distributions, although it was used for purely descriptive purposes in the thesis since its core foundation is, to a certain extent, too close to IPF, as it is shown.

However, some techniques which have been applied in the thesis to other aims may be considered for this task.

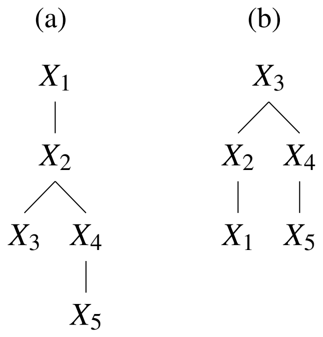
For example, it would be possible to use Gibbs sampling for Bayesian inference (Appendix B.2) to fit a model for the whole joint distribution, which is the usual case when dealing with many parameters. However, exactly because of this large number of parameters, the assumptions made would also be relevant, even more than IPF, which was also based upon joint distributions (and which imposed that the fitted contingency table can be rewritten as the outer product of two vectors).

Another possibility is given by data mining techniques, which are becoming more promising nowadays since the necessary computational power they require has been achieved, and they do not also make too many assumptions on the underlying distribution.

For example, a sort of joint distribution can be derived through MCA (Appendix C.1): normalised eigenvalues can be used as probability weights of the eigenagents. Synthesis of agents can then come from a weighted average of eigenagents, similarly to what is usually done in computer graphics with *eigenfaces* (Hancock and Frowd, 2002). However, in this case a generation from MCA would not be optimal since (1) the total of eigenagents is only equal to the number of considered variables (which, in the Singapore sample, is large to directly handle its joint distribution, but not so much for the number of obtainable eigenagents, related to the variability offered by an MCA-based generation), and (2) a linear combination for categorical variables is not even well defined.

A better idea would be to use SOM to derive the joint distribution. The methodology is simply

Figure 16: An example of a Chow-Liu Tree, shown with two different orderings



introduced in Appendix D.2 and might be developed in a future work, with some possible extensions (Appendix D.3).

D.1 Chow-Liu Tree

The standard approach to derive a joint probability distribution of many variables is a *Chow-Liu Tree* (Kirshner et al., 2012). This graph sketches the relationships among attributes appearing in a dataset in a way that it is then easy to approximate their joint distribution (and, in the context of this thesis, generate new populations accordingly).

E.g., assume the CLT displayed in Fig. 16(a), where X_i are the considered attributes. The theory behind CLTs implies that the joint distribution $\mathrm{P}(X_1, X_2, ..., X_5)$ can be approximated with $\mathrm{P}(X_1)\,\mathrm{P}(X_2 \mid X_1)\,\mathrm{P}(X_3 \mid X_2)\,\mathrm{P}(X_4 \mid X_2)\,\mathrm{P}(X_5 \mid X_4)$; in other words, starting from a node j, multiply $\mathrm{P}\left(X_j\right)$ with conditional probabilities corresponding to the edges of the tree, with the variable closer to X_j as predictor and the other as response (i.e. the closer conditions the farther). Hence, the only kind of interactions among variables in this approximation are second-order, in the sense that at most two variables can appear in each factor.

Since a tree is an undirected graph (more specifically, an undirected graph in which any two vertices are connected by exactly one path of edges), the way the considered sample tree has been organised is arbitrary, and any node can be positioned at the top (*parent node*). An example is provided in Fig. 16(b), and the approximation of the joint distribution is now $\mathrm{P}(X_3)\,\mathrm{P}(X_2 \mid X_3)\,\mathrm{P}(X_1 \mid X_2)\,\mathrm{P}(X_4 \mid X_3)\,\mathrm{P}(X_5 \mid X_4)$. This multiplicity can be justified according to Eq. (25):

$$\left.\begin{array}{l} \mathrm{P}(X \mid Y) = \frac{\mathrm{P}(X \cap Y)}{\mathrm{P}(Y)} \\ \mathrm{P}(Y \mid X) = \frac{\mathrm{P}(X \cap Y)}{\mathrm{P}(X)} \end{array}\right\} \Longrightarrow \mathrm{P}(X \mid Y) \cdot \mathrm{P}(Y) = \mathrm{P}(X \cap Y) = \mathrm{P}(Y \mid X) \cdot \mathrm{P}(X) \tag{25}$$

X and Y are two distinct attributes which are not considered in specific categories; hence, the probabilistic terms which appear in this equation are functions. The final formula is the well-known *Bayes' rule*.

It is noteworthy that fitting a CLT on a dataset is quite a fast procedure, since its time complexity quadratically depends upon the number of attributes N, but linearly upon the size K of the available data: $\mathrm{O}\left(KN^2M_{\mathrm{max}}^2\right)$, where M_{max} is the largest number of possible categories over all variables (Kirshner and Smyth, 2007). Without entering into details about this algorithm, its theoretical foundation would rather be mentioned, that is to provide the approximation of the joint distribution with second-order interactions minimising the global Kullback-Leibler divergence (Kirshner et al., 2012, as explained in), which is given in Eq. (13). To this aim, the edges of a CLT are chosen so that the sum of the corresponding *pairwise mutual informations* between the attributes at the vertices of its edges is maximal. The pairwise mutual information of the couple of categorical attributes (X,Y) is:

$$\mathrm{I}(X,Y) = \sum_{m_x=1}^{M_x} \sum_{m_y=1}^{M_y} \left(\mathrm{P}\left(X = x_{m_x} \cap Y = y_{m_y}\right) \cdot \log \frac{\mathrm{P}\left(X = x_{m_x} \cap Y = y_{m_y}\right)}{\mathrm{P}\left(X = x_{m_x}\right) \cdot \mathrm{P}\left(Y = y_{m_y}\right)} \right) \tag{26}$$

$\mathrm{I}(X,Y)$ is a measure of the relationship between X and Y, expressed through their joint distribution: the more these two variables are related, the likelier an edge between them will appear. It also takes into account nonlinear relationships, and it becomes zero when the variables are independent since the argument of the logarithm would become equal to one. Besides, its expression is symmetric, which further confirms the “undirected” nature of the approximation provided by a CLT.

To understand why a minimal global Kullback-Leibler divergence is obtained through mutual informations, Appendix A.2 should be considered, where a test for contingency tables expressible in terms of both Kullback-Leibler divergence and mutual information is mentioned. Since CLTs are based upon minimising the former parameter while also taking into account only second-order interactions, the resulting joint probability distribution should at least be similar to the IPF contingency table, which also minimises this measure but under other assumptions. Therefore, because IPF was not chosen for implementation in this thesis, the CLT fitted on the Singapore data given in Section 5.4 was not used to derive its joint probability distribution, but only to obtain some information about the relationships among variables.

D.2 Self-Organising Map for Joint Probability Distribution: Methodology

After having fitted a SOM, the neurons constitute a set of standard agents, forming an estimation of the underlying joint probability distribution. The population generation is then performed through uniformly draws of these neurons, which are the new agents.

However, SOM is usually suited for continuous variables. For categorical variables, as shown in the description of MCA (Appendix C.1), two ways are possible: dummy variables or a non-Euclidean distance. In the `R` package `kohonen` a way to treat categorical variables with SOM is already implemented, based upon the *Tanimoto distance*, which is proportional to the number of equal attributes between two agents.

With respect to IPF, the other approach explored here which is based upon fitting a joint distribution, the advantage of SOM is that its result is not unique, but it varies according to the number of neurons and the type of grid along which they are disposed.

SOM would also prevent the zero-cell problem of IPF, while preserving its computational advantage of considering a limited number of possible combinations, discarding very unrealistic ones. Indeed, the total of considered neurons is equal to the number of final possible combinations of attributes, which can also be not unique. Hence, exactly because of this point, a flaw of SOM should be that it excludes the outliers of the reference data sample. For instance, the agent of AGE 14 who lived alone present in the Singapore sample could never be considered in the new populations.

D.3 Self-Organising Map for Joint Probability Distribution: Extensions

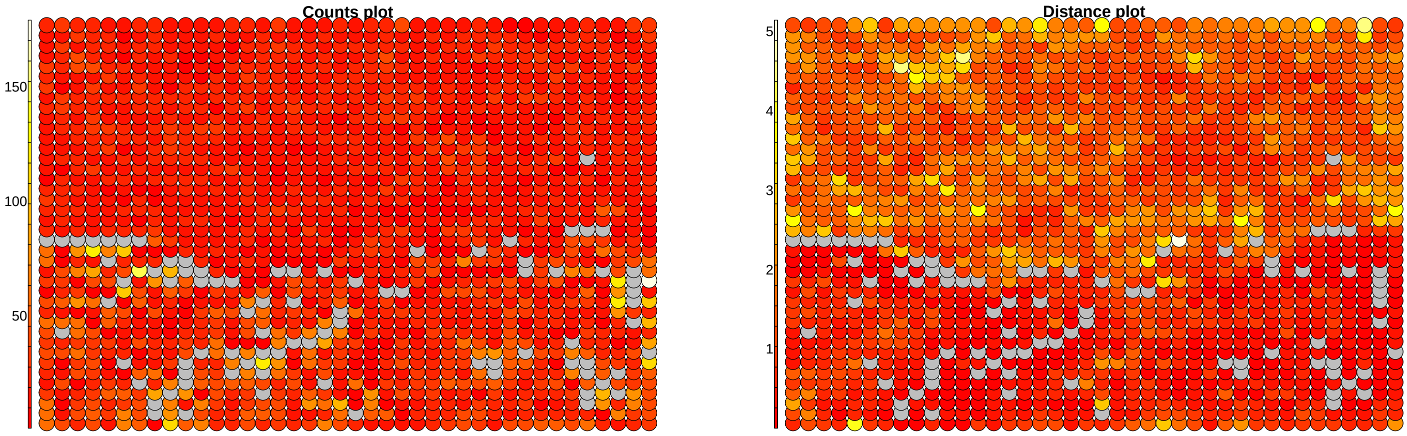
Being SOM originally adapted for continuous variable, a direct extension would be to add the latitude and longitude position of the households, derivable from their postal code in the reference data. This would help to create better transportation models, where the location of people is actually very important.

One may also obtain a better joint probability distribution by assigning some weights to the set

Figure 17: Counts and mean distances of original agents fitted to the SOM through their BMUs

(a) Counts plot of SOM

(b) Distance plot of SOM



of fitted neurons (which however already contains this information, since there can be multiple neurons for the same combination of attributes). These weights may be the frequencies with which every neuron is chosen as Best Matching Unit for the reference agents. About this point, please consider Fig. 17(a); this counts plot refer to the SOM of the reference data provided in Section 5.3. Here, with a limited number of neurons (40x40 on a square grid), all have been assigned to at least one agent of the reference sample. If this were not the case, to derive the weights a default count of 1 can be assigned to all neurons.

Figure 17(b) instead reports its distance plot, showing for each neuron the mean distance of the agents having that point as BMU (grey colours are zeros).

To derive hierarchical populations with SOM, a possibility would be to add variables for the inhabitants of the household (agent variables on the household level), as mentioned above.

Another way stems from the fact that, in the final SOM, neurons which are close in the grid represent similar agents, given all their attributes. Hence, the clusters of neurons in this SOM correspond to groups of agents that reasonably share the same type in their household. It would then be possible to sample from each of these SOM clusters and then join these agents together in the same household.

This can also be done by directly fitting separate SOMs for subsamples of the reference data defined according to agent types, where the later types should also be characterised by variables referring to the other agents living together (as in Table 5). The draws of these agents should then be conditioned by the previous ones (e.g., sampling a SPOUSE only among the corresponding neurons characterised by a certain $\text{AGE}_{\text{OWNER}}$), similarly to hMCMC.

Finally, a completely different approach to derive a joint distribution from SOM is described in Lamppinen and Kostiainen (2002). The idea illustrated there is to exploit the distances between the data points and the corresponding BUMs and set them proportional to the negative log-likelihood of the SOM generation, which is actually dependent upon the unknown joint probability distribution.

E Extension of Graph-Theoretic Solution

To generate hierarchies under MCMC, given its recent application in transportation modelling, the only previous work is Anderson et al. (forthcoming). There it is described an optimisation approach based upon a graph-theoretic solution, handling hierarchies as a post-processing operation, after the population generation. A similar method was also suggested in Barthelemy and Cornelis (2012), where various types of agents are subsequently linked to their household through combinatorial optimisation (Section 2.1.2).

This graph-theoretic approach has been extended by defining agent types and dealing with large households. However, its discussion has been moved here because, having tried this method on the Singapore dataset, it did not prove successful, not being theoretically solid—the underlying model was too simplistic—and, despite this, computationally infeasible as well.

E.1 Extension of Graph-Theoretic Solution: Methodology

In Anderson et al. (forthcoming), three agent types were considered, which were already defined in their dataset: OWNERS and SPOUSES. Their separate populations were generated under iMCMC, together with a population of HOUSEHOLDS.

First, a *bipartite graph* was built between HOUSEHOLDS and OWNERS. Then, if the former were not single-person, SPOUSES were also assigned to HOUSEHOLDS through another bipartite graph. Both these graphs were built with the *Hungarian algorithm*, which aims at finding edges such that the sum of their weights is maximal.

To this aim, a full matrix of the weights of all possible connections has to be defined (*edge matrix*). In the reference paper, these weights were assigned through a sort of multinomial logistic model, choosing one household variable as response (their SIZE or TYPE), and some of the other attributes of both HOUSEHOLDS and agent type to be connected as predictors. However, in that paper an unusual definition of this model is employed, assuming that the coefficients of the attributes are constant for all the alternative categories of the response, and only the intercept is allowed to vary:

$$\begin{gathered}\log \mathrm{P}\left(\hat{X} = x_j \mid X_1 = x_1, ..., X_N = x_N\right) = \beta_{j0} + \sum_{n=1}^{N} \beta_n \cdot x_n \\ j = 1, 2, ..., M\end{gathered} \tag{27}$$

$\hat{X}$ is the selected response attribute. These logarithms are then directly used as entries of the edge matrix.

Notice that all the coefficients (apart from the intercept) are the same for all the categories of the response variable. In Anderson et al. (forthcoming)—and many other works which employ the same model—this is still called "multinomial logit model", like the one which is defined in Eqs. (1) and (2) in the frame of MCMC. However, this is a misnomer, as reported in some documentation about the `R` package `mlogit` (Croissant, 2012). Besides, this kind of model is not always feasible: it can easily result in too large intercepts compared to the other coefficients, exactly because these are the only parameters allowed to vary (which however does not happen in the reference paper).

Furthermore, when estimating the edge matrix of the SPOUSES, the approach of this paper does not take into consideration the already connected OWNERS, thus creating some odd matches. To distinguish between OWNERS' and SPOUSES' connections they simply use different coefficients to build their edge matrices.

To handle the further complexities of the dataset considered in this thesis, this algorithm was extended to deal with more agent types (Section 7.1.1). OWNERS with household characteristics were generated, and then other types were assigned according to their NUMPAX. For each type, a set of coefficients to estimate the edge weights was fitted under the canonical multinomial logit model (Eq. (1)), to prevent the flaws described above.

However, the idea of considering already connected agents in the edge matrix was discarded. Indeed, even the simple model with NUMPAX as response and the household attributes of OWNERS and those of the type to be connected as predictors (but without relations to other agents living together) proved too expensive, as it is discussed in the next section.

E.2 Extension of Graph-Theoretic Solution: Implementation

Similarly to the implementation of MCMC (Sections 6.1 and 7.1), two languages were used: `Biogeme` for model fitting and `Java` for the bipartite graph.

`Biogeme` was chosen analogously to the reference paper, despite the difficulties to handle it and its slowness. The employed `Java` code implementing the Hungarian algorithm is freely available on the Internet, written by Konstantinos A. Nedas (University of Maine).

However, this graph-theoretic solution proved to be a dead end: even with the simplistic model previously discussed, it still took more than five hours to generate hierarchies on a population with the same size of the reference sample, using an Intel Core i7 processor.

Besides, and most importantly, it was extremely memory consuming. Indeed, `Java` was not able to deal with, e.g., 10^4 x 10^4 matrices, where the dimensions correspond to the size of the generated subpopulations of agent types, thus making this approach not feasible for handling large populations (which would be the actual aim of population generation).

Since this approach should return worse results than hMCMC, given its theoretical flaw of not taking into account already connected agents, no results are provided.

F Markov Chain Monte Carlo: Code

The code reported below implements a step of the MCMC algorithm, valid for both iMCMC and hMCMC, where the right probabilities are also computed from the parameters fitted in `R`.

`t` is the type of the agent `X` to be generated (`t`= 0 is an owner or simple iMCMC).
`nResAtt` and `nConAtt` allow more generality to the implemented models, and are the number of variables to synthesize and of those which are also used as predictors. The former is always larger than the latter, and the variables must be ordered so that the first `nConAtt` are both responses and predictors (after `nResAtt`, the other variables only condition, coming from already generated agents).
`nCat` is an array of numbers of categories per attribute, and `coeffs` a multidimensional array of coefficients fitted in `R` organised as follows: *response attribute* `X` *response category* `X` *predictor attribute* `X` *predictor category*.

```
private static void stepMCMC( int t )
```

```
{
  for ( int i=0; i<nResAtt; i++ )
  {
    double[] prob = new double[nCat[i]];
    prob[0] = 1;
    double C = 1;
    for ( int j=1; j<nCat[i]; j++ )
    {
      for ( int k=0; k<nResAtt; k++ )
      {
        if ( k != i )
        {
          prob[j] += coeffs[i][j-1][k][X.get(t)[k]];
          // There is 0 in pivot categories of other attributes.
        }
        else
        {
          prob[j] += coeffs[i][j-1][k][0];
          // There is the intercept in position "i" (the attribute
            being currently generated).
        }
      }
      for ( int k=nResAtt; k<nCat.length-t*nConAtt; k++ )
      // Consider household variables to condition "prob[j]".
      {
        prob[j] += coeffs[i][j-1][k][X.get(0)[(nConAtt-nResAtt)+k]];
      }
      for ( int k=0; k<t; k++ )
      // Consider variables of already connected agents to condition
        "prob[j]".
      {
        for ( int l=0; l<nConAtt; l++ )
        {
          prob[j] +=
            coeffs[i][j-1][nCat.length+(k-t)*nConAtt+l][X.get(k)[l]];
        }
      }
      prob[j] = Math.exp(prob[j]);
      C += prob[j];
    }
    C = 1/C;
    double rand = Math.random();
    for ( int j=0; j<nCat[i]; j++ )
    {
      prob[j] *= C;
      rand -= prob[j];
      if ( rand < 0 )
      {
        X.get(t)[i] = j;
        break;
      }
    }
  }
}
```