ABSTRACT

Title of Thesis: USING DEEP GENERATIVE MACHINE LEARNING METHODS TO GENERATE SYNTHETIC POPULATION

Zhichao Yang, Master of Science in Engineering, 2022

Thesis Directed By: Professor Cirillo Cinzia, Department of Civil and Environmental Engineering

Population synthesis is an important area of research aiming at generating synthetic data about households and individuals that would be representative of real large populations. Scholars in different fields have worked on synthetic population generation: statisticians, computers scientists, economists, social scientists, and engineers. In transportation modeling, synthetic agents are a key input for agent-based models, that are gradually replacing zone-based aggregate four steps models. Traditional methods for population synthesis include Iterative Population Fitting

(IPF), that weights sample data until marginals for the variables of interest match official statistics (often from CENSUS) at a certain geographical area. Recently, Machine Learning algorithms have been tested and compared to IPF, which suffers from several well-known limitations. In this M.S. thesis, advanced deep generative machine learning methods are applied to generate synthetic populations, including CTGAN and TVAE. CTGAN is an advanced GAN algorithm that models tabular data distribution and sample rows from the underlying distribution. It has been shown that CTGAN can solve issues that challenge conventional GAN model, including mixed data types, non-Gaussian distributions, multimodal distributions, learning from sparse one-hot-encoded vectors and highly imbalanced categorical columns. TVAE is also an advanced VAE model that adapts VAE to tabular data by using preprocessing and modifying the loss function. As a case study, this research applies these two machine learning methods to generate synthetic population based on a sample from the American Community Survey relative to the State of Maryland. To demonstrate the performance of the proposed methods, we compare our results to those obtained with IPF and Bayesian Network using metrics that evaluate the ability of the population synthetizer to reproduce the dependency structure and the marginals in the real population and to solve the problem of zero cells in IPF.

USING DEEP GENERATIVE MACHINE LEARNING METHODS TO GENERATE SYNTHETIC POPULATION

by

Zhichao Yang

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
[Master of Science]
[2022]

Advisory Committee:
Professor [Cirillo, Cinzia], Chair
[Fabian Bastin]
[Takumi Saegusa]

Table of Contents

Table of Contents ii
Chapter 1: Introduction 1
Chapter 2: Literature Review 5
2.1 IPF 5
2.2 MCMC 16
2.3 Machine Learning 18
Chapter 3: Data Description 23
Chapter 4: Methodology 34
4.1 IPF 34
4.2 Bayesian Network 36
4.3 CTGAN 40
4.4 TVAE 43
4.5 Copula 44
4.6 Discrete Case 46
Chapter 5: Results 48
5.1 Evalutation Methods 48
5.2 State Level 49
5.3 County Level 52
5.4 Puma Level 54
Chapter 6: Conclusion 59
Contributions 61
References 62

Chapter 1: Introduction

Micro-agents are key inputs to most agent-based models in transportation modeling. In microsimulation several methods have been explored and their performance tested to assess the quality of the data synthetized with respect to individual and household characteristics of the real population. Iterative Proportional Fitting was first adopted by scholars in transportation and applied to several case studies. Recently, other techniques, like Markov Chain Monte Carlo methods and Machine Learning techniques (e.g. GAN, VAE and Bayesian Network) were also applied to the problem of population synthesis. My research also focuses on Machine Learning methods and explores advanced deep generative methods to generate synthetic populations. The machine learning methods we explore in this research include Bayesian Network, CTGAN and TVAE and we also compare the machine learning methods with IPF. Here Bayesian Network is a probabilistic model that represents a set of variables and their conditional dependencies with a directed acyclic graph (DAG). CTGAN is an advanced GAN model that models tabular data distribution and sample rows from the distribution. Specific techniques are put forward in CTGAN model, which are mode specific normalization and conditional generator to solve the issues that conventional GAN encounters. Just like CTGAN, TVAE is an advanced VAE model which adapts VAE to tabular data by using the same preprocessing and modifying the loss function.

In addition, we also apply a copula framework to all the machine learning methods (Bayesian Network, CTGAN and TVAE) to improve the quality of synthetic population generated by these methods. Copula is a multivariate cumulative

distribution function (CDF). The marginal probability distribution of each variable is uniform on the interval [0,1] for the CDF. Copulas can fully capture the dependence structure between the input variables. For the discrete case, which means the variables in the dataset have discrete supports, the normalized vector does not have uniform marginals and thus the decoding step will not allow us to recover the marginals of the target population. Therefore, an intermediate uniform sampling step is added to my population synthesis procedure, which is to sample uniformly each discrete variable between its current value and the largest value lower than it.

To evaluate the performance of the synthetic population generated by all these methods, we also use the metrics including standardized root mean squared error (SRMSE) and sampling zero and marginal fits to evaluate each method's performance. SRMSE captures whether the synthetic combination appears in the real data thus assessing the fitting of the multi-dimensional dependencies. It is used to evaluate the accuracy of the synthetic population generated. Sampling zero counts the combinations of variables which are in the test set but not in the training set. It is used to evaluate the diversity of the synthetic population generated. Marginal fit is also used to make the comparison for all the methods for each single variable in the dataset.

In the case study, the research proposed a large-scale application relative to the State of Maryland and makes use of data extracted from the American Community Survey (ACS). The marginals were collected from Decennial census data and the Internal Revenue income (IRS). Three experiments were conducted to compare all the methods' performance in generating synthetic population, which were conducted at

the state level, county level and puma level. For the state level, we select Maryland as the full population. For the county level, we select Anne Arundel county to be the full population. And for the puma level, the source is puma 1201 and the target is census tract 702204. We compare the performance of all the methods at different geographical levels from accuracy and diversity two aspects. The accuracy is evaluated by computing the average SRMSE for all possible i-uples of variables range from 1 to the number of variables in the dataset. We have ten variables in the dataset, so we will compute the average SRMSE for all possible ten uples of variables. And the number of sampling zero is used to evaluate diversity. Higher value in sampling zero means more diversity. The comparison among each method's performance and comparison before and after applying copula framework will finally be discussed in each experiment. Based on the results figured out in three experiments, we will summarize the generative model that is most recommended to generate synthetic population under different situations (for which geographical level, for accuracy or diversity) and whether the technique of copula is helpful in improving generative models' performance in generating synthetic population not only for the methods explored in the research but also for the future new models.

The remaining part is organized as follows: Chapter 2 is the literature review that summarizes the previous work in population synthesis area, including IPF, MCMC and machine learning methods including Bayesian Network, VAE and GAN. Chapter 3 describes the data we use in the research. The sample dataset was obtained from the American Community Survey (ACS), and the marginals were collected from Decennial census data and the Internal Revenue income (IRS). Chapter 4 introduces

the methodologies we use to generate synthetic populations including IPF, Bayesian Network, CTGAN, TVAE and copula framework that is applied to the three machine learning methods including Bayesian Network, CTGAN and TVAE. The technique that aimed at discrete variables is also discussed in this chapter. Chapter 5 presents the results of my experiments. We evaluate the synthetic population generated by the methodologies mentioned in chapter 4 at the state level, county level and puma level. We use root mean squared error (SRMSE), sampling zero and marginal fits to evaluate the performance of the synthetic population generated by different methods. Chapter 6 presents the major findings and draws the final conclusions based on the results in chapter 5.

Chapter 2: Literature Review

Population synthesis is an important topic that discusses how to generate data that are synthetic but at the same time enough to be representative of the real population in the transportation modeling area since micro-agents in synthetic populations are a key input to most agent-based models. Therefore, many techniques are put forward to deal with the problem of population synthesis.

2.1 IPF

Beckman et al 1996[1] gives the standard IPF (Iterative proportional fitting)-based procedure to generate synthetic population. For the census data used in the paper, one major source is from census tract summary tables in STF-3A (A collection of summary tables of demographics, such as the number of persons per household, for census tract or census block group sized areas and is often used in transportation studies) that is used in the creation of synthetic households, which are divided into three categories: family households, nonfamily households and group quarters. Another major source of census data is PUMS, which are a representative 5% sample of households and group quarters from the PUMA. Weights are assigned to each household and person in the sample so that weighted summary statistics can be formed. To construct cross-classified tables of demographics for each census tract in the area of study, IPF is used to complete these multiway tables. In situations where the marginal totals of a multiway table are known and a sample from the population which generated these totals is provided IPF gives a constrained maximum entropy estimate of the true proportions in the population multiway table. In addition, IPF

estimates maintain the same odds ratios as those in the sample table in the absence of any marginal information to the contrary. For the IPF procedure, it considers all census tracts and parts of tracts that contribute to the PUMS. First, the marginal tables for all n census tracts in the PUMS are added. Then an m-dimensional multiway table is obtained by IPF of the PUMS against the summed marginals. It can be viewed as the construction of an (m+1)-dimensional table. The first m dimensions of the table are the m marginals, and the last dimension is created by "stacking" the n tables for the census tracts. The final estimated tables are obtained by IPF of a (m+1)-dimensional table with entries of 1 against these marginal tables. Then the synthetic population of households is constructed by selecting entire households from the PUMS in the proportion to the estimated probabilities given in the multiway table obtained by the technique mentioned above. The number of households to be generated of each demographic type is determined for each census tract. These numbers can be obtained either by multiplying the total number of households by the probabilities in the estimated multiway table, or by drawing the numbers at random according to these probabilities. Once the number of households of each demographic type to be selected is determined, households with different demographics are considered separately.

In **Eluru, Naveen, et al 2008**[2], the paper stresses that conventional wisdom has long indicated that the linkages among socioeconomics, land use, and the transportation is significant for realistic forecasts of travel demand. However, conventional methods use aggregate forecasts of socioeconomics and land use to feed into travel models and thus cannot acquire the multitude of interactions that arise over

space and time among the different decision makers. To overcome the shortcomings of conventional approaches, some integrated land-use transportation modeling systems, which lay emphasis on the interactions among population socioeconomic processes, the households' long-term behaviors, and the economic markets within which households act, are developed by researchers. At the same time, the paper highlights three issues these modeling systems need to consider: First, over a long-range multi-year forecasting time frame, individuals go through different life cycle stages and household compositions; Second, as the socioeconomic process unfolds, individuals may begin/finish schooling, move onto different life-cycle stages, enter/exit the labor market, and change the jobs ; Third, interactions between households and other decision makers(such as businesses, institutions, and real estate developers) within the housing, labor, and transportation markets ultimately shape land-use patterns. Based on these, the paper discusses their efforts at designing and developing a Comprehensive Econometric Microsimulator for Urban Systems (CEMUS) that is behaviorally oriented and focuses on the underlying decisions of households and individuals, and businesses, and developers, which manifest themselves in the form of aggregate passenger travel patterns. CEMUS takes aggregate socioeconomics for the base year, activity-travel environment characteristics for the base year and policy actions for the future year as base year inputs. The process of simulation for the base year is to first feed aggregate-level socioeconomics data into the synthesis population generator (SPG) to produce a disaggregate-level synthetic dataset describing a subset of the socioeconomic characteristics of the households and individuals residing in the study area. The base-

year socioeconomic attributes that are difficult to synthesize (or cannot be synthesized) directly from the aggregate socioeconomic data for the base year are simulated by the Comprehensive Econometric Microsimulator for SocioEconomics, Land-use and Transportation System (CEMSELTS). Then the base year socioeconomic data and activity-travel environment characteristics are run through the Comprehensive Econometric Microsimulator for Daily Activity-travel Patterns (CEMDAP) to obtain individual-level activity-travel patterns. The activity-travel patterns are subsequently passed through a dynamic traffic micro-assignment scheme to determine path flows, link flows, and transportation system level-of-service by time of day. The resulting transportation system level-of-service characteristics are fed back to CEMSELTS along with the socioeconomic data to generate revised individual activity-travel pattern. This "within-year" iteration is continued until consistency and based-year equilibrium is achieved. The next phase as the population go forward in time (one year for example) begins with CEMSELTS updating the population, urban form, and the land-use markets. An initial set of transportation system attributes is generated by CEMSELTS for this next time step based on three types of contents: (1) the population, urban form, and the land-use markets; (2) the transportation attributes from the previous year in the simulation; (3) the future year policy scenarios provided as input to CEMUS. The CEMSELTS outputs are then put into CEMDAP, which interfaces with a dynamic microassignment scheme in a series of consistency/equilibrium iterations for the next time step to obtain the "one time step" outputs. The loop continues for several time steps forward until the socioeconomics, land-use, and transportation system path/link flows and

transportation system level of service are obtained for the forecast year specified by the analyst. During this iterative process, the effects of the prescribed policy actions can be evaluated based on the simulated network flows and speeds for any intermediate year between the base year and the forecast year. These are the processes of simulation for the base year and the years after that. In addition, the structure of the modeling system for population updating within the CEMSELTS module of the CEMUS framework that is focused on in this paper includes two major subsystems: (1) the migration model system and (2) the socioeconomic evolution model system. The migration model system comprises the models that determine the movement of existing households out of the study region. The socioeconomic evolution (SE) model system includes three major components: (1) individual-level evolution and choice models; (2) household formation models; (3) household-level long-term choice models. Once the population is determined from the migration model system, the SE model system will focus on simulating the changes in the population. These two model systems together determine the changes in population characteristics, residential pattern and employment patterns over the course of one simulation year.

In **Bar-Gera, Hillel, et al 2009**[3], the paper discusses the challenge faced when transportation professionals want to accurately expand the survey households to represent the population. When each individual in the surveys is considered as a separate response unit, the determination of weights can be accomplished by simply calculating the ratio between the proportion of the subgroup in the population and their proportion among survey respondents. Here subgroups are divided from the entire population. This approach assumes that we know about the information about

the proportion of each subgroup in the entire population exogenously to the survey. In addition, the proper consideration is required about which characteristics to control for and how to divide the population into subgroups. However, the weights are determined by a straightforward closed form computation once these choices are made. The choice of weights may be slightly more complicated when exogenous information is available on the marginal distribution of each control variable separately rather than the joint distribution of the combination of all of the variables of interest. And determining the weights for travel surveys is even more challenging because the response unit is not an individual but a household instead. Typically, separate exogenous information exists about the distribution of household characteristics and person characteristics. The distribution should be given by complete household structure in order to apply the simplistic weighting scheme described above, but distributions given by complete household structure are not available or not relevant in most practical cases. Therefore, finding a weight for each household so that the distributions of characteristics in the weighted sample match the exogenously given distributions in the population for both household and person characteristics becomes an important goal. No closed form computation seems to be there unlike the simple weights described above and an iterative process is probably needed. In addition, it is only possible to attain distributions as close as possible to the target distributions since perfect match cannot be obtained when the exogenous data is inconsistent. Finally, the same target distributions can be obtained by many different sets of weights, but the assumption should be that they receive the same weight. Based on these problems, the paper presents an Entropy Maximization

methodology to estimate household survey weights to match the exogenously given distributions of the population, including both household and person. The Entropy Maximization methodology can provide solutions to constrained optimization problems, which means the problem of estimating survey weights can also be formulated as a constrained optimization problem, where one is attempting to minimize the difference between the weighted sample distributions and known population distributions across a set of control variables at both the household and person-levels. It is applied as the strict formulation in order to choose the most reasonable set of weights subject to the constraints in the equation where the frequency matrix multiply by the weight column vector should be equal to the column vector which contains the exogenous information about the distribution of household characteristics and about the distribution of person characteristics. For the frequency matrix, each column corresponds to a sample household and each row within a column gives the distribution from a sample household to a certain population characteristic. The objective function is to minimize the negative of the Entropy function. Since it's a strictly convex problem, there is only a unique solution. The solution can be obtained using a coordinate-by-coordinate search algorithm. However, the cases exist where a perfect match between the weighted sums and the exogenous distributions of population characteristics cannot be found because of infeasibility in the constraints. Therefore, the paper also proposes a relaxed formulation to deal with certain cases. Each of the constraints is relaxed using a relaxation factor and a new vector representing the chosen marginals. The chosen marginals are figured out by multiplying the original marginals and the relaxation factor. Then the objective

function from the strict formulation is modified by adding a new term including this relaxation factor.

In **Ye, Xin, et al 2009**[4], the paper points out the disadvantage of standard iterative proportional fitting (IPF) procedure and puts forward a heuristic approach called the Iterative Proportional Updating (IPU) to solve the issue. The IPF procedure yields the maximum entropy estimates of the joint distribution under the constraints of the given marginal distributions (**Wong 1992**[5]) and then it is used to estimate joint distributions of household attributes (**Beckman et al 1996**[1]). Sample frequency tables used in the study were generated from the Public Use Microdata Sample (PUMA) data using critical household attributes, for which marginal distributions were available from the Census Summary Files. Synthetic households were then generated by randomly drawing households from the PUMA according to the estimated joint distributions. The synthetic population then consisted of all persons from the selected households. However, this process doesn't guarantee the consistency for the person attributes of the interest and thus fail to match the known distributions of person characteristics from the Census Summary Files. The reason is that the procedure will naturally result in two different sets of weights, one set for matching household distributions and one set for matching person-level distributions. Except under extreme unrealistic conditions, household weights will never match person weights. As a result, a synthetic population that is generated based on the application of household weights will yield joint distributions of person attributes that do not match the person-level marginal distributions. The inconsistency in person-level distributions can be reduced if the household weights are adjusted based on the

person weights obtained from the IPF procedure. It is suitable if the households in the sample are small, which is to minimize the objective functions that represent different measures of inconsistency between the weighted frequency of the household/person type and the given frequency distribution constraints that need to be met. However, the number of households in the samples in real-world can be thousands and thus making the solution of this optimization problem computationally intractable using traditional optimization methods. Thus, the IPU algorithm is put forward to solve the problems mentioned above. The IPU algorithm starts by assuming equal weights for all households in the sample. The algorithm then proceeds by adjusting weights for each household/person constraint in an iterative fashion until the constraints are matched as closely as possible for both household and person attributes. The updated weights will first perfectly satisfy household level constraints. Then the weights are updated to satisfy person constraints. The completion of all adjustments to weights for one full set of constraints is defined as one iteration. After enough number of iterations, the weighted sums almost perfectly match the household type and person type constraints and the household weights for households belonging to a particular household type are no longer identical. At the meantime, the household weights have been reallocated to match the given constraints for both weighted household and person sums. In summary, IPU algorithm works with joint distributions of household and person attributes derived using the IPF method, and then iteratively adjusts and reallocates weights across households such that both household and person-level attributed distributions are matched as closely as possible. The algorithm is flexible because it can accommodate a multitude of household and person-level variables of

interest and meets dual household- and person-level constraints with reasonable computational times. These advantages make IPU able to generate more rational synthetic population over previous synthetic population generation algorithms.

Then in **Konduri, Karthik C., et al 2016**[6], the paper extends the work of Ye, Xin, et al 2009 which introduced the Iterative Proportional Updating (IPU) algorithm. Due to the shortcoming that the IPU algorithm in **Ye, Xin, et al 2009**[4] have not been able to accommodate household and person-level attributes of interest at multiple geographical resolutions simultaneously, it may lead to a potential mismatch between the synthetic population and true population on known characteristics of interest. For example, if control distributions for a few variables of interest are available at the Traffic Analysis Zone (TAZ) level, and distributions of others are available only at the census tract level, the existing algorithm cannot be used to control variables of interest at both geographical resolutions simultaneously. The resulting synthetic population may not be as representative of the true population as it might have been had information available at both geographic resolutions been used. Inaccuracies in population representativeness will inevitably have adverse downstream impacts on forecasts obtained from activity-based microsimulation models that take the synthetic population as input. So, based on the issue of the previous IPU algorithm, this paper puts forward an enhanced IPU algorithm that can accommodate constraints at multiple spatial resolutions. This algorithm contains three steps. Firstly, the enhanced IPU algorithm begins by assigning an initial set of weights to all sample households in all geographic units. Unit weights are assigned to each sample household to start the sample weight estimation process. Separate marginal distributions are available at

the region level and at the level of two geographic units. Person-level marginal distributions are assumed to be available only at the level of geographic units. The IPF procedure is run for the region as a whole and for individual geographic units to obtain constraints that need to be matched at various spatial resolutions. Secondly, the sample weight for all geographic units in a Region are adjusted to match the marginal distributions at the Region level. This procedure consists of three sub-steps: (1) An adjustment factor for the first household type is computed as the Region level constraint divided by the corresponding weighted sum in all geographic units taken together; (2) Weight values for the sample households that correspond to the household type under consideration are multiplied by the adjustment factor; (3) All weights sum and deviation values are updated based on the new weights for all household and person types at both Region and Geo levels. The objective of the final step is to satisfy the household type and person type constraints at a finer spatial resolution by adjusting sample household weights within each geographic unit (Geo). The sample weighting process is applied separately to each geographic unit to realize this objective. First, an adjustment factor for the first household type in a geographic unit is computed as the corresponding constraint divided by the weighted sum. Second, weight values for the sample households that belong the first household type are adjusted by multiplying the current weight with the adjustment factor. This process is repeated for all household and person types in the geographic unit. Weighted sums and corresponding deviation values are updated based on the new weights for the geographic unit under consideration. This procedure is operated for all geographic units within a Region to complete one full iteration of the enhanced

algorithm. The weighted sum and deviation values at the Region level are also updated at the end of each adjustment. The enhanced IPU algorithm is terminated when the improvement in the average deviation value drops below a user-specified threshold (When the average deviation value approaches zero, it indicates that the sample weights are converging and weighted sums for all households and person types matching the geographic unit level constraints.) However, if there are inconsistencies in marginal distributions across the geographic levels, then the solution is likely to result in a perfect match for some constraints and only a close match for others, which requires the consistency of input data across geographic levels. The enhanced IPU algorithm helps advance the development of synthetic population generators that can control for attributes of interest at multiple spatial resolutions simultaneously and shows the better performance of controlling for variables at the resolution for which data is available than not controlling for them at all.

2.2 MCMC

Farooq, Bilal, et al 2013[7] points out the key shortcomings of the techniques that focus on treating synthesis as a fitting problem, which includes Iterative Proportional Fitting (IPF) and Combinatorial Optimization based techniques. The shortcomings are: 1) fitting of one contingency table, while there may be other solutions matching the available data; 2) due to cloning rather than true synthesis of the population, losing the heterogeneity that may not have been captured in the microdata; 3) overreliance on the accuracy of the data to determine the cloning weights; 4) poor scalability with respect to the increase in number of attributes of the synthesized agents. Then they

put forward a Markov Chain Monte Carlo (MCMC) simulation-based approach to solve these issues. MCMC methods are computer-based simulation techniques that can be used to simulate a dependent sequence of random draws from very complicated stochastic models/processes. These methods have the following advantages: 1) provide flexibility in terms of using various data sources at various spatial scale; 2) bring in prior knowledge in a systemic way; 3) wherever the data is not available; 4) implement assumptions in a coherent manner; 5) are computationally and memory-wise robust. In this paper, the problem they want to deal with is to synthesize independent populations by drawing agents from the joint distribution of the attributes in the real population instead of fitting a single optimization-based solution and they propose to use MCMC techniques to draw from the real population distribution to obtain a synthetic population, instead of using the conventional fitting procedures. MCMC techniques can overcome the shortcomings mentioned above while keeping at least the same quality of data as the fitting-based techniques require. The process of the MCMC techniques in this paper is as follows: Firstly, they use Gibbs sampling, which is a MCMC method, to generate the synthetic population. Secondly, they prepare conditionals. They use parametric models to construct the conditional distribution here because in practice not all conditionals can be counts by category for each attribute like in the straightforward case. The flexibility of using such parametric models is that the data from various sources can be combined to estimate the parameter values. Thirdly, they deal with incomplete conditionals. They put forward two methods to solve the issue. One is to assume the conditional independence on the incomplete conditional which is not available, and another is to

use the domain knowledge about the incomplete part of the conditional to construct full conditionals. Finally, they realize the synthetic population. Using the full and consistent conditionals, it eventually reaches a stationary state if the Gibbs sampler is run for an extended number of iterations. At that point any draw will be as if the draw was from the original joint distribution. Then using this mechanism, a synthetic population can be realized by simply drawing the number of individuals equaling the size of the required population. Through comparing the performance of MCMC using the real population from Swiss census with the standard IPF, the standard root means square error statistics indicated even the worst-case simulation-based synthesis outperformed the best case IPF synthesis.

2.3 Machine Learning

Recently, many methodologies have been put forward to deal with the problem and as machine learning becomes popular in recent years and many generative models from the machine learning area are also utilized to solve this problem. One is the Bayesian Network. **Sun et al 2015**[8] proposes a new alternative for population synthesis based on Bayesian networks. The Bayesian network encodes probabilistic relationships (causality or dependence) among a set of variables by using a graphical model. Essentially, a Bayesian network for a set of variables contains two parts: 1) the qualitative part is a network structure in the form of a directed acyclic graph (DAG), in which nodes are in one-to-one mapping with the random variables and links characterize the dependence among connected variables; 2) the quantitative part is a set of local probability distributions/tables for each node/variable, conditional on its parents. For the learning problem in Bayesian network analysis, there are two

types of learning problem given a set of observations: 1) learning only model parameter when network structure is known; 2) learning both model structure and model parameter. Most practical problems belong to the second type, in which expert knowledge is not available or not sufficient enough for us to build the network structure from scratch. So, it's necessary to make full use of observations available to learn network structure and model parameter simultaneously, which is often referred to as structural learning. It can be divided into two stages: model selection and model optimization. To proceed with model selection stage, a score-based approach is often applied and computing a score function that quantifies how well a hypothetical structure fits the data. Two main score functions are introduced. The most used score function is the Bayesian information criterion (BIC). It contains two terms: the first term is the optimal likelihood, which quantifies how well the hypothetical structure fits the data and the second term is a penalty function on the complexity of the model, preventing the overall structural learning process from overfitting. Another popular candidate score function is the so-called Akaike information criterion (AIC). It also includes two terms like BIC. The first term is still the optimal likelihood, but the second term is the number of free parameters in the model parameter, being independent from the size of observations. So, both BIC and AIC are constructed by adding penalty terms to the optimal likelihood that balances model fit and model complexity, but BIC penalize free parameters more strongly than AIC. There are also many other score functions can be chosen. After selecting a score function, the goal of the optimization stage is to identify the hypothetical structure with the highest score. In practice, it's infeasible to enumerate all potential candidates and evaluate

score of each of them, so the common approach is to use heuristic search techniques. In this paper, they mainly introduce the Tabu search method. It is an iterative searching procedure to move from one solution to its neighboring solution unit stopping criterion is satisfied. The performance of tabu search method is enhanced by using a memory structure (tabu list) while exploring the neighborhood of each solution during search processes compared with the local searching techniques. It is also able to escape from local optima, in which normal local search techniques often get stuck. After learning both model structure and model parameter, we can generate/sample values of the variables given the factorized joint probability distribution defined by the Bayesian network. Unlike the MCMC approach, samples generated from the Bayesian network are independent and thus the procedure can be paralleled and it's unnecessary to thin the results to reduce correlation between sequential samples. The Bayesian network model also provides us with an efficient approach to sample based on evidence.

The paper finally also applies the Bayesian network to generate synthetic population using the Household Interview Travel Survey (HITS) and the results shows the powerfulness of Bayesian network in characterizing the underlying joint distribution and meanwhile the overfitting of data can be avoided as much as possible.

There are also many other deep generative models in machine learning that are applied to generate synthetic populations. In **Borysov, Stanislav S., et al 2019**[9], a deep generative model called Variational Autoencoder (VAE) is introduced to generate micro-agents. The VAE is an unsupervised generative model which can learn the joint distribution of the observable variables. It is a latent variable model

which relates the observable variables to a multivariate latent variable. The intuition behind is that given some known joint distribution of the multivariate latent variable (e.g multivariate Gaussian), it can be mapped using a cleverly chosen nonlinear transformation such that it approximates the observable variables' joint distribution. VAE uses an encoder-decoder architecture, where the encoder maps the observable variables to a multivariable latent variable and the decoder maps the multivariable latent variable back. It is like the deterministic Autoencoder which usually has a bottle neck structure with the dimensionality of the multivariate latent variable much less than the dimensionality of the observable variables. This bottleneck structure allows for learning a compressed representation of sparse data in the low-dimensional latent space. VAE uses the reconstruction error and the Kullback-Leibler divergence to measure the performance after a VAE model is trained. The goal of training is of course to minimize the two types of errors. When the model has been trained, new samples can be generated by sampling the latent variable from the prior distribution and transforming it through the decoder to the data space. After introduce the VAE, the paper also compares the method with some other previous generative methods like Iterative Proportional Fitting (IPF), Gibbs sampling, Bayesian Networks or Hidden Markov in high-dimension cases. To compare the performance of these methods, standardized root mean squared error (SRMSE) based on marginal, bivariate and trivariate distributions are used as the measurement. The final comparison results turn out that although previous generative methods perform better in low-dimension datasets, VAE outperforms these methods when the dimension of datasets is much higher.

While this is not the end. Next year the article **Garrido, Sergio, et al 2020**[10] extends the previous work, which introduces another deep generative model called Generative Adversarial Network (GAN) to apply for a large-scale population synthesis. The idea is to train two neural networks at the same time. One network, which is the generator, is initiated with a draw from a latent variable. The draw is then transformed in such a way that the output is as realistic as possible. The second network is the discriminator, which receives an observation. This data can either be from real data or these from the generator. The objective of the discriminator is to tell whether the information it receives comes from the real data, and the objective of the generator, on the other hand, is to generate samples with the aim of "fooling" the discriminator. The name "adversarial" the refers to the competition between these two networks. In the process, feedback from the discriminator network is used in the generator to improve its capability of generating realistic agents. If the discriminator guessed that the sample generated by the generator was likely to be real, the generator doesn't move much away from that parameter configuration. The loss function used during the training of a GAN model is a minmax loss function, where the generator tries to minimize the loss function while the discriminator tries to maximize it. Just as mentioned above, it's a competition between the two networks. While instead of basic GAN architecture, the paper uses a more advanced GAN model called Wasserstein Generative Adversarial Network (WGAN), which is proven to have properties that makes it more desirable generation tasks. The losses of the generator and the discriminator in WGAN are based on a distance called the Wasserstein-1 distance. It is proved that if the generator is continuous in the parameters and locally Lipschitz,

the Wasserstein-1 distance is continuous everywhere and differentiable everywhere, which makes the Wasserstein-1 distance more desirable in an optimization procedure. The practical difference between these and the normal GAN comes in the loss function and from clipping the weights to force the generator to be Lipschitz. Then they make comparison with the previous generative methods including VAE as well. Besides SRMSE based on marginal, bivariate and trivariate distributions in their previous article, they add sampling zero to somewhat test the diversity of the data generated. The comparison results demonstrate that WGAN outperforms other methods in terms of prediction power but at the same time less diverse compared with VAE.

Chapter 3: Data Description

In this research study, a sample dataset of the total population was combined with the total aggregates (marginals) of selected variables to generate a synthetic population for the entire population. The sample dataset was obtained from the American Community Survey (ACS), and the marginals were collected from Decennial census data and the Internal Revenue income (IRS). The remaining of the chapter will mainly introduce the ACS data. PUMA and Decennial census data will also be introduced, and the dataset we use will also be included.

The ACS is a national household survey conducted every year by the U.S. It is on the leading edge of survey design, continuous improvement, and data quality and it is the nation’s most current, reliable, and accessible data source for local statistics on critical planning topics.

For the source of addresses for ACS, the Master Address File (MAF) is used, which is the Census Bureau's official inventory of known housing units (Hus), group quarters (GQs), and selected non-residential units (public, private, and commercial) in the United States and Puerto Rico. It contains mailing and location address information, geocodes, and other attribute information about each living quarter. A geocoded address is one for which state, county, census tract, and block have been identified. The MAF is linked to the Topologically Integrated Geographic Encoding and Referencing (TIGER) system. TIGER is a database containing a digital representation of all census-required map features and related attributes. It is a resource to produce maps, data tabulation, and the automated assignment of addresses to geographic locations in geocoding. The resulting database is called the MAF/TIGER database (MTdb). The MAF was used as the initial frame for the ACS, in its state of existence at the conclusion of Census 2000. Updates from nationwide 2010 Census operations were incorporated into the MTdb and were included in the ACS sampling frame in the middle of 2010. The Census Bureau continues to update the MAF using the DSF and various automated, clerical, and field operations, such as the Demographic Area Address Listing (DAAL).

For the sample selection, they will select independent HU address samples and independent full-implementation samples of GQ facilities and persons, which are the two separate samples included in the ACS that are drawn from the MAF. They select independent HU address samples for each of the 3,143 counties and county equivalents in the U.S., including the District of Columbia, as well as for each of the 78 municipalities in Puerto Rico. In 2004, they selected samples of HU addresses for

every county and county equivalent for field data collection in 2005.1 Each year from 2005–2010, they selected approximately 2.9 million HU addresses in the U.S. and 36,000 HU addresses in Puerto Rico. Beginning in 2011, they implemented the following changes to the ACS sample designs:

(1) They increased the HU sample in June 2011, bringing the size of the sample selected to 3.54 million addresses per year.

(2) They added several new HU sampling rates that better control the allocation of the sample and improve estimate reliability for small areas.

(3) They increased the follow-up sample to 100 percent in select geographic areas.

Full-implementation samples of GQ facilities and persons are selected independently within each state, including the District of Columbia and Puerto Rico. This began in 2006. In 2006 and 2007, the ACS and the PRCS included approximately 2.5 percent of the expected number of residents in GQ facilities. Beginning in 2008, they increased the sampling rates in 16 states with small GQ populations to meet publication thresholds.

For the survey rules, the ACS uses residence rules based on the concept of current residence. Residence rules are the series of rules that define who (if anyone) should be interviewed at a sample address, and who is considered, for purposes of the survey or census, to be a resident. Residence rules decide the occupancy status of each HU and the people whose characteristics are to be collected. ACS data are collected nearly every day of the year. The survey's residence rules are applied, and its reference periods are defined as of the date of the interview. For mail or Internet

responses, this is when the respondent completes the questionnaire; for telephone and personal visit interviews, it is when the interview is conducted.

For the content collected by the ACS, they can be grouped into four main types of characteristics: social, demographic, economic and housing. Social characteristics include topics such as education, marital status, fertility, veterans, disability status, place of birth and others. Basic demographic characteristics such as sex, age, race, Hispanic origin are also collected by the ACS, which are also the same information collected on the decennial census. Economic characteristics include topics such as employment status, income commuting to work, occupation, industry, health insurance and others. Housing characteristics include topics such as tenure, information about occupancy, and the structure itself which includes house value, housing cost, utilities, plumbing, kitchen facilities and others.

For the data collection operation for housing units HUs, it consists of four modes: Internet, mail, telephone, and personal visit. For most HUs, the first phase includes a mailed request to respond via Internet, followed later by an option to complete a paper questionnaire and return it by mail. If no response is received by mail or Internet, the Census Bureau follows up with computer assisted telephone interviewing (CATI) when a telephone number is available. If the Census Bureau is unable to reach an occupant using CATI, or if the household refuses to participate, the address may be selected for computer-assisted personal interviewing (CAPI). For the data collection operation for GQs, it is conducted in two phases. First, U.S. Census Bureau Field Representatives (FRs) conduct interviews with the GQ facility contact person or the administrator of the selected GQ (referred to as the GQ level interview), and

second, the FR conducts interviews with a sample of individuals from the facility (referred to as the person- or resident-level interview). The GQ-level data collection instrument is an automated Group Quarters Facility Questionnaire (GQFQ). Information collected by the FR using the GQFQ during the GQ-level interview is used to determine or verify the type of facility, population size, and to draw a random sample of residents to be interviewed. FRs conduct GQ-level data collection at approximately 20,000 individual GQ facilities each year.

For the data preparation and processing for HUs and GQs, the main purpose is to take the response data gathered from each survey collection mode to the point where they can be used to produce survey estimates. Data returning from the field typically arrive in various stages of completion, from a completed interview with no problems to one with most or all of the data items left blank. There can be inconsistencies within the interviews, such that one response contradicts another, or duplicate interviews may be returned from the same household but contain different answers to the same question.

Upon arrival at the U.S. Census Bureau, all data undergo data preparation, where responses from different modes are captured in electronic form creating Data Capture Files. The write-in entries from the Data Capture Files are then subject to monthly coding operations. When the monthly Data Capture Files are accumulated at year-end, a series of steps are taken to produce Edit Input Files. These are created by merging operational status information (such as whether the unit is vacant, occupied, or nonexistent) for each HU and GQ facility with the files that include the response data.

These combined data then undergo a number of processing steps before they are ready to be tabulated for use in data products.

For the weighting and estimation, the basic estimation approach is a ratio estimation procedure that results in the assignment of two sets of weights: a weight to each sample person record, both household and group quarters (GQ) persons, and a weight to each sample housing unit (HU) record. As with most household surveys, weights are used to bring the characteristics of the sample more into agreement with those of the full population by compensating for differences in sampling rates across areas, differences between the full sample and the interviewed sample, and differences between the sample and independent estimates of basic demographic characteristics. In particular, the ACS uses ratio estimation to take advantage of independent population estimates by sex, age, race, and Hispanic origin, and estimates of total HUs produced by the Population Estimates Program (PEP) of the Census Bureau. This results in an increase in the precision of the estimates and corrects for under-/over coverage by geography and demographic detail. This method also produces ACS estimates consistent with the population estimates by these characteristics and the estimates of total HUs for each county in the United States. For any given geographic area, a characteristic total is estimated by summing the weights assigned to the people, households, families, or HUs possessing the characteristic. Estimates of population characteristics are based on the person weight. Estimates of family, household, and HU characteristics are based on the HU weight.

For the variance estimation, all published ACS estimates are accompanied either by 90 percent margins of error or confidence intervals, both based on ACS direct

variance estimates. Due to the complexity of the sampling design and the weighting adjustments performed on the ACS sample, unbiased design-based variance estimators do not exist. As a consequence, the direct variance estimates are computed using a replication method that repeats the estimation procedures independently several times. The variance of the full sample is then estimated by using the variability across the resulting replicate estimates. Although the variance estimates calculated using this procedure are not completely unbiased, the current method produces variances that are accurate enough for analysis of the ACS data. For Public Use Microdata Sample (PUMS) data users, replicate weights are provided to approximate standard errors for the PUMS-tabulated estimates. Design factors are also provided with the PUMS data, so PUMS data users can compute standard errors of their statistics using either the replication method or the design factor method.

Finally for the ACS data products, they include the tables, reports, and files that contain estimates of demographic, social, economic and housing characteristics. These products cover geographic areas within the United States and Puerto Rico. The Public Use Microdata Sample (PUMS) files, which enable data users to create their own estimates, are also data products. More details about ACS data can be found on their official website.

The ACS provides data for Public Use Micro Areas (PUMAs) at two different levels: household level and individual level. The PUMA is a geographic unit defined by the U.S. Census and contains at least 100k people; PUMAs do not overlap and nested within a single state. The sample dataset used in this study included 9 variables from

both household level and individual level. Table 3 lists the selected variables and their names, levels, definitions and associated values.

Table 3: List of variables included in the dataset of this study

Name	Definition	Level	Values
AGEP	Age of person	Individual	0,99 - 0 to 99 years
SEX	Gender of person	Individual	1: Male, 2: Female
RAC1P	Race of person	Individual	1: White alone, 2: Black or African American alone, 3: American Indian alone, 4: Alaska Native alone
ESR	Employment status	Individual	1: Civilian employed, at work, 2: Civilian employed, with a job but not at work, 3: Unemployed, 4: Armed Forces, At Work, 5: Armed

			Forces, With a Job but Not at Work, 6: Not in Labor Force
HINCP	Household income (past 12 months)	Household	1: $1 to less $25k, 2: $25k to less $50k, 3: $50k to less $75k, 4: $75k to less $100k, 5: $100k 50 less $200k, 6: $200k or more
HHT	Household/family type	Household	1: Married couple household, 2: Male householder, no spouse present, 3: Female householder, no spouse present, 4: Male householder: Living alone, 5: Male householder:

			Not living alone, 6: Female householder: Living alone, 7: Female householder: Not living alone
NP	Number of persons in the household	Household	1,6: Number of persons in household
WIF	Workers in family during the past 12 months	Household	0: No workers, 1: 1 worker, 2: 2 workers 3: 3 or more workers in family
HUPAC	HH presence and age of children	Household	1: With children under 6 years only, 2: With children 6 to 17 years only, 3: With children under 6 years and

			6 to 17 years, 4: No children

The decennial Census Data is updated every ten years at years ending with zeros (i.e. 2000 and 2010). The obtained sample dataset was included the ACS sample data for five consecutive years (2012 to 2016) to ensure having a large enough sample to train the developed model. Moreover, the selected years were chosen as the middle ground between the most recent census data (2010) for the marginal dataset and the year of the study.

The developed model has been tested in generating synthetic populations for different regions in Maryland with different community types. In this study, the input dataset involved two counties in Maryland: Anne Arundel County and Frederick County as they have different community types; rural and urban, respectively. Anne Arundel County has four PUMAs (1201 to 1204), and Frederick County has two PUMAs (301 and 302). The sample size for Arundel County is 22,345 observations combined from the four PUMAs for the five years at the individual and household levels. Similarly, the sample dataset combined from the two PUMAs in Frederick County has 13,653 observations.

The real observations in the sample dataset were transformed into pseudo-observations by applying the empirical distribution function to the data. The pseudo-observations were used to train the model to learn the dependency between the selected variables to generate synthetic pseudo-observations for the whole population. The inverse of the cumulative distribution function (CDF). Then, the inverse

sampling from the cumulative distribution function (CDF) was applied to transform the synthetic pseudo-observations to the synthetic population. In this study, the CDFs were collected from two data sources. The CDF functions of the following variables: AGEP, SEX, RAC1P, HHT, NP, and HUPAC were obtained from the decennial census 2010 at the census tract level then was aggregated at the PUMA level. For the WIF and ESR, the CDF functions are not available at the census tract level in the decennial census data, therefore, it was constructed from the sample dataset at the PUMA level and assumed that all census tracts belonging to this PUMA have the same CDF functions. The CDF functions of HINCP were also not available in the census data, however, they were generated from the IRS data at the census tract level.

Chapter 4: Methodology

4.1 IPF

In the paper, we use the standard IPF procedures to generate synthetic population. The procedure of the IPF can be summarized as follows:

(1) Choose household-level control variables.

(2) Obtain the marginal distributions on these variables from census summary files (SF).

(3) Generate a seed matrix of the joint distribution from a microdata sample data set (PUMA, travel survey).

(4) Expand the seed matrix using an IPF-procedure to match the given marginal control totals while maintaining the joint distribution implied by the seed matrix.

There is also something needs to be mentioned during the procedure of IPF:

(1) Selection probabilities are estimated for households in the microdata sample.

(2) Households are drawn using the selection probabilities to match the expanded cell frequencies.

(3) The resulting synthetic population is checked for goodness-of-fit and households are redrawn if necessary.

(4) The synthetic population is comprised of all individuals within the synthesized (drawn) households.

Fig 4.1 gives a simple example of how IPF progress:

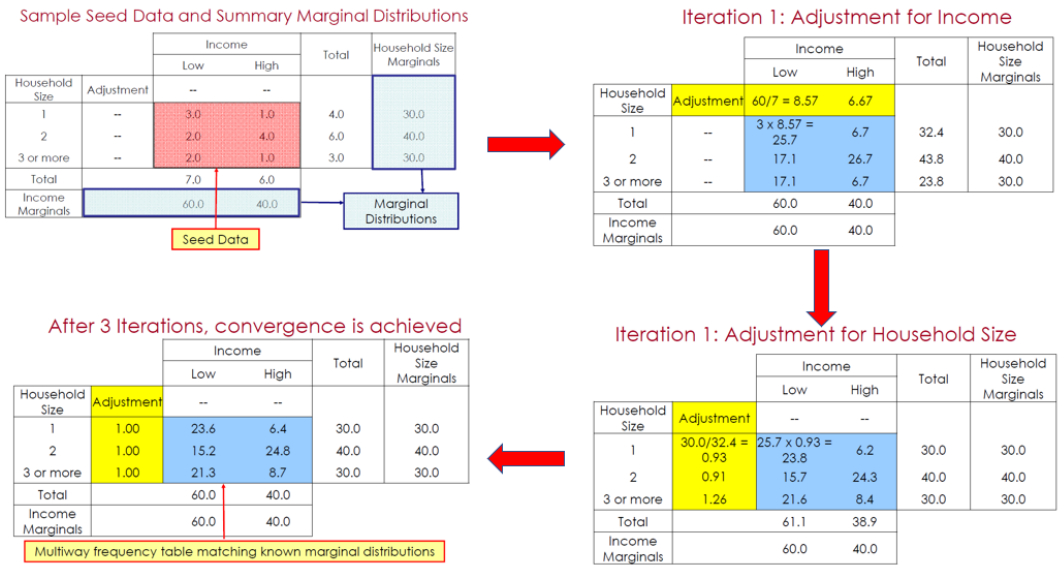

Fig 4.1 IPF Procedure

In the paper, we will use the same procedure, but apply it to the ACS data to generate the synthesis population we need.

4.2 Bayesian Network

The definition of Bayesian network and how it works in the context of population synthesis comes from **Sun et al 2015**[8]. It is a graphical model that encodes probability distributions for a set of variables $X = \{X_1 \dots X_n\}$ of interests and the variables X consists of qualitative and quantitative parts:

(1) For the qualitative part, it is a network structure G in the form of a directed acyclic graph (DAG) where nodes are in one-to-one mapping with the random variable X and links characterize the dependence among connected variables.

(2) For the quantitative part, it is a set of local probability distributions/tables $\Theta = \{P(\Pi_1), \dots, P(\Pi_n)\}$ for each node /variable X_i, conditional on its parents Π_i.

Here G is referred to as model structure and Θ is referred to as model parameters. X_j is referred to as a parent of X_i if there exists a direct link from X_j to X_i and Π_i is used to denote the set of variables X_j. No certain parent variables Π_i will make the local probability distribution collapse to its marginal $P(X)$. The DAG topology of Bayesian Network only asserts conditional dependence of children given parents. Then the joint distribution for X in a Bayesian Network can be decomposed, by integrating G and Θ and using the chain rule, into a factorized form with smaller and local probability distributions, each of which consists of one node and its parents only:

$$P(X) = \prod_{i=1}^{n} P(X_i|\Pi_i)$$

In other words, the joint probability distribution $P(X)$ can be exclusively encoded by the pair (G, Θ). The Bayesian network representation allows us to approximate and represent an unknown distribution $P(X)$ into a concise graphical form ($P(X) \cong$

$P(X)$). Bayesian network system can thus offer us an intuitive framework to reproduce $P(X)$ of the studied population system in terms of population synthesis.

The way to construct full conditions in Bayesian network model involves the concept of Markov blanket. Markov blanket $Mb(X_i)$ for a particular node X_i is the union of three sets: (1) its parents; (2) its children; (2) the coparents-a set consists of other parents of its children (excluding X_i). Since we can derive $P(X_i|X_{i-1}) = P(X_i, X_{-i})/P(X_{-i})$ by canceling out all terms that do not involve X_i from both numerator and dominator, we have:

$$P(X_i|X_{i-1}) \propto P(X_i|\Pi_i) \prod_{k \in ch(j)} P(X_k|\Pi_k)$$

Where $ch(j)$ denotes the children nodes of X_i.

Based on this expression, we can find that the full condition distribution $P(X_i|X_{i-1})$ only depends on its Markov blanket $Mb(X_i)$, which demonstrates that the inference and sampling of X_i can be achieved by only looking at its Markov blanket $Mb(X_i)$ instead of the full conditionals. It also gives us a warning to the use of partial/incomplete conditionals to replace the full conditionals in the MCMC approach.

For the learning problem in Bayesian network analysis, we encounter the type in which expert knowledge is not available or not sufficient enough for us to build the network structure from scratch in this research, which means we should make full use of available observations to learn G and Θ simultaneously. It is often referred to as structure learning that in general can be divided into two stages: model selection and model optimization. For model selection we usually apply a score-based approach

and the most used score function is the Bayesian information criterion (BIC), which is defined by (Schwarz,1978):

$$BIC(G^h|D) = log\ P\left(D|G^h, \hat{\theta}\right) - \frac{d}{2} log\ m$$

Where $\hat{\theta}$ is the maximum likelihood estimates of parameters given a hypothetical structure G^h, d is the number of free parameters (degrees of freedom) in Θ, and m is the size of observation D. The first term on the right-hand side is the optimal likelihood, which quantifies how well the hypothetical structure G^h fits the data; the second term is a penalty function on the complexity of the model, preventing the overall structural learning process from overfitting. After selecting a score function, the goal of the optimization stage is to identify the hypothetical structure with the highest score and tabu search method, which is a heuristic search technique, is used to deal with the situation where too many potential candidates exist. It is an iterative searching procedure to move from one solution to its neighboring solution until some stopping criterion is satisfied. It is used because its performance is enhanced by using a memory structure (tabu list) while exploring the neighborhood of each solution during the search process. It is also capable of escaping from local optima.

The Bayesian network method does not require marginals as input and any conditionals as well since structural learning and parameter estimation are integrated in the learning of a Bayesian network model. Therefore, the only input that is required in learning a Bayesian network model is a specified score function.

When we want to realize the population synthesis, we can generate/sample values of X given the factorized joint probability distribution $P(X)$ defined by the Bayesian network after learning both model structure and parameter. Samples generated from

the Bayesian network are independent, so we don't need to thin the results to reduce the correlation between sequential samples. The Bayesian network is also able to sample based on evidence efficiently, like using marginal distributions as evidence to control the global sampling of X. The quality and quantity of observation D may determine the functionality of estimated network substantially since the learning of the Bayesian network relies on D. So keeping the structure of a Bayesian network as simple as possible is necessary to reduce the occurrence of undefined local conditional distribution during the process of learning of the Bayesian network. Adopting an appropriate score function, reducing the number of categories in each variable, and adopting a Bayesian framework specifying prior distributions of potential parameters are the strategies to achieve the goal. Then by sampling from the obtained Bayesian network we are allowed to generate a large list of individuals as a population pool.

In this research, we apply a score function called the minimum description length (MDL) consisting of two components that estimate the structural complexity and the likelihood of the data given the model. The best model is chosen when it has the shortest description of the data. For the dependencies among nine variables in the dataset, structural learning is used to learn the dependencies just as mentioned before. And the implementation of Bayesian Network is included in a package pomegranate for python. The greedy algorithm is used to learn the network and the rejection method is used for sampling.

4.3 CTGAN

The paper will use the CTGAN model instead of the conventional GAN model as one of the machine learning methods used to generate synthetic population, which was put forward in **Xu, Lei, et al. 2019**[11]. It is an advanced GAN model that models tabular data distribution and sample rows from the distribution. There are many issues of tabular data challenge the conventional GAN model, including mixed data types, non-Gaussian distributions, multimodal distributions, learning from sparse one-hot-encoded vectors and highly imbalanced categorical columns. To solve these issues, specific techniques are put forward in CTGAN model, which are mode specific normalization and conditional generator. The mode specific normalization is designed to solve the issues of non-Gaussian and multimodal distribution and conditional generator is used to deal with imbalanced discrete columns.

For the mode-specific normalization, it contains three steps:

(1) For each continuous column C_i, use variational Gaussian mixture model to estimate the number of modes mi and fit a Gaussian mixture.

(2) For each value $c_{i,j}$ in C_i, compute the probability of $c_{i,j}$ coming from each mode.

(3) Sample one mode from given the probability density and use the sampled mode to normalize the value.

The representation of a row become the concatenation of continuous and discrete columns:

$$r_j = \alpha_{1,j} \oplus \beta_{1,j} \oplus \ldots \oplus \alpha_{N_c,j} \oplus \beta_{N_c,j} \oplus d_{1,j} \oplus \ldots \oplus d_{N_c,j}$$

Where $d_{i,j}$ is one-hot representation of a discrete value.

For conditional generator, it can be interpreted as the conditional distribution of rows given the particular value at the particular column, which can be shown as: $\hat{r} \sim P_G(row|D_{i*} = k^*)$. Here k^* is the value from the i^*th discrete column D_{i*}, $\hat{r}$ are the generated samples. While integrating conditional generator into a GAN architecture needs to devise a representation for the condition as well as to prepare an input for it, to preserve the condition as it is given for generated rows and to learn the real data conditional distribution for the conditional generator. The solution here includes three key elements, which are the conditional vector, the generator loss and the training-by-sampling method:

(1) Conditional vector. It is introduced to indicate the Condition $D_{i*} = k^*$. Let the ith mask vector be $m_i = [m_i^{(k)}]$, where:

$$m_i^{(k)} = \begin{cases} 1 \; if \; i = i^* \; and \; k = k^* \\ 0 \; other \; wise \end{cases}$$

Then the conditional vector can be figured out use:

$$cond = m_1 \oplus \ldots \oplus m_{N_c}$$

(2) Generator loss. To enforce the conditional generator to produce $\hat{d}_{i*} = m_{i*}$ instead of being free to produce any set of discrete vectors, the cross-entropy between $\hat{d}_{i*}$ and m_{i*} is added and the losses are averages over all the instances of the batch.

(3) Training-by-sampling. Following steps are proposed to assess the output produced by the conditional generator:

1) Create N_d zero-filled mask vectors $m_i = [m_i^{(k)}]_{k=1..|D_i|}$ for $i = 1, \ldots, N_d$.

2) Randomly select a discrete column D_i out of all the N_d discrete columns, with equal probability.

3) Construct a PMF across the range of values of the column selected in 2), D_{i^*}, such that the probability mass of each value is the logarithm of its frequency in that column.

4) Let k^* be a randomly selected value according to the PMF above.

5) Set the k^*th component of the i^*th mask to one: $[m_{i^*}^{(k^*)}] = 1$.

6) Calculate the conditional vector: $cond = m_1 \oplus ... \oplus m_{N_c}$.

Finally, the conditional generator can be formally described as:

$$\begin{cases} h_0 = z \oplus cond \\ h_1 = h_0 \oplus ReLU(BN(FC_{|cond|+|z|\to 256}(h_0)) \\ h_2 = h_1 \oplus ReLU(BN(FC_{|cond|+|z|+256\to 256}(h_1)) \\ \hat{\alpha}_i = \tanh\left(FC_{|cond|+|z|+512\to 1}(h_2)\right)\ 1 \le i \le N_c \\ \hat{\alpha}_i = \text{gumbel}_{0.2}\ (FC_{|cond|+|z|+512\to m_i}(h_2))\ 1 \le i \le N_c \\ \hat{d}_i = \text{gumbel}_{0.2}\ (FC_{|cond|+|z|+512\to |D_i|}(h_2))\ 1 \le i \le N_d \end{cases}$$

Here $gumbel_\tau(x)$ refers to Gumbel softmax with parameter τ on a vector x, $FC_{u\to v}(x)$ refers to a linear transformation on a u-dim input to get a v-dim output.

The PacGAN framework with 10 samples in each pac is used to prevent mode collapse. The architecture of the critic can be formally described as:

$$\begin{cases} h_0 = r_1 \oplus ... \oplus r_{10} \oplus cond_1 \oplus ... \oplus cond_{10} \\ h_1 = drop(\text{leaky}_{0.2}\left(FC_{10|r|+10|cond|\to 256}(h_0)\right)) \\ h_2 = drop(\text{leaky}_{0.2}(FC_{256\to 256}(h_1))) \\ C(\cdot) = FC_{256\to 1}(h_2) \end{cases}$$

Here $leaky_\gamma(x)$ refers to a leaky ReLU activation on x with leaky ratio γ,

The model is trained using WGAN loss with gradient penalty and Adam optimizer with learning rate $2 \cdot 10^{-4}$ is used.

The model of CTGAN can also be visualized as Fig 4.3:

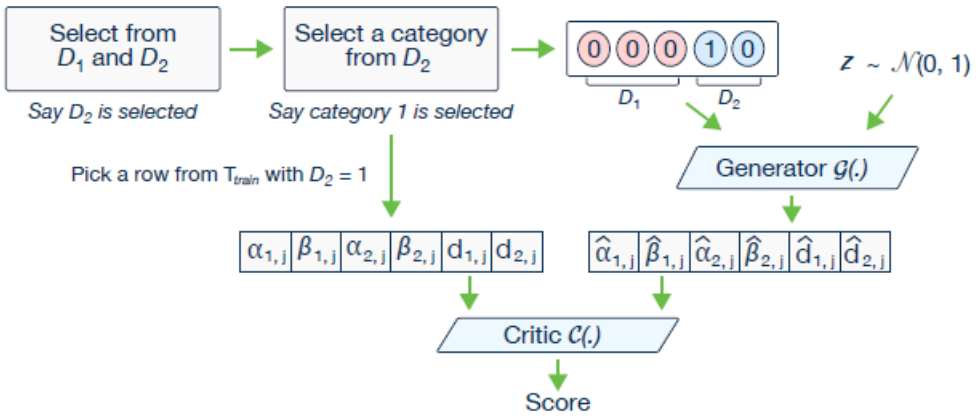

Fig 4.3: CTGAN model

The network architectures and parameters shown above are the default architectures and values which have already been defined in the package ctgan in python. In the paper, We directly use the default ctgan to train the dataset and generate the synthesis population.

4.4 TVAE

Just like GAN, in the paper TVAE model is used instead of conventional VAE to generate synthesis population, which adapts VAE to tabular data by using the same preprocessing and modifying the loss function. The TVAE model is also proposed in **Xu, Lei, et al. 2019** and they use two neural networks to model $p_\theta(r_j|z_j)$ and $q_\emptyset(z_j|r_j)$ and train them using evidence of lower-bound (ELBO) loss. Here $q_\emptyset(z_j|r_j)$ can be seen as the encoder part that maps observable variable to latent space and $p_\theta(r_j|z_j)$can be seen as the decoder part that maps the latent space back.

The design of $P_\theta(r_j|z_j)$ is shown as follow:

$$\begin{cases} h_1 = ReLU(FC_{128\to128}(z_j)) \\ h_2 = ReLU(FC_{128\to128}(h_1)) \\ \bar{a}_{i,j} = tanh(FC_{128\to1}(h_2)\ 1 \le i \le N_c \\ \hat{a}_{i,j} \sim N(\bar{a}_{i,j}, \delta_i)\ 1 \le i \le N_c \\ \hat{\beta}_{i,j} \sim softmax(FC_{128\to m_i}(h_2))\ 1 \le i \le N_c \\ \hat{d}_{i,j} \sim softmax(FC_{128\to|D_i|}(h_2))\ 1 \le i \le N_d \\ p_\theta(r_j|z_j) = \prod_{i=1}^{N_c} P(\hat{a}_{i,j} = a_{i,j}) \prod_{i=1}^{N_c} P(\hat{\beta}_{i,j} = \beta_{i,j}) \prod_{i=1}^{N_d} P(\hat{a}_{i,j} = a_{i,j}) \end{cases}$$

Here $\hat{a}_{i,j}$, $\hat{\beta}_{i,j}$ and $\hat{d}_{i,j}$ are random variables. They assume $\hat{a}_{i,j}$ follows a Gaussian distribution with different mean and variance and $\hat{\beta}_{i,j}$ and $\hat{d}_{i,j}$ follow a categorical PMF. Weight matrices and variance are parameters in $P_\theta(r_j|z_j)$, which are trained using gradient descent.

The design of $q_\emptyset(z_j|r_j)$ is similar to conventional VAE, which is shown as follow:

$$\begin{cases} h_1 = ReLU(FC_{|r_j|\to128}(r_j)) \\ h_2 = ReLU(FC_{128\to128}(h_1)) \\ \mu = FC_{128\to128}(h_2) \\ \sigma = \exp\left(\frac{1}{2} FC_{128\to128}(h_2)\right) \\ q_\emptyset(z_j|r_j) \sim N(\mu, \sigma I) \end{cases}$$

TVAE uses 1e-3 as the training rate of Adam.

In the paper, we also directly use the default tvan to train the dataset and generate the synthesis population and the network architectures and parameters shown above are the default architectures and values defined in the package tvae in python.

4.5 Copula

A copula $C: [0,1]^d \to [0,1]$ is a multivariate cumulative distribution function (CDF) for which the marginal probability distribution of each variable is uniform on the interval [0,1].

Consider a random vector $X = (X_1, \dots, X_d)$. If its marginals $F_i(x) = P[X_i \leq x]$ are continuous, then applying the probability integral transform to each component gives the random vector:

$$U = (U_1, \dots, U_d) = (F_1(X_1), \dots, F_d(X_d))$$

which has uniform marginals over $[0,1]$. This normalized vector is referred as the copula-uniform dual representation of X. It allows us to study the structure of our problem in a way that is robust to the peculiarities of the marginal distributions. The copula of X characterizes the dependency structure of its copula-uniform dual representation. In other words, it is the joint CDF of U

$$C(u_1, \dots, u_d) = P[U_1 \leq u_1, \dots, U_d \leq u_d]$$

Given a procedure to generate samples from the copula function, we can map back the synthetic samples to the data space by applying the inverse of the marginal CDFs

$$(X_1, \dots, X_d) = (F^{-1}(U_1), \dots, {F_d}^{-1}(U_d))$$

Copulas fully capture the dependence structure between the input variables, while the marginals are informative about how the underlying phenomenon was observed. For example, in a census data generation context, it is reasonable to conjecture that marginals encode demographic elements and that multidimensional dependences encode more complex socioeconomic patterns shared across related regions.

Suppose we wish to generate synthetic data of a target population from which we only have marginals' information and that we have access to a sample of another source, population sharing the structure of the target. Our population synthesis procedure is as follow:

(1) Normalize the source population with the probability integral transform.

(2) Generate synthetic population from the normalized data.

(3) Decode the synthetic population with the inverse of the marginal CDFs of the

target population.

In practice, it is unlikely to have access to true marginals. To normalize X, one would then use the empirical CDFs (ECDF):

$$\hat{F}(x) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}_{x_i \leq x}$$

Where $\mathbb{I}$ is the indicator function and the pseudo-inverse function:

$$\hat{F}^{-1}(u) = min\ (x: F(x) \geq u)$$

to map the generated data back in the primal space.

The copula framework has been applied to most of my methods in order to improve the performance of the synthetic population generated by these methodologies. We also compare the methods plus copula with the ones without checking whether the copula framework does improve the performance of the methods we research. We define the code of copula transfer learning by myself. The class of copula can encode features with probability integral transform and decode features with pseudo-inverse of the provided data. It can also be used to resample discrete encoded features uniformly between the steps of their ECDF.

4.6 Discrete Case

When the variables of X have discrete supports, the normalized vector U does not have uniform marginals. In that case, the decoding step will not allow us to recover the marginals of the target population. We can see from the definitions that it is possible to lose values in some dimensions when applying the ECDF and the pseudo-inverse of two different populations to the same discrete data. Consider the example of a binary attribute. If the first possible value has a smaller ratio in the target distribution than in the source, then applying the pseudo-inverse of the target

population on the synthesized normalized examples will map every generated first value to the second, loosing the first value in the synthetic population.

We heuristically solve this issue with the following idea. Consider a generated example u. A discrete component u_i of u reside in the copula dual space and its domain is given by the ECDF of the source $u_i \in \{u_i^1, \ldots, u_i^{n_i}\}$ with $u_i^k \in [0,1]$ and $u_i^{k-1} \leq u_i^k$ for all k. Suppose that $u_i = u_i^k$, to recover the uniform behavior of this marginal, we sample its value between its current value and the largest possible value that is lower than it $u_i \sim U\{u_i^{k-1}, u_i^k\}$. We do this for each discrete dimension of every generated normalized sample.

In short, we add an intermediate uniform sampling step to our population synthesis procedure:

(1) Normalize the source population with the probability integral transform.

(2) Generate synthetic population from the normalized data.

(3) For each generated example, sample uniformly each discrete variable between its current value and the largest value lower than it.

(4) Decode the synthetic population with the inverse of the marginal CDFs of the target population.

Chapter 5: Results

This chapter will present all the results in my case study at the state level, county level and puma level. We compare all methodologies, including target, independent baseline, IPF, machine learning methods and Bayesian Network, using metric evaluations and marginal fit. We'll first focus on introducing metric evaluations we use and then present the results at the state level, county level and puma level.

5.1 Evalutation Methods

To assess the quality and fitness of the synthetic data several metrics are used and their introductions are shown as below:

(1) Standardized root mean squared error (SRMSE). The standardized root mean squared error (SRMSE) is a popular metric in the population synthesis scientific community. We follow the definition from Sun and Erath [2015], which is

$$SRMSE = \sqrt{\sum_{m_1=1}^{M_1} \dots \sum_{m_d=1}^{M_d} (f_{m_1 \dots m_d} - \hat{f}_{m_1 \dots m_d})^2 \times (M_1 \dots M_d)}$$

Where $f_{m_1 \dots m_d}$ and $\hat{f}_{m_1 \dots m_d}$ are the relative frequency of a particular combination in the reference data and in the synthetic data and M_i is the number of categories for attribute X_i. A value of 0 means a perfect match while large values mean distance between true and synthetic data.

As suggested by Borysov et al. [2018] we compute the average SRMSE for all possible i-uples of variables range from 1 to the number of variables in the dataset. For example, SRMSE averaged over the uni-variate distributions evaluate the fitting

of the marginals and SRMSE averaged over the possible pairs, triplets, and quadruplets of variables evaluate the fitting of the joint bi, tri, and quadri distributions. SRMSE captures whether the synthetic combination appears in the real data thus assessing the fitting of the multi-dimensional dependencies.

(2) Sampling zeros. SRMSE does not take into account the diversity of data. A synthetic combination might be desired even if it does not appear in the reference data. To assess the diversity of the produced synthetic data, we implement sampling zeros: we count the combinations of variables which are in the test set but not in the training set.

5.2 State Level

The experiment at the state level has several details need to be mentioned here:

(1) We assume the aggregation of Maryland's PUMAs to be a full population. Doing so allows me to accurately evaluate the multi-variate dependencies as if we knew the ground truth population.

(2) In this experiment, the aggregation of the PUMAs is the target and we use a random micro sample of 1% as the source.

(3) We use the empirical marginals of the target to transfer the learning of the dependencies from the source for IPF and our copula framework

Then the results of evaluation metrics are shown as Table 5.2:

	1	2	3	4	5	6	7	8	9	10	SZ
Ind	0.1042274043	0.4518935993	0.7102783809	1.604762656	4.507975642	10.93504464	12.27675129	24.34981605	47.24383404	122.1489898	7018
CTGAN	0.476184927	0.7940800443	1.243964983	2.882589763	7.186395281	14.48834985	16.27199799	29.89048243	54.84665431	133.0780349	26483
CTGAN+Copula	0.4609034478	0.7537656087	1.132869228	2.36523742	5.988960757	12.35977219	14.104627	26.56519116	50.26095556	123.2135545	26136
TVAE	1.269049356	1.723485698	2.700154978	6.708494713	14.53298521	29.84451156	31.53614861	51.10719453	88.93562612	203.6859545	18500
TVAE+Copula	1.171565823	1.545851852	2.393809697	5.260580512	11.32089855	23.05547586	24.49852065	40.28289263	70.69660189	162.7348373	22793
BN	0.09647206347	0.4472713068	0.6217259876	1.421844283	4.246599087	9.417012769	10.94415921	22.13597781	43.58435851	113.9449343	**37992**
BN+Copula	0.00946935709 4	0.4280410949	0.6106936121	**1.367383669**	**4.164179524**	**9.326245447**	**10.87232969**	**22.04757031**	**43.40697815**	**113.7785025**	34956
IPF	**0.0006386592122**	**0.2274147576**	**0.5174202018**	5.236730654	15.7470527	38.4385957905	54.7010404858	122.7033419	245.6906063	650.3500978	0

Table 5.2 Metrics Evaluation for State Level

Based on the results in Table 5.2, we have the following conclusions:

(1) IPF nicely matches the low dimensional conditional distributions but does worse than the independent baseline in higher dimensions.

(2) After applying copula transfer learning to three machine learning methods including Bayesian Network, CTGAN and TVAE, we can find that the performance of all these methods are in general improved compared with the ones without copula transfer learning, which are lower SRMSE in all dimensions in the three machine learning methods and higher sampling zeros in CTGAN and TVAE that hint more diverse synthetic data. For Bayesian Network, sampling zeros perform worse after applying copula framework. Importantly, IPF evidently has a nul score for sampling

zeros since it only replicates examples from the source. It cannot generalize to realistic but unknown combinations.

(3) Among the three machine learning methods, both Bayesian Networks perform best compared with CTGAN and TVAE in generating accurate synthetic populations since both Bayesian Networks have lowest SRMSE in all dimensions and highest sampling zero reflecting more diversity.

(4) Among all the methods, Bayesian Network plus Copula performs best in all dimensions of SRMSE and Bayesian Network performs best in sampling zero.

Then we compare the Bayesian Network and the one plus Copula (The two methods that perform best) with target and the independent baseline using the marginal fit among all the counties in Maryland is shown as Figure 5.1:

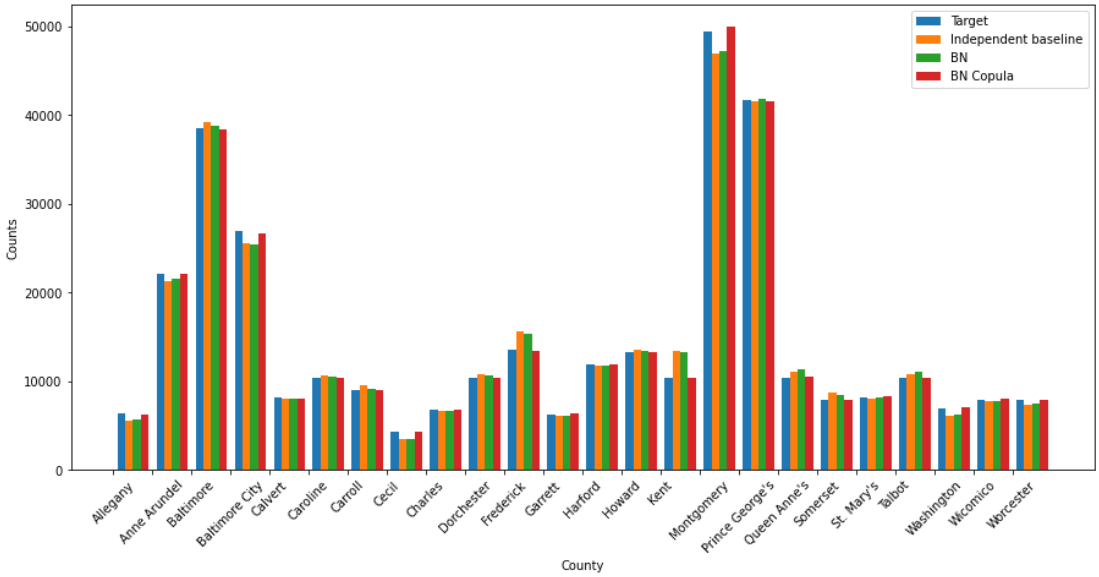

Figure 5.2 Marginal Fit at the State Level

We can find the similar conclusion that Bayesian Network plus Copula performs best in fitting the target data among all counties in Maryland.

5.3 County Level

The experiment at the county level also has several details need to be mentioned here:

(1) This experiment is similar to the previous, but we study the population synthesis capabilities at the county level. The source has a limited number of samples.

(2) We assume Anne Arundel county to be a full population.

(3) In that experiment, the county is the target and we use a random micro sample of 1% as the source.

Then the results of evaluation metrics are shown as Table 5.3:

	1	2	3	4	5	6	7	8	9	10	SZ
Ind	0.1272215183	0.3632546723	1.403442519	3.735783888	9.108546462	10.14627092	20.02362682	38.6781245	99.9818079	162.7823915	361
CTGAN	0.3011687643	0.5965593546	1.481913223	3.940497752	9.522843088	10.52974393	20.4308687	39.04500791	100.6719695	163.4089641	264
CTGAN+Copula	0.1779922895	0.5155970233	1.238247905	3.403892431	9.143073166	10.17011135	20.08846587	38.88744418	100.4440273	163.1311098	287
TVAE	0.1917865886	0.4531648119	4.057150911	10.88775758	23.97737917	27.06705786	49.5445576	90.41172616	220.7243117	348.9282139	1017
TVAE+Copula	0.1244953957	0.336316169	2.612728314	6.963648541	15.43533788	17.51597086	32.14573163	59.2709662	144.7920403	233.5453045	**1451**
BN	0.13484659	0.3727049798	1.397760086	3.696429966	8.943928153	10.01455985	19.87294334	37.81889478	98.01863583	161.6106233	756
BN+Copula	0.02143807508	**0.328127674**	**1.116433787**	**3.093551179**	**8.477610613**	**9.583896734**	**19.61445094**	**37.69558301**	**97.99671018**	**161.3770241**	684
IPF	**0.0005029778**	0.3647612808	5.518864114	16.5158186	40.6184343584	57.8823939614	130.0141199	260.4224076	689.8589054	1381.920488	0

Table 5.3 Metrics Evaluation for County Level

Based on the results in Table 5.3, the following conclusions can be arrived at similar to the previous experiment:

(1) Lowering the source size harms IPF's performance. IPF just nicely matches the first two low dimensional conditional distributions but does worse than the independent baseline in higher dimensions compared with the previous experiment that matches the first five dimensions.

(2) After applying copula transfer learning to three machine learning methods including Bayesian Network, CTGAN and TVAE, we can find that the performance of all these methods are perfectly improved compared with the ones without copula transfer learning, which are lower SRMSE in all dimensions and higher sampling zeros that hint more diverse synthetic data. Still, IPF evidently has a nul score for sampling zeros since it only replicates examples from the source. It cannot generalize to realistic but unknown combinations.

(3) Among the three machine learning methods, both Bayesian Networks perform best compared with CTGAN and TVAE in generating accurate synthetic populations since the two Bayesian Networks have lowest SRMSE in all dimensions. However, TVAE performs better in data diversity in this case since both TVAE architectures have higher values in sampling zero

(4) Among all the methods, Bayesian Network plus Copula performs best in all evaluation metrics in this experiment.

Then we also compare the Bayesian Network and the one plus Copula (The two methods that perform best) with target and the independent baseline using the marginal fit among all the pumas in Anne Arundel county is shown as Figure 5.2:

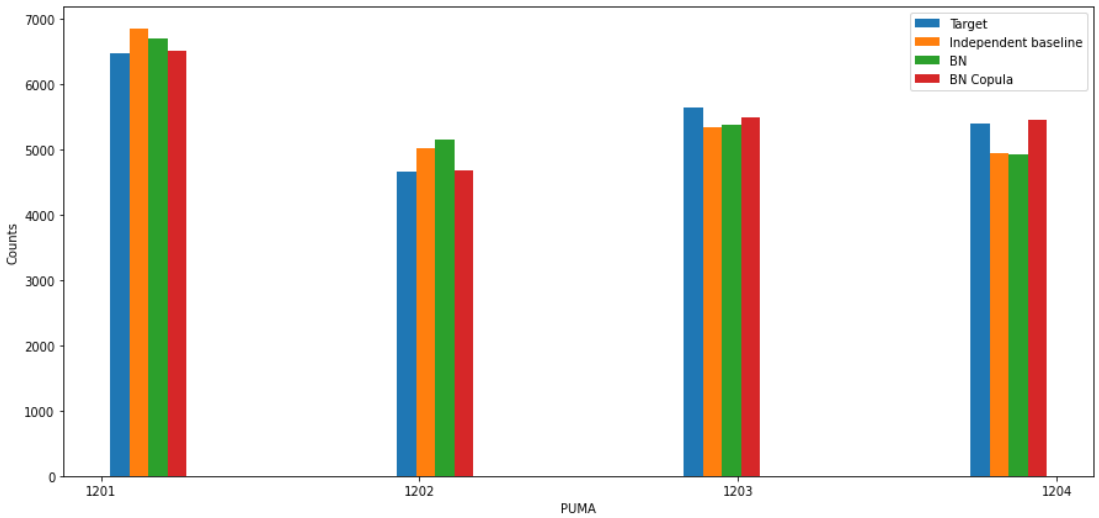

Figure 5.3 Marginal Fit at the County Level

We can still find the similar conclusion that Bayesian Network plus Copula performs best in fitting the target data among all the pumas in Anne Arundel county.

5.4 Puma Level

Table 5.4.1-5.4.6 show the results of the experiments at puma level at PUMA 1201, 1202, 1203, 1204, 301 and 302. Here we also add the parametric copula (These ML methods plus copula are non-parametric copula) as one of the methods to compare.

	1	2	3	4	5	6	7	8	9	SZ
Ind	0.2410529572	0.42791267	2.128125369	4.926008029	11.13564557	13.426192	27.6463497	53.717974637	140.4108854	70
CTGAN	0.2850136781	0.4897244658	2.375732739	5.307673053	11.38876522	13.688455603	27.775938 92	53.915685 92	140.2959068	86
CTGA	0.20880	0.39728	1.47765	3.83247	10.2884	12.7243	27.0579	53.3436	139.710	103

N+Copula	29794	82928	2401	9886	5	1907	7595	2878	71	
TVAE	0.2468967689	0.9030977911	5.729520227	14.595855094	29.752942667	38.493090887	75.900962228	121.3859201	305.7028763	176
TVAE+Copula	0.3300144101	0.6793281386	2.737845044	7.129245478	15.12846944	20.4965937	38.898899912	65.7368826	169.2792456	**372**
BN	0.2169675327	0.4328370393	2.135447544	4.920154542	11.021999722	13.339399738	27.609444188	53.622155286	140.0568036	93
BN+Copula	**0.02170237124**	**0.2760276786**	**1.42508711**	**3.753386914**	**10.160009951**	**12.615846634**	**27.003855174**	**53.230477853**	**139.3548492**	105
Copula	0.4556811881	0.7107491015	1.755604361	4.280399996	10.25521612	12.695711644	27.686411108	53.948556735	140.2959068	70
IPF	0.283277304	0.7917309614	9.612514814	27.385055826	67.196122648	95.455611482	213.86268	428.2505719	1133.08842	0

Table 5.4.1 Metrics Evaluation for PUMA 1201

	1	2	3	4	5	6	7	8	9	SZ
Ind	0.5311604863	0.7720259509	2.494084666	6.250468318	12.79852274	15.601399274	31.729677754	61.887488094	161.2249072	52
CTGAN	0.5506049664	0.9818642858	2.917420171	6.811316185	13.15626402	16.037553043	31.921122242	62.003854735	161.2547112	56
CTGAN+Copula	0.1996624166	0.5352189951	1.653666582	4.424276634	11.37651122	14.345112116	30.913211985	61.341511896	160.6276652	47
TVAE	0.5693794676	1.03358 6738	6.481842707	17.440833735	36.534299058	45.894366658	86.175355047	150.7184593	364.9875842	93
TVAE+Copula	0.06983565976	0.3001359902	2.414465998	6.717864143	14.732322252	19.538006488	38.165566801	71.000499703	176.111666	**168**
BN	0.5032805483	0.7720536383	2.564276635	6.39098651	13.069888094	15.868622468	32.073999067	62.086833268	161.9238516	37
BN+Copula	**0.04779288205**	**0.4242594786**	**1.546979017**	**4.182319769**	**10.91139999**	**14.012242239**	**30.771319962**	**61.207077163**	**160.6575801**	48
Copula	0.5094062634	0.7395758112	1.803727153	4.804583786	11.38555935	14.5010057	31.449877611	61.9706221	161.2845097	36
IPF	0.5265514114	1.065182688	11.806899289	33.509866951	82.194001502	116.398219	260.5471223	521.0406606	1378.564595	0

Table 5.4.2 Metrics Evaluation for PUMA 1202

	1	2	3	4	5	6	7	8	9	SZ
Ind	0.3413090432	0.5757610715	2.401156782	6.328680042	13.31171167	15.59693433	31.16607433	60.11710347	156.9901802	126
CTGAN	0.5106357559	0.8194517898	2.610307135	6.575829393	13.74839027	16.09942693	31.55445161	60.40085047	157.2823329	77
CTGAN+Copula	0.1638302906	0.5068090192	1.727822806	4.664691656	11.8729658	14.31836894	30.41680771	59.42348919	156.2888838	84
TVAE	0.4425471456	0.9505566282	6.50836408	18.30945271	36.84378899	46.1062643	88.05858841	152.9425267	392.471273	182
TVAE+Copula	0.1775984163	0.4898756451	2.585230474	7.189790751	15.66025376	20.00777623	38.26548324	70.93036962	180.2031971	**366**
BN	0.3472878931	0.5956118018	2.398208269	6.299467383	13.51860794	15.8272506	31.3645983	60.1910842	157.178055	114
BN+Copula	**0.02069072726**	**0.3926083802**	**1.639647758**	**4.493985698**	**11.65092857**	**14.2231747**	**29.83172102**	**58.99229085**	**155.4051754**	150
Copula	0.3994397926	0.6341443773	1.843484274	4.945975734	11.80966924	14.3795815	31.01540352	60.05473323	156.6346908	104
IPF	0.20162097	0.9114930804	9.968902414	28.28399259	69.8280417	99.23126848	222.2522842	445.2271834	1178.122151	0

Table 5.4.3 Metrics Evaluation for PUMA 1203

	1	2	3	4	5	6	7	8	9	SZ
Ind	0.2538747653	0.6014095617	2.607974146	5.534160108	11.55185389	13.70587999	27.27486485	53.12703867	139.2030098	78
CTGAN	0.3678753435	0.7021547692	2.681948099	5.880901111	11.84346681	13.82054044	27.25898428	53.11979325	139.2030098	57
CTGAN+Copula	0.2458465297	0.5189610799	1.744839909	4.190408609	10.45569848	12.69026246	26.51094159	52.47094414	137.9782406	91
TVAE	0.4895993909	0.9927202666	6.819176795	15.69006526	32.98587955	41.75186952	81.54599731	148.1779419	375.9424537	133
TVAE+Copula	0.2774891166	0.624078894	2.804285557	6.638856984	15.02512281	18.90350481	37.38804346	69.21609774	177.4497159	**359**
BN	0.2729984757	0.5699414193	2.611354143	5.61100114	11.60089598	13.81705876	27.33653489	53.26812737	139.1836532	65

BN+Copula	**0.02490985405**	**0.41155 3628**	**1.63752 2522**	**3.91518 8339**	**9.80994 3007**	**12.1185 4543**	**26.1215 2412**	**51.9771 41**	**137.508 7924**	117
Copula	0.42272 73683	0.70564 45814	1.79222 0955	4.29156 5421	10.0104 49	12.2731 8937	27.1209 6322	53.2356 0163	139.048 0816	61
IPF	0.24861 08779	0.92423 86622	9.87388 3546	24.3029 0473	60.0973 5804	85.3110 7353	191.420 8231	383.248 747	1014.11 2248	0

Table 5.4.4 Metrics Evaluation for PUMA 1204

	1	2	3	4	5	6	7	8	9	SZ
Ind	0.24105 02073	0.51387 0958	2.35342 4273	6.10743 3274	13.0990 9496	14.7821 6417	28.1582 8628	53.8899 9158	140.109 8024	174
CTGAN	0.30085 44405	0.73861 50626	2.43394 0426	6.11371 5835	13.2590 1645	15.0159 9724	28.5583 1947	54.1437 1334	140.620 9756	119
CTGAN+Copula	0.13858 64596	0.68093 66467	1.66565 3501	4.48084 8237	11.7180 0512	13.6200 0751	27.2795 1798	53.1843 9203	138.931 0286	209
TVAE	0.31219 66514	0.75810 99802	5.63885 5064	15.5238 757	34.4444 4547	39.6694 1226	69.0264 2844	121.402 4626	301.094 3664	261
TVAE+Copula	0.15446 8137	0.65260 32657	2.76975 3329	7.68924 7691	16.8272 8136	20.3976 416	39.0729 5575	70.4920 7383	177.930 1269	**645**
BN	0.24079 77511	0.55097 86542	2.36385 1322	6.16678 1378	13.1151 3095	14.8253 4204	28.0917 4354	53.9348 5274	140.586 5729	177
BN+Copula	**0.01957 054963**	**0.35775 85852**	**1.48836 3659**	**4.02332 8029**	**11.1113 6491**	**13.0261 3612**	**26.8797 3414**	**52.7713 4534**	**138.256 2633**	254
Copula	0.38057 84387	0.60688 42902	1.67434 5863	4.46873 0133	10.8658 5051	12.8427 1042	27.6183 2878	53.7080 2791	139.006 4447	244
IPF	0.18491 63528	0.72240 79395	9.15501 2796	25.6567 9395	63.5045 3809	90.1447 7203	202.517 6849	405.250 0005	1072.75 1925	0

Table 5.4.5 Metrics Evaluation for PUMA 301

	1	2	3	4	5	6	7	8	9	SZ
Ind	0.19732 4888	0.83138 21698	2.41724 6625	6.09360 5588	12.6388 3328	15.0301 2952	30.2544 4563	58.7510 4667	153.039 7572	88
CTGAN	0.23744 15215	0.76698 49239	2.41687 548	5.91950 554	13.0072 5527	15.4275 9834	30.9320 5553	59.5414 7368	154.350 9261	125

CTGAN+Copula	0.08627003666	0.5407647642	1.654503687	4.423536024	11.045133404	13.8376819	29.44872435	58.140596682	152.7355813	82
TVAE	0.1809967416	1.15552107	7.102355124	19.405555732	42.503522208	52.794956606	103.4492656	187.3407718	482.7997089	109
TVAE+Copula	0.0641131764	0.5875026518	3.040191888	7.783527304	16.429000061	21.322844799	41.448338867	78.069338044	200.7845769	**201**
BN	0.2033180251	0.7883860984	2.420584378	6.046651748	12.686925587	15.139411727	30.199409855	59.284700324	153.6463024	119
BN+Copula	**0.03509129002**	**0.3680821859**	**1.557716075**	**4.217808699**	**10.893115912**	**13.633446413**	**29.304223868**	**57.779704848**	**151.898707916**	105
Copula	0.3736374511	0.619269174	1.758513781	4.560477519	10.964644211	13.712644542	29.98896105	58.91415572	153.2422057	58
IPF	0.2743472251	1.177862985	10.151088353	28.422355811	70.060285556	99.277661746	222.5115489	444.8466852	1177.519393	0

Table 5.4.6 Metrics Evaluation for PUMA 302

Based on the results from Table 5.4.1-5.4.6, the most conclusions are similar as previous experiments except that IPF don't perform best even at the first dimension, which again shows that lower source size harms IPF's performance. And the parametric copula performs nearly as good as BN and BN+Copula when the dimension is high.

Chapter 6: Conclusion

In the research we focus on exploring different methods to generate synthetic populations, including IPF, and machine learning methods (Bayesian Network, CTGAN and TVAE). We also apply a framework called copula to improve the quality of synthetic populations generated by these methods and compare them with the original ones. We compare the performance of all these methods at three different levels: state level, county level and puma level. The results can give us the following conclusions in general:

(1) IPF performs well at the state level and when the dimensions of the variables in the data are low. When the data comes from state level to county level and puma level and the dimensions of the variables in the data become higher, the performance of IPF becomes worse and worse. So, IPF is more suitable to apply at a higher geographical level and lower dimensions of the data. In contrast, the performances of machine learning methods and Bayesian Network aren't influenced too much when it comes to a lower geographical level.

(2) Machine learning methods perform worse compared with independent baseline and IPF when the dimensions of the variables in the data are low in generating accurate synthetic population. But when the dimensions of the variables in the data becomes higher, machine learning methods finally exceed the performance of IPF and are very close to the performance of independent baseline. For the diversity of data, machine learning methods show huge advantages compared with independent baseline and IPF.

(3) Among the machine learning methods including Bayesian Network, CTGAN and TVAE, Bayesian Network always performs best in generating accurate synthetic population compared with CTGAN and TVAE no matter which geographical level. For the diversity of the synthetic population, it depends on the geographical level. At the state level, Bayesian Network still performs best in generating a diverse synthetic population. But when it comes to the county level, the situation changes that TVAE exceeds the performance of Bayesian Network and CTGAN in sampling zero. It shows that Bayesian Network and CTGAN are better at generating an accurate synthetic population, but TVAE is better at generating a diverse synthetic population at a lower geographical level. Therefore in general, Bayesian Network is the best generative model when we want the most accurate synthetic populations. While at some geographical level, other machine learning methods like TVAE will exceed Bayesian Network in generating a more diverse synthetic population.

(4) In the experiment at puma, parametric copula also performs well compared with BN and BN+Copula when the dimension is high, which means the copula itself is also a good generative method.

(5) For the methods that apply copula framework, including CTGAN, TVAE and Bayesian Network, all of them are improved in the performance not only the accuracy of the synthetic population but also their diversity. Therefore, it turns out that copula is a very useful technique to improve the quality of the synthetic population when we want to use the current generative models to generate synthetic populations or explore new generative models.

Contributions

In this research, I mainly make the following several contributions:

(1) Join choosing the methodologies that are suitable for the research;

(2) Add parametric copula into comparisons at the puma level and make sure all the methodologies are compared at the same level;

(3) Join the modifications of codes used for other methodologies;

(4) Generate the results included in this paper;

(5) Join the final writing of literature review, methodology, results and conclusions.

References

[1] Beckman, Richard J., Keith A. Baggerly, and Michael D. McKay. "Creating synthetic baseline populations." Transportation Research Part A: Policy and Practice 30.6 (1996): 415-429.

[2] Eluru, Naveen, et al. "Population updating system structures and models embedded in the comprehensive econometric microsimulator for urban systems." Transportation Research Record 2076.1 (2008): 171-182.

[3] Bar-Gera, Hillel, et al. "Estimating survey weights with multiple constraints using entropy optimization methods." 88th annual meeting of the Transportation Research Board, Washington, DC. 2009.

[4] Ye, Xin, et al. "A methodology to match distributions of both household and person attributes in the generation of synthetic populations." 88th Annual Meeting of the transportation research Board, Washington, DC. 2009.

[5] Wong, D.W.S. The Reliability of Using the Iterative Proportional Fitting Procedure. Professional Geographer, 44(3), 1992, pp. 340-348.

[6] Konduri, Karthik C., et al. "Enhanced synthetic population generator that accommodates control variables at multiple geographic resolutions." Transportation Research Record 2563.1 (2016): 40-50.

[7] Farooq, Bilal, et al. "Simulation based population synthesis." Transportation Research Part B: Methodological 58 (2013): 243-263.

[8] Sun, Lijun, and Alexander Erath. "A Bayesian network approach for population synthesis." Transportation Research Part C: Emerging Technologies 61 (2015): 49-62.

[9] Borysov, Stanislav S., Jeppe Rich, and Francisco C. Pereira. "How to generate micro-agents? A deep generative modeling approach to population synthesis." Transportation Research Part C: Emerging Technologies 106 (2019): 73-97.

[10] Garrido, Sergio, et al. "Prediction of rare feature combinations in population synthesis: Application of deep generative modelling." Transportation Research Part C: Emerging Technologies 120 (2020): 102787.

[11] Xu, Lei, et al. "Modeling tabular data using conditional gan." Advances in Neural Information Processing Systems 32 (2019).