Progress Report: Privacy-preserving Synthetic Data Generation for Sortition Algorithms

Maya Gambhir

December 2025

1 Introduction

As the study of sortition and its use in democratic deliberation becomes relevant and more closely studied, an important barrier to research in this area becomes clear. Due to the personal and sensitive nature of the data this work involves, data sharing in the field is limited to a select few researchers, making it difficult for those without data access to make meaningful research progress and effectively test their algorithms. We aim to construct a "data generator" that produces realistic sortition data that has provable guarantees of both privacy, similarity to real data, and its ability to replicate algorithmic behavior.

Sortition objectives under consideration. We have six instances of pools and corresponding quota sets to work from. Our current testing includes algorithms and objectives from the following papers:

1. LEXIMIN: maximizes the minimum selection probability given to any agent; if ties occur, it maximizes the second-lowest probability, and so on
2. MINIMAX: minimizing the maximum selection probability
3. MAXIMIN: maximizing the minimum selection probability
4. NASH: maximizes the geometric mean of the selection probabilities
5. Goldilocks: penalizes multiplicative deviations both above and below the average selection probability (k/n). It aims to recover the best available trade-offs for both maximum and minimum probabilities in a given instance
6. LEGACY (heuristic approach)

These metrics (barring the heuristic legacy approach) rely entirely on the spread of selection probabilities of each volunteer for the final panel. This provide substantial motivation one of our key objectives, replicating selection probabilities.

1.1 Problem Model (Sketch)

We model the target inputs and outputs of our algorithm below:

Input: A quota-pool pair (Q, P) where Q defines a list of min and max quotas for each feature value and P contains a particular set of features, with associated values, for each of n people.

Output (Q, P') where (Q, P') performs similarly to (Q, P) on target objectives. These target objectives are often captured by min and max selection probabilities for each person. We also hope to replicate other aspects of the "real" pool such as uniqueness of each pool member, with their feature forming a vector.

Hypothesis If we can replicate both the *mean rates* of feature-value occurrence (relative to the quotas) and the *covariance/correlation* between various feature-values then we can replicate the following aspects of P':

1. The spread/distribution of selection probabilities
2. The spread/distribution of unique feature-value vectors

We believe replicating these aspects of P' will lead to approximation of how it performs on standard sortition objectives.

Current Guiding Question: How can replicate the marginal probabilities of feature values as well as their covariances?

2 Empirical Experiments

We completed a selection of empirical approaches to this problem, notably the simple baseline of directly sampling pool members' feature-values individually, using the marginal probability of a value v for each feature. We find that replicating the marginals does replicate the selection probability distribution with moderate success.

First we introduce some quick notation:

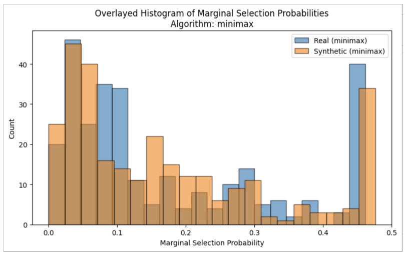
Quick Notation:

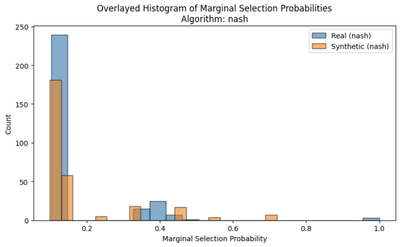
1. $r_{f,v}$ = fraction of P that has value v for feature f.
2. $c((f_1, v_1), (f_2, v_2))$ & $c(f_1, f_2)$ denotes some measure of correlation between particular features or specific feature value pairs.

After completing empirical experiments on our chosen subset of sortition algorithms, we found that if we replicate $r_{f,v}$ we decently approximate objective 1 (spread/distribution of selection probabilities). However, we perform extremely poorly on objective 2 (Spread/distribution of unique feature-value vectors), which was expected given the random and independent nature of the sampling approach.

2.0.1 Spread of selection probabilities

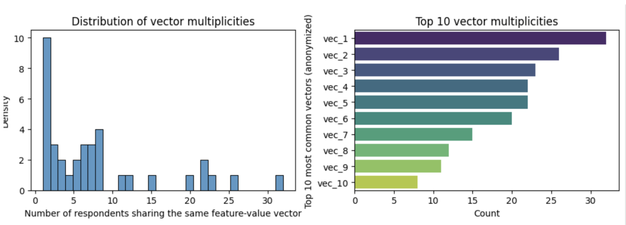
Below are plots that demonstrate the selection probabilities using both the original data and synthetic data for minmax and nash objectives on one instance.

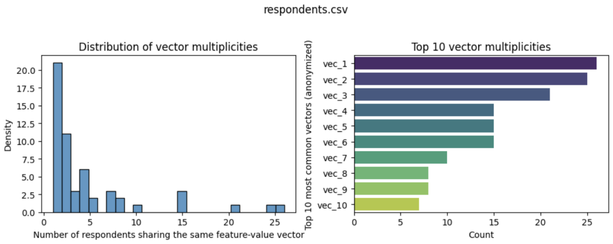




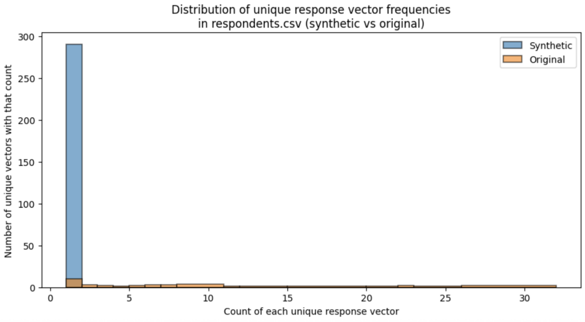
2.1 Comparing vector multiplicities across instances

Now we consider our second objective, replicating vector multiplicity. The distribution of vector multiplicities has a relatively consistent shape across our current instances. With a large amount of unique feature vectors, and a longer tail of repeated, common feature vectors.





As mentioned previously, our baseline performed poorly on replicating this spread of multiplicities. The example below shows that there are far less unique feature vectores than previously.



2.2 Additional Data Exploration

What do feature correlations look like? Weak but evidently important for replicating behavior.

What do clustering attempts reveal about the uniqueness of feature vectors? They are not very effective to cluster, likely because of the large number of very unique vectors?

3 How do we create a distribution to draw from that replicates correlations between features *and* their marginals?

Central Question : Can replicating pairwise covariance and marginal frequency of feature-values in synthetic data allow us to approximate the **probability distribution** and **relative uniqueness** of vectors in the final pool?

3.1 How can we define covariance?

The key question is whether we are willing to define per feature-value correlations or want a more general per-feature correlation, only the former works with standard covariance.

Option 1: Per-feature covariance matrices For each feature, for each unique pair of values those features can take on, we calculate the covariance. This gives us a $kt \times n$ matrix where feature A has k distinct values and feature B has n distinct values.

Option 2: Cramers V Provides a normalized measure of dependence between two features. Because the raw chi-squared statistic scales with sample size and number of values for each feature, Cramér's V rescales it into a 0–1 effect size that is comparable across variables.

$$V = \sqrt{\frac{\chi^2}{n\,(k-1)}},$$

Where χ^2 is the chi squared statistic and n, k are the number of values for each feature in the current pair.

Option 3: Mutual Information For two discrete variables X and Y with joint distribution $p(x, y)$ and marginals $p(x)$ and $p(y)$, the mutual information is defined as

$$I(X;Y) = \sum_x \sum_y p(x,y) \log\left(\frac{p(x,y)}{p(x)p(y)}\right).$$

This quantity captures how much knowing the value of one demographic attribute reduces uncertainty about the other, with $I(X;Y) = 0$ indicating independence and larger values reflecting stronger associations.

3.2 Methods for sampling with defined covariances

Gaussian-copula Sampling from a copulae allows you to produce multivariate samples based on a correlation matrix and marginal distributions.

Inputs (Maringals and covariances): $p_{i,c} = P(X_i = c)$ for $i \in [1, F], c \in [1, k_i]$ for F features and k_i values for feature i.

Covariances we have: $Cov(X_i = c, X_j = d)$, but we need a fxf matrix form ideally for this approach. We can also treat each value as an indicator of a separate feature. We want to sample vectors $X \in R^d$ that match the above statistics. We can then map each value to an interval on $[0, 1]$ based on their marginal probabilities. This lets us sample from a uniform distribution and end up with a categorical variable. We get CDF of the categorical variable as below:

$$F_{X_i}(c_j) = \sum_{m=1}^{j} p_{i,c_m}$$

At a high level, the procedure as follows:

categorical → uniform → gaussian → gaussian + covariance → uniform → categorical

The theory states that adding the covariance matrix in and mapping back to uniform/categorical preserves the correlations.

3.2.1 Discussions on copulae:

The approach is conceptually appealing because it cleanly separates marginal distributions from dependence structure and naturally yields a generative model over full feature vectors. However, the theoretical guarantees for categorical data remain unclear, as existing work appears to rely primarily on empirical justification and warrants closer examination. There are also technical challenges in mapping discrete covariances to a Gaussian correlation matrix and in ensuring that one-hot constraints are preserved after the transformations.

3.2.2 Related work (synthetic populations with copulae)

This paper uses aggregate data from m size d chunks of the population to construct a full $m * d$ population. They do this by calculating the aggregate statistics on indicators of each value (which gives a covariance matrix and marginal rates/percentages of people having that value). I don't think this paper in particular provides solid theoretical guarantees, but the approach is motivated by existing theory.

3.3 Another idea: iterative proportional fitting

We start with a contingency table across features (values). The table is an f dimensional matrix where f is the number of features. "Scale up", cycling through dimensions until we match our marginals. All pairwise correlations are preserved, which gives exact covariances rather than approximate (although it is slower). We can sample by treating each cell as a probability.

Iterative Proportional Fitting (IPF) Outline:

We have a F dimensional matrix, where the jth dimension is the number of values that feature has.

Can this method synthesize across multiple matrices? Does it treat the draw of the pool that we see as a draw from random generation? Currently, we belive the answer is no.

- Let $p(x_1, \ldots, x_n)$ be the joint probability table over n categorical variables.
- Suppose we are given target marginals or pairwise distributions, e.g. $p_i^{\text{target}}(x_i)$ or $p_{ij}^{\text{target}}(x_i, x_j)$.
- Initialize $p^{(0)}(x_1, \ldots, x_n)$ (uniform or any positive table).
- Iterate until convergence:

$$p^{(t+1)}(x_1, \ldots, x_n) = p^{(t)}(x_1, \ldots, x_n) \cdot \frac{p_i^{\text{target}}(x_i)}{\sum_{x_{-i}} p^{(t)}(x_1, \ldots, x_n)}$$

  for each variable i (or each pair (i, j) if fitting pairwise marginals), where x_{-i} denotes all variables except x_i.

- Repeat the updates for all marginals iteratively until $p^{(t)}$ matches the target marginals within tolerance.

3.3.1 Discussions on IPF

IPF's main strength is that it is a classical method with strong guarantees: when it converges, it exactly matches specified marginals and preserves pairwise relationships. However, in high-dimensional settings it becomes computationally heavy and is not naturally suited to serving as a generative model. This raises open questions about its convergence behavior in our dimensional regime and about how to efficiently sample full vectors from the resulting table. Overall, IPF is useful as a baseline or intermediate tool, but its scaling issues and limited generative capability make it unlikely to be the final solution.

3.3.2 Related with IPF: Generating Synthetic Populations using Iterative Proportional Fitting

The paper uses IPF to fuse two imperfect data sources: a detailed but small sample of people (the 5% microdata) and coarse census tables that only give totals for each demographic category. IPF repeatedly adjusts the sample's joint distribution so that it lines up with the real census totals for age, sex, and later each municipality, essentially "stretching" the sample to match what the full population must look like. After aligning the data at the canton level, the authors run IPF again using each municipality's own totals so that every local area ends up with the right counts while keeping the overall correlation patterns from the sample. They then draw individuals from this adjusted joint distribution to create a full synthetic population that statistically mirrors the real one.

4 Next Steps

In the following semester we aim to create a formal mathematical model for sampling, and validate it on our instances. There is more related work than originally anticipated on synthetic data generation (as described above), so exploring this work, and its potential to inform our own model is important.