Data Approximation by Statistical Inferencing

Xander Wilcke

1 Introduction

Synthetic data is becoming an increasingly important topic within and outside of academic research, especially when privacy sensitive data is being considered. Directly using privacy sensitive data in experiments or simulations is generally considered unsafe, as there is a non-zero chance that some elements of these data find their way into the results. Even when the data has been thoroughly anonymised, it remains impossible to guarantee that there are no lingering dependencies between the data points which can reveal sensitive information. By using synthetic data, however, we can overcome these problems as there is no trace of the original data left to be concerned about. Instead, a synthetic dataset merely mimics the privacy sensitive data, enabling researchers to perform experiments and simulations as if it were the actual dataset, and allow them to produce results and draw conclusions that also hold on these data.

To generate a synthetic dataset it is necessary to have access to the original and privacy-sensitive data. This might not always be possible, however, for example when the providers of the data have restricted access to the dataset or when this dataset is only accessible through a restricted environment, hence limiting the use of data generation methods. Special clearance must be requested in those cases, which, if granted, often allow for but a short window of opportunity to generate the synthetic data. Since this leaves little time to build the necessary methods, it is important that these methods are already in a (near) finished state when such an opportunity presents itself. This poses an interesting conundrum, however, since we need to have access to the data in order to build and test our methods.

A mock dataset can be used to build and test our methods outside of a restricted environment. Different from a synthetic dataset, a mock dataset is but a naive approximation of the actual data, typically derived through basic statistical inferencing. Here, we propose a method that combines statistical inferencing with Monte Carlo sampling to cost-effectively generate a mock dataset [1].

[1] A R implementation is available at `github.com/SPINLab/synthpop`

2 Mock Data Generation Process

A two-phase approach is used to generate a mock dataset: first, the parameters of the underlying distributions are inferred column by column from the original data, after which these parameters are used to locally reconstruct an approximation of these data. A more detailed overview of these two phases follows.

2.1 Statistical Inferencing

The first step in the inference process involves Monte Carlo sampling. By repeatedly sampling random values from the column at hand, we are constructing a subset of the data, X^-, that is just enough to retain the original underlying distributions. When the dimensions of the original data are high, this also helps reduce the number of samples to a more manageable amount, whereas if the dimensions are low, this method enables us to create a sufficiently large dataset from which statistical inference is possible. The sample size is the only necessary hyperparameter of our method, and offers a trade off between precision and computability [2].

Once X^- is created, the next step involves the conversion of symbolic data to their closest numerical equivalents. This is a necessary precursor for allowing statistical inference methods, which operate solely on numerical data. For nominal or ordinal data, we can simply enumerate the distinct values and use their indices as replacement values. Dealing with natural language is more challenging, since this would require a more complex approach that enumerates at the word or symbolic level. Instead, a cheap alternative is chosen that captures the structure of the sentences while omitting their semantics. For this purpose, only the minimum, maximum, and average number of words in the sentences is retained.

With a numerical sample at our disposal, the third step involves the addition of noise to X^-, creating $\tilde{X}^-$. This ensures that any future estimations cannot be traced back to the original data, allowing us to safely determine the type and number of distributions which underlie these data. To estimate the type of distribution, we first calculate relevant measures from $\tilde{X}^-$, which are then used to populate a new random distribution for each of the considered types. Which distribution best fits the data can now be estimated by comparing $\tilde{X}^-$ to these random distributions. Similarly, we can estimate whether we require a mixture model by computing the Kullback-Leibler (KL) divergence between a random distribution and the data, and by comparing the result with the KL-divergence between two random distributions (the null hypothesis).

In the event that the KL-divergence suggests the use a mixture model, we can determine the exact number of distributions, or k, empirically. Hereto, we split $\tilde{X}^-$ into two random halves for which we compute the parameters of their mixture models for all values of k that are considered [3]. For each value of k, we

[2] In our research we use a sample size of 10E4.

[3] In our research we consider $2 \geq k \geq 8$.

can now use the found parameters to populate two new random distributions, one for each half, for which we once again compute the KL-divergence. Different from before, however, the KL-divergence used here is weighted to prefer values of k. The value for for which the KL-divergence is the lowest represents the optimal number of distributions.

With the type and number of distributions estimated, we have all information necessary to compute the final approximation of the distribution underlying $\tilde{X}^-$. As a final step, we once again fit the distribution to the data, but now with the assumed type and number of centres. The parameters of this approximate distribution are saved for use by the reconstruction phase.

2.2 Data Reconstruction

The reconstruction phase begins by sampling N values from a random distribution which matches the type and parameters that were inferred previously, with N the number of rows in the original dataset. In the event of a mixture model with k centres, the samples are proportionally drawn from all k distributions. The result is a set S of length N. To also match the original number of distinct values $M \leq N$, a set of M different values is drawn from S, from which N new samples are drawn with repetition to form $\hat{X}$.

With the underlying distribution reconstructed, the next step involves the restoration of non-numerical data. For nominal and ordinal columns, the numerical values in $\hat{X}$ can be replaced by symbolic values from a dictionary-encoded corpus, hereby repurposing the rounded numerical values as indices. For natural language, a *lorem ipsum* generator can be used to produce syntactically sound sentences. To approximate the sentence structure of the original data, each generated sentence is cut off after a random number of words, hereby taking the minimum, maximum, and average sentence length of the original into account.

A final step is the introduction of erroneous values, such as `NA`, in $\hat{X}$. This can be easily be accomplished by randomly and proportionally replacing a value with an erroneous value for all columns in the original dataset for which such values were observed. It is important that these erroneous values are not left out, since any method developed for and tested on clean data might run into problems with real-world data.

3 Conclusion

The method proposed here can be used to generate a mock dataset when the actual data is only accessible through a restricted environment. This mock dataset is but a rough approximation of the original data, and is therefore only suitable for developing and testing purposes.

Several improvements can be made that were omitted because of time constraints. Firstly, the current methods forgoes any row-wise harmonisation, which means that all dependencies between the values on the same row are lost. A direct consequence of this is that any natural clustering of the instances in the original data does not return in the reconstructed data. Another improvement that can be made is the reduction of the number of assumptions made during the inference phase, which would improve the precision of the approximation.