Sharing confidential data for algorithm development by multiple imputation

Sicco Verwer
Research and Documentation Centre, Ministry of Security and Justice, the Netherlands

Institute for Computing and Information Sciences, Radboud University Nijmegen, the Netherlands

Susan van den Braak
Research and Documentation Centre, Ministry of Security and Justice, the Netherlands

Sunil Choenni
Research and Documentation Centre, Ministry of Security and Justice, the Netherlands

ABSTRACT

The availability of real-life data sets is of crucial importance for algorithm and application development, as these often require insight into the specific properties of these data. Often, however, such data are not released because of their proprietary and confidential nature. We propose to solve this problem using the statistical technique of multiple imputation, which is used to as a powerful method for generating realistic synthetic data sets. Additionally, it is shown how the generated records can be combined into networked data using clustering techniques.

1. INTRODUCTION

The availability of real-life data sets is of crucial importance for research and development in various fields. For example, in computer science, real-life data sets are used to test the effectiveness and performance of algorithms, while in social sciences these data sets are collected to test or generate hypotheses. However, it is common that such databases are not publicly available, for instance, because of the privacy-sensitive, confidential or proprietary nature of the data. Nevertheless, to test tailor-made algorithms and applications access to the real-life data is required.

For a current project on automatically detecting fraud by legal persons, we obtained access to openly available data on legal entities and their relations and representatives. These data are enriched with information on possbile indicators of fraud. Given the confidential nature of these fraud indicators and the fact that these data contain a large amount of personal information, it is not desirable to share these data. For development purposes, however, it is not required to know all details (e.g., names) of legal persons that have a high probability of committing fraud. It suffices to know how many persons are suspicious and which characteristics they have.

We are therefore interested in developing a method for creating realistic synthetic copies of the fraud detection database that can be used to develop and test our fraud detection algorithm, without running the risk of exposing sensitive information. Additionally, the artificial data can be shared with the scientific community to further improve our algorithms and verify our claims. Afterwards, the developed algorithm can be run on the real data by the data owners.

Some alternatives to our approach exist (Figure **??**). For instance, privacy enhancing technologies may be applied to encrypt or discard privacy sensitive data attributes from a database [3]. The newly obtained database can then be handed over for development purposes. Unfortunately, however, in addition to privacy sensitive information, such technologies often also hide (delete) and distort (e.g., by data binning) several attributes and their properties entirely. These methods are therefore not suitable for development purposes [2]. Another alternative is that a third trusted party takes care of of the coordination of the development process from an organizational point of view [10]. Since such an organization has to judge frequently which part of a database, algorithm, or output may be shared with another party, this is a costly solution.

We propose to synthesize realistic data using the statistical technique of multiple imputation. Multiple imputation is a popular method for imputing (filling in) missing values in data sets. By applying it to a data set with additional rows filled with missing values, it can be turned into a powerful tool for generating realistic synthetic data. Although the idea of using multiple imputation to synthesize a data set is not new, see e.g., [7, 8, 5, 4, 1]), to the best of our knowledge, we are the first to use this technique on a database consisting of *networked data*, that is, the fraud detection database. In addition to generating realistic values for the attributes of the rows in every data table (e.g., people), it is important to generate realistic links between these rows. Our method consists of two parts: 1) generate the attributes of pairs of linked rows; and 2) combine these links into a network using

clustering techniques. Our main contributions are providing a method for testing whether the generated data set is realistic, gearing it towards maintaining algorithmic performance, and showing how such method can be used in practice.

2. MULTIPLE IMPUTATION

Imputation is the process of filling-in unknown or missing data values by estimates of those values. Many of such imputation methods exist. For instance, replacing every missing value by the variable mean is a very simple imputation method. Another common imputation method is hot-deck imputation [6]. This replaces a missing value by copying a known value from a randomly selected similar data record.

For example, suppose we have information on the income, mortgage, and age of two people: $\langle 35K, 200K, 28\rangle$ and $\langle 55K, 500K, 53\rangle$. For another two people, we know only two of these values: $\langle 40K, 300K, ?\rangle$ and $\langle ?, 300K, 45\rangle$. Based on these data, possible values for the first unknown are 28, 53, and 45. A biased die is rolled to decide which one is used, with 45 as most probable outcome since this value comes from a record that with the same mortgage. The most probable outcome for the second unknown is $40K$.

Imputation methods such as this can be used to impute the missing values in a data set. When such an imputation method is probabilistic, the resulting data set can be viewed as a draw from a (very complex) probability distribution. Consequently, if one would repeat the imputation process, there can be differences in the imputed values. The key idea of multiple imputation [6] is to overcome this uncertainty of the imputed values by imputing the data multiple times. The uncertainty in the imputed values is then represented by the difference between (variance of) the values assigned in the different imputations.

In this paper, we use a sophisticated tool in R[1] called MICE for imputing our data [9]. MICE stands for multiple imputation by chained equations. Depending on the type of an attribute (nominal, ordinal, or continuous), MICE chooses a different method to impute the missing values. The default setting is to use predictive mean matching for continuous attributes. In addition to imputing the data multiple times (the default is 5), MICE uses a technique called chained equations for conditioning the imputation method for one attribute on the already imputed values of another attribute.

Intuitively, the chained equations method works is as follows. Suppose that in our example, $\langle ?, 300K, 45\rangle$ is imputed to obtain $\langle 40k, 300K, 45\rangle$. This influences the imputation of $\langle 40K, 300K, ?\rangle$ because they now match on income as well as mortgage, increasing the probability that 45 is going to be imputed. On the other hand, if $\langle 55k, 300K, 45\rangle$ is obtained instead of $\langle 40k, 300K, 45\rangle$, the probability of imputing 45 in $\langle 40K, 300K, ?\rangle$ will be decreased since the incomes are now different. The chained equations technique is designed to overcome these issues. The technique starts without any imputed values and selects an attribute A. All the missing value in A are then imputed. Then another attribute B is selected, and its missing values imputed. These new B values are thus conditioned on the already imputed values for A. Since the new B values can influence the imputation of A, all of the missing values of A are later imputed again, now conditioned on the new B values. Afterwards, the B values are imputed again, etcetera. This iteration of imputations is continued until the values of A and B do not change much, i.e., until they have converged. The chained equations technique is an example of a Markov-chain Monte Carlo method.

[1] `http://http://www.r-project.org/`

3. IMPUTING GOVERNMENTAL DATA

In this section, we demonstrate how a multiple imputation methods such as MICE can be combined with clustering techniques in order to synthesize a realistic copy of a networked real-life data set, in this case the fraud detection database.

3.1 The data

A legal person (LP) allows one or more natural persons to act as a single entity for legal purposes. Examples of legal persons are cooperations, companies, partnerships, firms, associations, and foundations. Each legal person must designate one or more representatives, such as a director, partner or manager, who is authorized by its articles of association or by power of attorney. Legal persons can be represented by one or more natural persons (NP), but also by other legal persons. Those legal persons can in turn be represented by natural persons or other legal persons, etcetera. This means that the network of persons related to a legal person may be several levels deep and may consist of relations between legal persons and natural persons (called 'LP-NP relations') or of relations amongst legal persons (called 'LP-LP relations').

Just like natural persons legal persons may commit financial crimes or employ fraudulent activities such as money laundering, tax fraud, or bankruptcy fraud. Therefore, in the Netherlands legal persons are frequently screened for misuse. In this automated process, based on a set of so-called risk indicators it is determined whether a legal person is likely to commit fraud.

However, assigning scores to legal persons is not straightforward. Therefore, the Research and Documentation Center of the Dutch Ministry of Security and Justice developed and tested several scoring functions based on a few rules of thumb. In order to evaluate whether these functions showed the desired behavior (not to many false positives or false negatives) they needed to be applied to a test database. Obviously, for the purpose of testing the behavior of the scoring functions, it is of utmost importance that the generated artificial database is realistic and resembles the real-life databases to a large extent.

3.2 Synthesizing links

For the purpose of this paper, we focus on the LP-LP relations in the fraud detection database. This means that all LP-NP relations are removed from the dataset. The thus obtained data set contains characteristics and risk indicators values of both LPs in the relations. MICE is used to impute likely values for a selection of these attributes. Because of memory restrictions, we sampled 50.000 of the LP-LP relations uniformly at random from the database, and used MICE to synthesize 15.000 new ones. In addition, we add

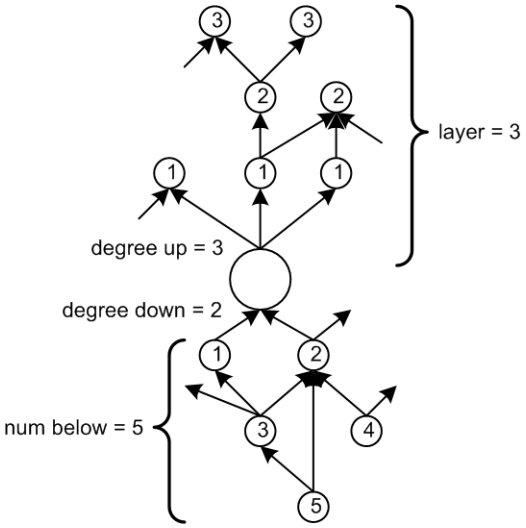

Figure 1: The network properties of LP nodes that are synthesized (the values shown are computed for the large central node).

attributes that indicate some of the network-related properties of these relations, depicted in Figure 1: 1) the degree (number of direct relations), 2) the number of layers above a relation, and 3) the number of LPs below a relation. Likely values for these attributes are then also synthesized using MICE. The purpose of these attributes is to indicate likely positions of the generated LP-LP relations in the resulting synthetic network. The idea is that as long as we make sure that these properties and their correlations with the attributes of LP-LP relations are kept intact, the resulting network will be realistic.

The set generated by MICE will consist of synthetic LP-LP relations, along with likely network properties. However, since every LP in these relations occurs exactly once, they do not yet form a network (see the left part of Figure 2).

3.3 Clustering synthesized links

A network of LPs is created by combining LPs from different synthetic relations into a single LP that occurs in all of these relations as shown in Figure 2. Intuitively, we should combine those LPs that are likely to be the same, i.e., those that look most similar. We use the standard Euclidean distance to measure the similarity between two generated LPs, and a standard clustering algorithm to select which ones to combine. We use the hierarchical clusterer available in the R cluster package[2].

Typically, clustering algorithm are applied in settings where one knows the number of clusters that should be created beforehand. In our case, however, it is more natural to let this be determined by the cluster content, i.e., the attribute values of the clustered LPs. Intuitively, LPs that are likely to have many relations in a realistic network should end up in large clusters. We achieve this by computing the mean value of the degrees (generated by MICE) of the LPs in a single

[2]`http://cran.r-project.org/web/packages/cluster/`

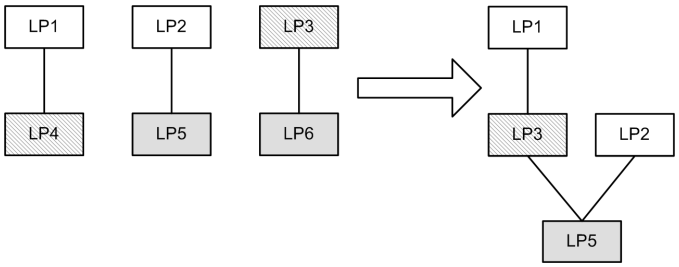

Figure 2: Our method for combining generated links into a network based on similarity.

cluster. This value should be roughly equal to the number LPs in that cluster. We used a simple greedy method using the ordering of pair-wise clusterings (a dendogram) resulting from a hierarchical clusterer:

1. Initially, every LP i is in cluster C_i containing only i.
2. Let C_i-C_j be the next pair of clusters combined by the hierarchical clusterer, with $|C_i| \leq |C_j|$.
3. Let $C^* := C_i \cup C_j$ be the result of this combination.
4. If $\frac{\sum_{c \in C_i} \mathsf{degree}(c)}{|C_i|} \leq |C_i|$, set $C^* := C_j$.
5. Else if $\frac{\sum_{c \in C_j} \mathsf{degree}(c)}{|C_j|} \leq |C_j|$, set $C^* := C_i$.
6. Else if $\frac{\sum_{c \in C^*} \mathsf{degree}(c)}{|C^*|} > \frac{3}{4} \cdot |C^*|$, set $C^* := C_i$.
7. If C_i-C_j was not the last pair, goto 2.

The set of clusters not combined with C^* gives us sets of similar LPs. Although we cannot guarantee that our proposed greedy method provides the best possible clustering or an approximation thereof, in practice the result was very sensible. The constant of $\frac{3}{4}$ is chosen such that the resulting set C^* cannot become much larger than the sum of the degrees of the clustered LPs. Also note that dividing the set of LPs into subsets with size constraints that depend on the obtained subsets is a combinatorial problem that is likely difficult to optimize, especially considering the fact that that finding a hierarchical clustering under a given bound is already NP-hard.

Every set of clustered LPs represent a single LP in the synthesized network. We create these single LPs by choosing one at random from every set. We then create relations between the LPs chosen from sets that contained LPs that shared a relation. This gives us a network of LPs. Moreover, due to the use of network-related properties during clustering, the resulting network structure is realistic in terms of these properties.

3.4 Results

There are two main criteria for our imputation method: 1) the artificial database should produce similar results on the fraud detection algorithms as the original and 2) the synthezised network has to be realistic.

Table 1: Distributions of the fraud detection scores in the original sample, and the synthesized data set after imputation and after clustering.

score	0.00	0.02	0.24	0.26	0.34	0.36	0.42	0.43	0.44	0.46	0.47	0.48	0.49	>0.5
original	0.82	0.02	0.06	0.00	0.04	0.00	0.02	0.00	0.01	0.01	0.02	0.00	0.00	0.00
imputed	0.75	0.03	0.07	0.00	0.06	0.00	0.03	0.00	0.02	0.01	0.03	0.00	0.00	0.00
clustered	0.81	0.04	0.06	0.00	0.04	0.00	0.02	0.00	0.01	0.01	0.01	0.00	0.00	0.00

We first evaluate and compare the performance of the developed fraud detection algorithm on the three databases: 1) the original sample, 2) the artificial data set after imputation, and 3) the data set after clustering. This algorithm is a simple scoring function that assigns a value between 0 and 1 to an LP based on its risk indicators. The distribution of the assigned values for each of the three databases is shown in Table 1. These scores are computed over the 100.000 LPs in the sample (two per relation), 30.000 imputed LPs, and the 17.669 LPs that remained after clustering. The scores are probabilities of fraudulant behavior assigned by the fraud detection algorithm.

All score values in Table 1 are possible values that can be obtained using the available indicators. Also the fact that most LPs get assigned a fraud score of 0.00 remains intact after both imputation and clustering. Although the imputation does create an increase in the number of LPs with a fraud score greater than 0, this increase is not unrealistic. Strangely, this number increases again after clustering. We believe this occurred due to the random selection of one LP from every clustered set. Another important observation is that the values that occurred with probability 0, still occur with probability 0 after both imputation and clustering. Thus, although these values can occur, they are not synthesized by MICE and our clustering method. The shapes of the three distributions are also very similar. These results indicate that the generated database is realistic in terms of the fraud indicators, including their correlations.

The pair-wise distributions of the network-related properties show no large differences between the original database and the synthesized database after imputation and only a few differences after clustering. Therefore, we conclude that the generated database of LP-LP relations is realistic with respect to the network-related attributes. We note, however, that the resulting network is incomplete. Due to sampling and clustering, too few relations were generated to ensure that every node is in the layer is should be and has the amount of relations it should have. Furthermore, note that a network that is realistic with respect to our network-related attributes does not imply that it is realistic with respect to all network-related attributes such as the number of triangles and they way they interconnect/overlap. Creating realistic networks with respect to these properties is a lot more difficult because they span over multiple nodes and links.

4. CONCLUSION

In this paper we showed how multiple imputation in combination with clustering methods can be used to synthesize a realistic copy of a real-life database. To the best of our knowledge, our method is the first that can be used to synthesize realistic networked data. We applied our method to a database used for fraud detection. The results are positive: the generated database is realistic with respect to both fraud-detection and network-related attributes.

Our method provides an interesting alternative to existing privacy enhancing technologies such as encryption and data distortion. One benefit is that since the generated data is entirely artificial, there is very little risk of exposing privacy sensitive information other than the overall data characteristics. The main advantage of our method over existing technologies is that it keeps all the data characteristics intact. This guarantees that algorithms developed using the artificial data are directly applicable to the real database.

5. REFERENCES

[1] A. Alfons, S. Kraft, M. Templ, and P. Filzmoser. Simulation of synthetic population data for household surveys with application to EU-SILC. Technical report, Vienna University of Technology, 2010.

[2] R. Choenni, J. van Dijk, and F. Leeuw. Preserving privacy whilst integrating data: applied to criminal justice. *International Journal of Government and Democracy in the Information Age*, 15(1-2):125–138, 2010.

[3] H. Federrath. Privacy enhanced technologies: Methods–markets–misuse. *Trust, Privacy, and Security in Digital Business*, pages 1–9, 2005.

[4] P. Graham and R. Penny. Multiply imputed synthetic datafiles. Technical report, Satistics New Zealand, 2007.

[5] J. P. Reiter. Releasing multiply imputed, synthetic public use microdata: an illustration and empirical study. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 168(1):185–205, 2005.

[6] D. B. Rubin. *Multiple Imputation for Nonresponse in Surveys*. John Wiley & Sons., New York, 1987.

[7] D. B. Rubin. Statistical disclosure limitation. *Journal of Official Statistics*, 9(2):461–468, 1993.

[8] D. R. T.E. Raghunathan, J.P. Reiter. Multiple imputation for disclosure limitation. *Journal of official statistics*, 19(1), 2003.

[9] S. van Buuren and K. Groothuis-Oudshoorn. mice: Multivariate imputation by chained equations in r. *Journal of Statistical Software*, 45(3):1–67, 2011.

[10] S. van den Braak, R. Choenni, R. Meijer, and A. Zuiderwijk. Trusted third parties for secure and privacy-preserving data integration and sharing in the public sector. In *Proceedings of the 13th Annual International Conference on Digital Government Research*, pages 135–140, 2012.