CATEGORICAL DATA GENERATOR

Jana Cibulková – Hana Řezanková

Abstract

Generated data are often required for quality evaluation of a newly proposed statistical method and its results. The number of datasets can easily affect the robustness and validity of the statistical analysis. Since real datasets with desired properties are often hard to obtain, there is a need for generated ones. Many generators of quantitative data have been implemented in different software products. However, the situation is different for multivariate categorical data. This paper presents an algorithm for generating categorical data in a new way proposed mainly for the needs of evaluation of cluster analysis methods. Datasets are generated as a mix of various multivariate distributions using Cholesky's decomposition. Generated datasets have required structure and properties, i.e. given number of clusters, given number of variables, given number of categories, given multivariate distribution and correlation within the cluster, etc. The algorithm is implemented in R software with the ambition to be included into R package *nomclust* in the future. The usage of the algorithm is illustrated on several examples.

Key words: categorical data, data generator, cluster analysis, NORTA transformation, Cholesky's decomposition

JEL Code: C38, C63, C88

Introduction

Multivariate statistical techniques are often applied to more complex datasets containing numerous variables and a larger number of observations. Scientists with focus on multivariate methods often find themselves needing datasets to evaluate quality of a newly proposed statistical methods and its results on relevant datasets, for example, when introducing a new similarity measure, see (Morlini & Zani, 2012), or when doing survey on existing similarity/distance measures, see (Alamuri, et al. 2014). Despite of numerous dataset repositories on-line, real datasets with desired features are still often hard to obtain. The situation is even more complicated, when a lot of datasets with a given specific set of features is needed in order to make the results of the analysis reliable.

The lack of multivariate data generators suitable for cluster analysis was the main motivation for this contribution. To authors' knowledge, there is no algorithm implemented in the R language and environment for statistical computing, see (R Core Team, 2018), that generates datasets suitable for cluster analysis. There is one function for generating categorical datasets in R package *overflow*, see (Campbell et al., 2013). However, its settings are very limited – it is possible to choose from only limited number of categorical variables and it is useful for only several very specific purposes. Matějka et al. (2016) presented more complex, more usable and more sophisticated algorithm for generating mixed-type datasets containing a required structure. This algorithm is based on generating random values based on multivariate normal distributions defined by covariance matrices and mean vectors. However, this algorithm is not actually included in the *nomclust* package, see (Šulc & Řezanková, 2015) as it was intended, hence it is not available to everyone who might be interested. Also, it generates samples only from multivariate normal distribution.

In this article, a simple method for generating multivariate datasets with given features is proposed. The newly proposed data generator is designed mainly for generating datasets appropriate for cluster analysis, especially (but not solely) for generating categorical datasets. Taking into consideration model-based clustering methods and their definition of a cluster, the generated dataset consists of a given number of clusters, where each cluster corresponds to one sample of a given multivariate distribution. The algorithm for generating samples from given multivariate distributions is inspired by the NORTA (NORmal-To-Anything) transformation in combination with Cholesky's decomposition. The NORTA transformation is widely used and it is one of the most popular methods for generating multivariate data, see (Chen, 2001; Henderson et al., 2000; Xiao, 2016).

In Section 1, the main idea and theoretical background of the algorithm is explained. The algorithm is inspired by model-based clustering logic and uses the NORTA transformation and Cholesky's decomposition. Section 2 focuses on the *gen.cd()* function, that follows algorithms and methods explained in the first chapter, and its implementation in the R language. Last chapter provides a brief illustration of *gen.cd()* function application and presents generated datasets with given features.

1 Theoretical background

As mentioned before, each generated dataset is a mixture of several samples from given multivariate distributions. This idea follows the logic of finite mixture models from model-based clustering. Finite mixture models assume that the population is made up of several distinct clusters, each following a different multivariate probability density distribution, see (Stahl & Sallis, 2012). Therefore, the problem of generating dataset with given features can actually be reduced to generating samples from given multivariate distributions.

We use the following NORTA algorithm, inspired by (Cario & Nelson, 1997), to generate samples from a given multivariate distribution, therefore to generate $\mathbf{x}_i$ with cumulative distribution functions F_i for $i = 1,2,\dots,n$:

1. Generate a multivariate standard-normal random vector $\mathbf{Z} = (\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_n)$, such that $Corr(\mathbf{z}_i, \mathbf{z}_j) = \rho^*_{ij}$, for $1 \le i, j \le n$.
2. Compute $\mathbf{u}_i = \Phi(\mathbf{z}_i)$ for $i = 1, 2, \dots, n$, where $\Phi(\cdot)$ is standard-normal cumulative distribution function.
3. Compute $\mathbf{x}_i = F_i^{-1}(\mathbf{u}_i)$ for $i = 1, 2, \dots, n$, where F_i^{-1} is the inverse of given marginal cumulative distribution functions F_i.

In the first step of the NORTA algorithm, Cholesky's decomposition is used. The Cholesky matrix transforms a vector of uncorrelated normally-distributed random variables into a vector of correlated normally-distributed random variables, see (Higham, 2009). Suppose, that $(\rho_{ij})_{i,j=1}^n$ is a given correlation matrix. Firstly, positive definiteness of the matrix is verified. If necessary, the matrix is adjusted to be the closest positive definite correlation matrix $(\rho^*_{ij})_{i,j=1}^n$. Then the correlated random vectors are generated using Cholesky's decomposition following this algorithm:

1. Let $\mathbf{Z}^{ind} = (\mathbf{z}_1^{ind}, \mathbf{z}_2^{ind}, \dots, \mathbf{z}_n^{ind})$ be an independent standard-normal random vector.
2. Let $\mathbf{C}$ be the Cholesky decomposition of the correlation matrix $(\rho^*_{ij})_{i,j=1}^n$, such that $\mathbf{C}^T\mathbf{C} = (\rho^*_{ij})_{i,j=1}^n$.
3. Then, $\mathbf{Z} = \mathbf{C}^T\mathbf{Z}^{ind}$ has multivariate standard-normal distribution with correlation matrix $(\rho^*_{ij})_{i,j=1}^n$.

Assuming each cluster in the dataset is generated from a given multivariate distribution, the dataset is a mixture of several samples obtained by algorithms described in this chapter. Since the dataset should consist of categorical data, numeric dataset is discretized using function *discretize()* in order to create nominal variables.

2 Implementation in R

The algorithms described in Section 1 are implemented in the R language and environment for statistical computing as function *gen.cd().*This function reads input arguments, processes them and generates a dataset with desired features as matrix or csv. Table 1 provides summary of input arguments of the function.

Tab. 1: Input arguments for the gen.cd() function

Argument	Type	Dimensions	Description
n.clu	Integer	1×1	Number of clusters.
n.var	Integer	1×1	Number of variables in dataset.
var.names	Vector	$\mathrm{n.var} \times 1$	Variables' names (optional).
cat	Vector	$\mathrm{n.var} \times 1$	Number of categories for each variable.
N	Vector	$\mathrm{n.clu} \times 1$	Number of observations in each cluster.
C	Array	$\mathrm{n.var} \times \mathrm{n.var} \times \mathrm{n.clu}$	Correlation matrices for each cluster (optional).
C.strength	Vector	$\mathrm{n.clu} \times 1$	It indicates “weak”, “medium” or “strong” correlation in randomly generated correlation matrices for each cluster. (When argument C is NULL.)
distrib	Array	$3 \times 1 \times \mathrm{n.clu}$	It specifies distribution (“binom”, “nbinom”, “pois”, “chisq”, “t”, “geom”, “weibull”, “alt”) and its parameters for each cluster.
save.csv	Logical	NULL	It indicates, whether the dataset should be saved as csv or not.

Source: The authors

The function requires specifying the number of clusters and the number of variables in the dataset, the number of categories of each variable, the number of observations in each cluster, the distribution of each cluster and its parameters as input attributes. The user can decide whether he wants to specify variables’ names, whether he wants to save the generated dataset in csv format or how he wants to assign correlation matrices for each cluster – user can enter an array of correlation matrices (in such case each given correlation matrix is adjusted to be the closest positive definite correlation matrix) or he can just indicate the strength of the correlation.

Following algorithms from Section 1, given number of samples from given multivariate distributions with given features are generated. The output of the function is a random permutation of all the samples combined, where each variable is discretized to desired number of categories at the very end. Moreover, the last column of the generated dataset represents corresponding multivariate distribution of each observation. This information can be very useful in cluster analysis, where “true” cluster structure is usually unknown.

The gen.cd() function requires package *Matrix*, see (Bates & Maechler, 2018), and package *arules*, see (Hahsler, 2018).

3 Illustrative examples

The gen.cd() function and its application is presented in two illustrative examples in this section. In the first example (Example 1), the output of the function should be a dataset with 130 observations, four categorical variables and the number of categories should be equal to 9 for each variable. The dataset should be a mixture of three samples, where the number of observations in the sample is equal to 50, 20 and 60 observations and the samples are generated from the following distributions – binom(10, 0.5), nbinom(5, 0.77), binom(20,0.83). There are no specific correlation matrices given, however, the observations in the first and the second sample are required to be strongly correlated and observations in the third sample are required to be moderately correlated. It is also required, that the generated dataset is saved as csv. Hence, the input of function gen.cd() with given input arguments looks like this:

```
gen.cd(n.clu=3, n.var=4, cat=c(9,9,9,9), N=c(50,20,60),C.strength=c("high",
"high", "medium"), distrib=array(c("binom", 10, 0.5, "nbinom", 5, 0.77,
"binom", 20, 0.83), dim=c(3, 1, 3)), save.csv=TRUE)
```

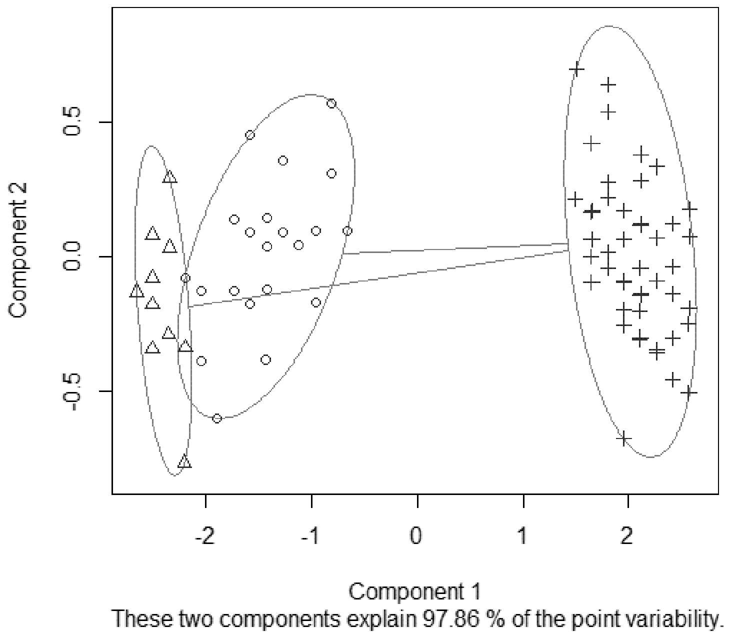
This command generates the dataset with given features and saves the output as csv. Figure 1 shows the structure of the dataset using principal component analysis, where all the variables are treated as ordinal ones, in order to provide a data visualization in two dimensions.

The second illustrative example (Example 2) of the *gen.cd()* function application is generating a dataset with 50 observations and four binary variables with given names indicating a status of company’s campaing – “isFunded”, “isProfitable”, “isInternational”, “isActive”. The dataset should be a mixture of two samples, where the number of observations in the first sample is 20 and the number of observations in the second sample is 30. The samples are generated from the following distributions – Bernoulli(0.21), Bernoulli(0.85) and correlation matrices are given. Then, function gen.cd() with given input arguments looks like this:

```
gen.cd(n.clu=2, n.var=4, var.names=c("isFunded", "isProfitable",
"isInternational", "isActive"),cat=c(2,2,2,2), N=c(20,30),
C.strength= array(c(c(1.00, -0.73, -0.63, -0.78,-0.73,  1.00, -0.65,
0.60,-0.63, -0.65,  1.00, -0.75,-0.78,  0.60, -0.75,  1.00),  c( 1.00, -
0.54,  0.41, -0.49,-0.54,  1.00,  0.48, -0.42,0.41,  0.48,  1.00, -0.38,-
0.49, -0.42, -0.38,  1.00)), dim=c(4,4,2)), distrib=array(c("alt", 0.21,
NA, "alt", 0.85, NA), dim=c(3, 1, 3)))
```

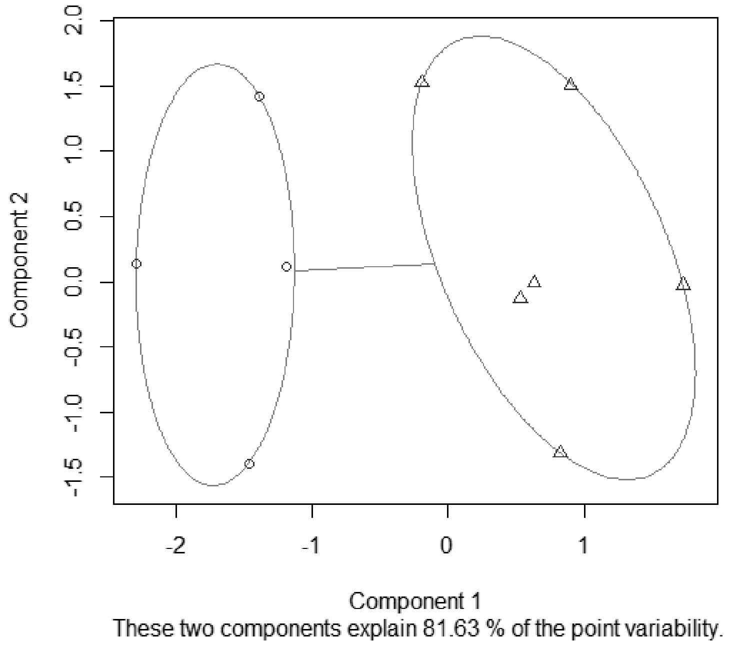
This command generates the dataset with desired features. Figure 2 shows the structure of this generated dataset using principal component analysis so visualization of the generated dataset in two dimensions is possible.

Fig. 1: Visualization of the dataset's structure for Example 1



Source: The authors

Fig. 2: Visualization of the dataset's structure for Example 2



Source: The authors

Conclusion

This contribution presented a simple algorithm for generating categorical datasets with given set of features, that is suitable for cluster analysis. The main idea of this categorical data generator follows the philosophy of finite mixture models from model-based clustering, where it is assumed that a population is made up of several distinct clusters, each following a different multivariate distribution. Therefore, the generated dataset is a mixture of several samples from given multivariate distributions.

This method for generating multivariate datasets was proposed mainly for the usage in cluster analysis. However, it can be used in wide range of application, where the multivariate datasets with given specific features are needed.

Additional function, that would simplify and automatize the process of generating a large number of multivariate datasets, will be implemented in the future. This will make it possible to generate large number of multivariate datasets with desired features. This could be especially useful when numerous datasets are needed, for example for quality evaluation of a newly proposed statistical methods or for studying a statistical method and its performance when applied to specific data.

Acknowledgment

This work was supported by the University of Economics, Prague under Grant IGA F4/44/2018.

References

Alamuri, M., Surampudi, B. R. & Negi, A. (2014). A survey of distance/similarity measures for categorical data. *2014 International Joint Conference on Neural Networks (IJCNN).* doi:10.1109/ijcnn.2014.6889941.

Bates, D. & Maechler, M. (2018). Matrix: Sparse and Dense Matrix Classes and Methods. R package version 1.2-14. https://CRAN.R-project.org/package=Matrix.

Cario, M. & Nelson, B. (1997). Modeling and generating random vectors with arbitrary marginal distributions and correlation matrix. Technical Report, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL.

Campbell, S., Mahto, A., Barnier, J. & Rinker T. (2013) overflow: Utility functions for answering R question on StackOverflow, R package version 0.2-1. http://github.com/sebastian-c/overflow.

Chen, H. (2001) Initialization for NORTA: Generation of Random Vectors with Specified Marginals and Correlations. *INFORMS Journal on Computing* 13(4), 312-331. http://dx.doi.org/10.1287/ijoc.13.4.312.9736.

Hahsler, M., Buchta, C., Gruen, B. & Hornik, K. (2018). arules: Mining Association Rules and Frequent Itemsets. R package version 1.6-1. https://CRAN.R-project.org/package=arules.

Henderson, S., Chiera, B. & Cooke, R. (2000). Generating "dependent" quasi-random numbers. *2000 Winter Simulation Conference Proceedings (Cat. No.00CH37165),* 527-536. doi:10.1109/wsc.2000.899760.

Higham, N. J. (2009). Cholesky factorization. *Wiley Interdisciplinary Reviews: Computational Statistics*, 1(2), 251-254. doi:10.1002/wics.18.

Matějka, M., Procházka, J. & Šulc, Z. (2016). Mixed data generator. *In The 10th International Days of Statistics and Economics.* Slaný: Melandrium, 1581-1590. https://msed.vse.cz/msed_2016/article/65-Matejka-Martin-paper.pdf.

Morlini, I. & Zani, S. (2012). A new class of weighted similarity indices using polytomous variables. *In Journal of Classification*, 29(2), 199-226.

R Core Team (2018). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Šulc, Z. & Řezanková, H. (2015). Nomclust: An r package for hierarchical clustering of objects characterized by nominal variables. In *The 9th International Days of Statistics and Economics.* Slaný: Melandrium, 1581-1590. https://msed.vse.cz/msed_2015/article/48-Sulc-Zdenek-paper.pdf.

Stahl, D. & Sallis, H. (2012). Model-based cluster analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 4(4), 341-358. doi:10.1002/wics.1204.

Xiao, Q. (2016). Generating correlated random vector involving discrete variables. *Communications in Statistics – Theory and Methods*, 46(4), 1594-1605. doi:10.1080/03610926.2015.1024860.

Contact

Jana Cibulková
University of Economics, Prague
W. Churchill Sq. 4, 130 67 Prague 3, Czech Republic
jana.cibulkova@vse.cz

Hana Řezanková
University of Economics, Prague
W. Churchill Sq. 4, 130 67 Prague 3, Czech Republic
hana.rezankova@vse.cz