Generator of unsupervised semi-artificial data

Marko Robnik-Šikonja

University of Ljubljana, Faculty of Computer and Information Science,
Večna pot 113, 1000 Ljubljana, Slovenia Marko.Robnik@fri.uni-lj.si

In many important application areas addressed by discovery science, there just isn't enough data available. There are several reasons for this, the data may be inherently scarce, difficult to obtain, expensive, or the distribution of the events of interests is highly imbalanced. This causes problems in model selection, reliable performance estimation, development of specialized algorithms, and tuning of learning model parameters.

Recently we presented a semi-artificial data generator limited to classification problems [3]. This generator first constructs a RBF network prediction model which consists of Gaussian kernels. The kernels estimate the probability density function from the training instances. Due to properties of Gaussian kernels, the learned kernels are used in a generative mode to produce new data. This approach was successfully used for a variety of data sets described with different number of attributes of both, numerical and categorical type. The generator was successfully tested in development of big data tools [1] and is freely available as R package semiArtificial. We expect such a tool to be useful in the data mining algorithm development and adaptation of data analytics algorithms to specifics of data sets. Possible other uses are data randomization to ensure privacy, simulations requiring large amounts of data, testing of big data tools, benchmarking, and scenarios with huge amounts of data.

The problem as well as the strength of the RBF-based generator is that the learned model tries to discriminate between instances with different class values, therefore the approach generates many kernels for each class. The widths of the kernels are estimated from the training instances which activate the particular kernel.

A simple demonstration of the generated data on a simple data set is presented in Fig. 1 The data set forms a two dimensional grid where attributes A_1 and A_2 are generated with three Gaussian kernels with centers at $(-5, -5)$, $(0, 0)$, and $(5, 5)$. Each group of 500 instances is assigned a unique class value (red, blue, and green, respectively) as illustrated in Fig 1a. The RBF generator based on this data consists of eight Gaussian kernels (two for red and blue class each, and four for green class). We illustrate 1500 instances generated with this generator in Fig 1b. As the RBF learner did not find the exact locations of the original centers it approximated the data with several kernels, so there is some difference between the original and generated data.

In this work we propose a different approach using density trees [2]. The density trees are similar to decision trees with a difference that there is no designated class variable. The split criteria used therefore try to separate areas with different density. We experimented with different types of density trees varying the split selection criteria, stopping criteria, and also the density estimation method used in the leaves of the trees. Our preliminary experiments show that in general the density tree approach produces semi-artificial data with better properties compared to the RBF generator (improved similarity to the original, better clustering performance).

The idea of proposed density trees data generator is to construct a forest of density trees using bootstrap sampling and random selection of a subset of attributes in each node (similarly to random forests) to assure tree diversity. When generating a new instance we start at the root of a randomly chosen tree. The splitting attribute in each interior node is used to generate the value of the new instance randomly but following the selected attribute's empirical cumulative distribution function (ecdf). Based on the generated value we recursively repeat the generation of values for yet unobserved attributes by following the left- or right-hand branch of the tree. Arriving to the leaf of the tree we assume that dependencies between attributes were resolved on the path from the the root to the leaf and we generate the remaining values of the attributes using univariate methods (kernel density estimation, log splines, or ecdf).

The working of the generator are demonstrated in Fig. 1. The generator using density trees was not using the class information, but was able to better capture locality information of the instances (Fig. 1c). If class information was provided the density tree based generator was using it as any other nominal attribute (Fig. 1d). The results on this and many other data sets show that the generated data is more similar to the original data set.

In further work we will investigate exact conditions when each of the proposed methods (RBF generator, density tree generator) produces favorable results.

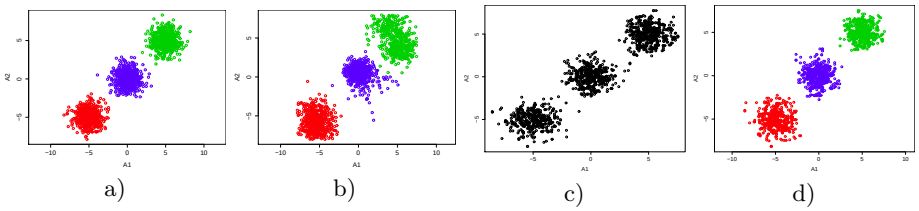

Fig. 1. An illustration of the generated data on the two dimensional dataset: a) the original data, b) data generated with RBF generator, c) data generated with density trees without class information, d) data generated with density trees with class information as a nominal attribute.

Acknowledgments

The author was supported by the Slovenian Research Agency (ARRS) through research programme P2-0209 and European Commission through the Human Brain Project (grant number 604102).

Bibliography

[1] J. Kranjc, R. Orač, V. Podpečan, M. Robnik-Šikonja, and N. Lavrač. ClowdFlows: Workows for big data on the cloud. Technical report, Jožef Stefan Institute, Ljubljana, Slovenia, 2014.

[2] P. Ram and A. G. Gray. Density estimation trees. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD*, pages 627–635. ACM, 2011.

[3] M. Robnik-Šikonja. Data generator based on RBF network. Technical report, University of Ljubljana, Faculty of Computer and Information Science, 2014. URL http://arxiv.org/abs/1403.7308v1.