AutoClassWrapper: a Python wrapper for AutoClass C classification

Jean-Michel Camadro\textsuperscript{1} and Pierre Poulain\textsuperscript{1}

\textsuperscript{1}Mitochondria, Metals and Oxidative Stress group, Institut Jacques Monod, UMR 7592, Univ. Paris Diderot, CNRS, Sorbonne Paris Cité, France.

Background

Data clustering or classification is widely used in many scientific fields and is often the very first step into exploratory data analysis. In 1996, the Ames Research Center at NASA developed AutoClass (P. Cheeseman & Stutz, 1996; Stutz & Cheeseman, 1996), an unsupervised Bayesian classification system, and its implementation in C, AutoClass C. The purpose of the Bayesian methodology is to find the classification that fits the data with the highest probability. The AutoClass algorithm is very efficient, notably in its ability to automatically find the best number of classes (or clusters) and to handle missing data. AutoClass can handle both real (numerical) and discrete values. It has been successful in classifying data as diverse as infrared spectra of stars (Goebel et al., 1989), protein structures (Hunter & States, 1992), introns from human DNA sequences (P. Cheeseman & Stutz, 1996), Landsat satellites images (P. Cheeseman & Stutz, 1996), body pattern in the common cuttlefish (Crook, Baddeley, & Osorio, 2002), patterns between rich and poor countries (Ardić, 2006), network traffic (Erman, Arlitt, & Mahanti, 2006), or catchments in the Australian landscape (Angus Webb et al., 2007). In proteomics and genomics, where thousands of proteins or genes are detected at once, the need for data classification is even more crucial. To this aim, AutoClass@IJM (Achcar, Camadro, & Mestivier, 2009), a web server that utilizes AutoClass C, has made Bayesian classification more accessible (see for instance results from (Elliott, Tanaka, Schwark, & Andrade, 2018; Léger, Garcia, Ounissi, Lelandais, & Camadro, 2015; Simpson et al., 2011)).

Albeit its proven efficiency and versatility, AutoClass C is not easy to configure and run locally for the end user. As far as we know of, there is only an R wrapper developed by M. Spivakov but with a limited interface.

Overview

The AutoClassWrapper library is a Python wrapper for AutoClass C. It aims to ease the usage of AutoClass C and offers:

- Data preparation. User input data must be formatted as tab-separated values (TSV) files. They are quality checked with the pandas library (McKinney, 2010) and converted into suitable parameter files for AutoClass C.

- Results extraction. AutoClass C output files are processed into more usable formats, such as clustered data (CDT) file for further visualization in Java Treeview (Saldanha, 2004) or TSV file for analysis with R, Python or any spreadsheet software.
For all classes, descriptive statistics of numerical features are produced. An additional hierarchical clustering is performed on output classes and provides, through a dendrogram, a convenient way to assess proximity between classes.

AutoClassWrapper has been implemented with good practices in software development in mind (Jiménez et al., 2017; Taschuk & Wilson, 2017):

- version control repository on GitHub (https://github.com/pierrepo/autoclasswrapper),
- open-source license (BSD-3-Clause),
- continuous integration through tests,

AutoClassWrapper is available in the Python Package Index (PyPI). All versions of the software are archived in the Zenodo repository (https://doi.org/10.5281/zenodo.2527058) and in the Software Heritage archive (https://archive.softwareheritage.org/swh:1:dir:

References


