AutoClassWrapper: a Python wrapper for AutoClass C classification

Jean-Michel Camadro, Pierre Poulain

The Journal of Open Source Software

AutoClassWrapper: a Python wrapper for AutoClass C classification

Jean-Michel Camadro[1] **and Pierre Poulain**[1]

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

DOI: 10.21105/joss.01390

Software

- Review
- Repository
- Archive

Submitted: 08 March 2019
Published: 25 July 2019

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 (Ardiç, 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,
- and documentation (https://autoclasswrapper.readthedocs.io/en/latest/).

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:330c16e8e3d999674f490135015d04de1f08e48a/).

References

Achcar, F., Camadro, J.-M., & Mestivier, D. (2009). AutoClass@IJM: A powerful tool for Bayesian classification of heterogeneous data in biology. *Nucleic Acids Research*, *37*(Web Server), W63–W67. doi:10.1093/nar/gkp430

Angus Webb, J., R. Bond, N., R. Wealands, S., Mac Nally, R., P. Quinn, G., A. Vesk, P., & R. Grace, M. (2007). Bayesian clustering with AutoClass explicitly recognises uncertainties in landscape classification. *Ecography*, *30*(4), 526–536. doi:10.1111/j.2007.0906-7590.05002.x

Ardiç, O. P. (2006). The gap between the rich and the poor: Patterns of heterogeneity in the cross-country data. *Economic Modelling*, *23*(3), 538–555. doi:10.1016/j.econmod.2006.02.006

Cheeseman, P., & Stutz, J. (1996). Bayesian Classification(AutoClass):Theory and Results. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, & R. Uthurusamy (Eds.), *Advances in Knowledge Discovery and Data Mining* (pp. 153–180). Boston, MA: AAAI/MIT Press.

Crook, A. C., Baddeley, R., & Osorio, D. (2002). Identifying the structure in cuttlefish visual signals. (R. A. Johnstone & S. R. X. Dall, Eds.)*Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, *357*(1427), 1617–1624. doi:10.1098/rstb.2002.1070

Elliott, M. C., Tanaka, P. M., Schwark, R. W., & Andrade, R. (2018). Serotonin Differentially Regulates L5 Pyramidal Cell Classes of the Medial Prefrontal Cortex in Rats and Mice. *eneuro*, *5*(1), ENEURO.0305–17.2018. doi:10.1523/ENEURO.0305-17.2018

Erman, J., Arlitt, M., & Mahanti, A. (2006). Traffic classification using clustering algorithms. In *Proceedings of the 2006 SIGCOMM workshop on Mining network data - MineNet '06* (pp. 281–286). Pisa, Italy: ACM Press. doi:10.1145/1162678.1162679

Goebel, J., Volk, K., Walker, H., Gerbault, F., Cheeseman, P., Self, M., Stutz, J., et al. (1989). A Bayesian classification of the IRAS LRS Atlas. *Astronomy and Astrophysics*, *222*, L5–L8.

Hunter, L., & States, D. (1992). Bayesian classification of protein structure. *IEEE Expert*, *7*(4), 67–75. doi:10.1109/64.153466

Jiménez, R. C., Kuzak, M., Alhamdoosh, M., Barker, M., Batut, B., Borg, M., Capella-Gutierrez, S., et al. (2017). Four simple recommendations to encourage best practices in research software. *F1000Research*, *6*, 876. doi:10.12688/f1000research.11407.1

Léger, T., Garcia, C., Ounissi, M., Lelandais, G., & Camadro, J.-M. (2015). The Metacaspase (Mca1p) has a Dual Role in Farnesol-induced Apoptosis in *Candida Albicans*. *Molecular & Cellular Proteomics*, *14*(1), 93–108. doi:10.1074/mcp.M114.041210

McKinney, W. (2010). Data Structures for Statistical Computing in Python. In S. van der Walt & J. Millman (Eds.), *Proceedings of the 9th Python in Science Conference* (pp. 51–56).

Saldanha, A. J. (2004). Java Treeview–extensible visualization of microarray data. *Bioinformatics*, *20*(17), 3246–3248. doi:10.1093/bioinformatics/bth349

Simpson, S. D., Jennings, S., Johnson, M. P., Blanchard, J. L., Schön, P.-J., Sims, D. W., & Genner, M. J. (2011). Continental Shelf-Wide Response of a Fish Assemblage to Rapid Warming of the Sea. *Current Biology*, *21*(18), 1565–1570. doi:10.1016/j.cub.2011.08.016

Stutz, J., & Cheeseman, P. (1996). Autoclass - A Bayesian Approach to Classification. In J. Skilling & S. Sibisi (Eds.), *Maximum Entropy and Bayesian Methods*, The Fundamental Theories of Physics: Their Clarification, Development and Application. Dordrecht, The Netherlands: Kluwer Academic Publishers. doi:https://doi.org/10.1007/978-94-009-0107-0_13

Taschuk, M., & Wilson, G. (2017). Ten simple rules for making research software more robust. *PLOS Computational Biology*, *13*(4), e1005412. doi:10.1371/journal.pcbi.1005412