

PyClustering: Data Mining Library

Andrei V. Novikov¹

¹ Independent Researcher

DOI: [10.21105/joss.01230](https://doi.org/10.21105/joss.01230)

Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Submitted: 22 January 2019

Published: 14 April 2019

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC-BY](#)).

Introduction

A variety of scientific and industrial sectors continue to experience exponential growth in their data volumes, and so automatic categorization techniques have become standard tools for dataset exploration. Automatic categorization techniques – typically referred to as clustering – help expose the structure of a dataset. For example, the generated clusters might each correspond to a customer group with reasonably similar needs and behavior. Because the resulting clusters are often used as building blocks for higher-level – often custom – predictive models, researchers have continually tweaked and invented new clustering techniques. PyClustering is an open source data mining library written in Python and C++ that provides a wide range of clustering algorithms and methods, including bio-inspired oscillatory networks. PyClustering is mostly focused on cluster analysis to make it more accessible and understandable for users.

Summary

The PyClustering library is a Python and C++ data mining library focused on cluster analysis. By default, the C++ part of the library is used for processing in order to achieve maximum performance. This is especially relevant for algorithms that are based on oscillatory networks, whose dynamics are governed by a system of differential equations. If support for a C++ compiler is not detected, PyClustering falls back to pure Python implementations of all kernels. In order to increase the performance of the Python implementations, PyClustering makes use of the NumPy (Oliphant, 2006) library for its array manipulations.

PyClustering provides optimized, parallel C++14 clustering implementations; on most platforms, threading is provided by `std::thread`, though the Parallel Patterns Library is used for Windows. Due to the standardization of these threading libraries, PyClustering is simple to integrate into pre-existing projects.

The core Python dependencies of PyClustering are NumPy and SciPy (Jones, Oliphant, Peterson, et al., 2019), and Matplotlib (Hunter, 2007) and Pillow are required for visualization support. The visualization functionality includes 2D and 3D plots of the cluster embeddings, image segments, and, in the case of oscillatory networks, graphs of the synchronization processes.

The PyClustering library is available on PyPi and from a github repository. Since the first release on PyPi in 2014, it has been downloaded more than 141.000 times. The quality of the library is supported by static and dynamic analyzers, such as `cppcheck`, `scan-build`, and `valgrind` (Nethercote & Seward, 2007). More than 93% code coverage is provided by more than 2200 unit and integration tests. Each commit to the repository triggers building, analysis, and testing on CI services such as `travis-ci` or `appveyor`. PyClustering

provides fully-documented code for each library version, including examples, math and algorithms description, and installation instructions.

Clustering Algorithms

Algorithms and methods are located in the Python module `pyclustering.cluster` and in the C++ namespace `ccore::clst`.

Algorithm	Python	C++
Agglomerative (Jain & Dubes, 1988)	✓	✓
BANG (Schikuta & Erhart, 1998)	✓	
BIRCH (Zhang, Ramakrishnan, & Livny, 1996)	✓	
BSAS (Theodoridis & Koutroumbas, 2009)	✓	✓
CLARANS (Ng & Han, 2002)	✓	
CLIQUE (Agrawal, Gunopulos, & Raghavan, 2005)	✓	✓
CURE (Guha, Rastogi, & Shim, 1998)	✓	✓
DBSCAN (Ester, Kriegel, Sander, & Xu, 1996)	✓	✓
Elbow (Thorndike, 1953)	✓	✓
EMA (Gupta & Chen, 2011)	✓	
GA - Genetic Algorithm (Cowgill, Harvey, & Watson, 1999)	✓	✓
HSyncNet (Shao, He, Böhm, Yang, & Plant, 2013)	✓	✓
K-Means (Macqueen, 1967)	✓	✓
K-Means++ (Arthur & Vassilvitskii, 2007)	✓	✓
K-Medians (Jain & Dubes, 1988)	✓	✓
K-Medoids (Jain & Dubes, 1988)	✓	✓
MBSAS (Theodoridis & Koutroumbas, 2009)	✓	✓
OPTICS (Ankerst, Breunig, Kriegel, & Sander, 1999)	✓	✓
ROCK (Guha, Rastogi, & Shim, 1999)	✓	✓
Silhouette (Rousseeuw, 1987)	✓	
SOM-SC (Kohonen, 1990)	✓	✓
SyncNet (A. V. Novikov & Benderskaya, 2014a)	✓	✓
Sync-SOM (A. V. Novikov & Benderskaya, 2014b)	✓	
TTSAS (Theodoridis & Koutroumbas, 2009)	✓	✓
X-Means (Pelleg & Moore, 2000)	✓	✓

Oscillatory Networks and Neural Networks

Networks are located in the Python module `pyclustering.nnet` and in the C++ namespace `ccore::nnet`.

Model	Python	C++
CNN - Chaotic Neural Network (Benderskaya & Zhukova, 2009)	✓	
fSync - Oscillatory network based on Landau-Stuart equation and Kuramoto model (Kuramoto, 2003)	✓	
HHN - Oscillatory network based on Hodgkin-Huxley model (Chik, Borisjuk, & Kazanovich, 2009)	✓	✓
Hysteresis Oscillatory Network (Jin'no, Taguchi, Yamamoto, & Hirose, 2003)	✓	

Model	Python	C++
LEGION - Local Excitatory Global Inhibitory Oscillatory Network (Wang & Terman, 1997)	✓	✓
PCNN - Pulse-Coupled Neural Network (Lindblad & Kinser, 2013)	✓	✓
SOM - Self-Organized Map (Kohonen, 1990)	✓	✓
Sync - Oscillatory network based on Kuramoto model (Arenas, Diaz-Guilera, Kurths, Moreno, & Zhou, 2008)	✓	✓
SyncPR - Oscillatory network for pattern recognition (Follmann, Macau, Rosa, & Piqueira, 2015)	✓	✓
SyncSegm - Oscillatory network for image segmentation (A. Novikov & Benderskaya, 2015)	✓	✓

Graph Coloring Algorithms

Algorithms are located in the Python module `pyclustering.gcolor`.

Algorithm	Python	C++
DSatur (Brelaz, 1979)	✓	
Hysteresis (Jin'no et al., 2003)	✓	
GColorSync (Wu, Jiao, Li, & Chen, 2011)	✓	

Containers

Containers are located in the Python module `pyclustering.container` and in the C++ namespace `ccore::container`.

Container	Python	C++
KD Tree (Samet, 1990)	✓	✓
CF Tree (Zhang et al., 1996)	✓	

References

- Agrawal, J., Rakeshand Gehrke, Gunopulos, D., & Raghavan, P. (2005). Automatic subspace clustering of high dimensional data. *Data Mining and Knowledge Discovery*, 11(1), 5–33. doi:[10.1007/s10618-005-1396-1](https://doi.org/10.1007/s10618-005-1396-1)
- Ankerst, M., Breunig, M. M., Kriegel, H.-P., & Sander, J. (1999). OPTICS: Ordering points to identify the clustering structure. *SIGMOD Rec.*, 28(2), 49–60. doi:[10.1145/304181.304187](https://doi.org/10.1145/304181.304187)
- Arenas, A., Diaz-Guilera, A., Kurths, J., Moreno, Y., & Zhou, C. (2008). Synchronization in complex networks. *Physics Reports*, 469(3), 93–153. doi:[10.1016/j.physrep.2008.09.002](https://doi.org/10.1016/j.physrep.2008.09.002)
- Arthur, D., & Vassilvitskii, S. (2007). K-means++: The advantages of careful seeding. In *In proceedings of the 18th annual acm-siam symposium on discrete algorithms*.
- Benderskaya, E. N., & Zhukova, S. V. (2009). Large-dimension image clustering by

- means of fragmentary synchronization in chaotic systems. *Pattern Recognition and Image Analysis*, 19(2), 306–314. doi:[10.1134/S1054661809020151](https://doi.org/10.1134/S1054661809020151)
- Brelaz, D. (1979). New methods to color the vertices of a graph. *Commun. ACM*, 22(4), 251–256. doi:[10.1145/359094.359101](https://doi.org/10.1145/359094.359101)
- Chik, D., Borisyuk, R., & Kazanovich, Y. (2009). Selective attention model with spiking elements. *Neural Networks*, 22(7), 890–900. doi:<https://doi.org/10.1016/j.neunet.2009.02.002>
- Cowgill, M., Harvey, R., & Watson, L. (1999). A genetic algorithm approach to cluster analysis, 37(7), 99–108. doi:[https://doi.org/10.1016/S0898-1221\(99\)00090-5](https://doi.org/10.1016/S0898-1221(99)00090-5)
- Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the second international conference on knowledge discovery and data mining, KDD'96* (pp. 226–231). Portland, Oregon: AAAI Press. Retrieved from <http://dl.acm.org/citation.cfm?id=3001460.3001507>
- Follmann, R., Macau, E. E. N., Rosa, E., & Piqueira, J. R. C. (2015). Phase oscillatory network and visual pattern recognition. *IEEE Transactions on Neural Networks and Learning Systems*, 26(7), 1539–1544. doi:[10.1109/TNNLS.2014.2345572](https://doi.org/10.1109/TNNLS.2014.2345572)
- Guha, S., Rastogi, R., & Shim, K. (1998). CURE: An efficient clustering algorithm for large databases. *SIGMOD Rec.*, 27(2), 73–84. doi:[10.1145/276305.276312](https://doi.org/10.1145/276305.276312)
- Guha, S., Rastogi, R., & Shim, K. (1999). ROCK: A robust clustering algorithm for categorical attributes. In *Proceedings of the 15th international conference on data engineering, ICDE '99* (pp. 512–521). Washington, DC, USA: IEEE Computer Society. doi:[10.1109/ICDE.1999.754967](https://doi.org/10.1109/ICDE.1999.754967)
- Gupta, M. R., & Chen, Y. (2011). Theory and use of the em algorithm. *Found. Trends Signal Process.*, 4(3), 223–296. doi:[10.1561/20000000034](https://doi.org/10.1561/20000000034)
- Jain, A. K., & Dubes, R. C. (1988). *Algorithms for clustering data*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc.
- Jin'no, K., Taguchi, H., Yamamoto, T., & Hirose, H. (2003). Dynamical hysteresis neural networks for graph coloring problem, 737–740. doi:[10.1109/ISCAS.2003.1206418](https://doi.org/10.1109/ISCAS.2003.1206418)
- Kohonen, T. (1990). The self-organizing map. In *Proceedings of the IEEE* (Vol. 78, pp. 1464–1480). doi:[10.1109/5.58325](https://doi.org/10.1109/5.58325)
- Kuramoto, Y. (2003). *Chemical oscillations, waves, and turbulence*. Chemistry series. Dover Publications.
- Lindblad, T., & Kinser, J. M. (2013). *Image processing using pulse-coupled neural networks* (3rd ed.). Berlin, Heidelberg: Springer-Verlag. doi:[10.1007/978-3-642-36877-6](https://doi.org/10.1007/978-3-642-36877-6)
- Macqueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *In 5-th berkeley symposium on mathematical statistics and probability* (pp. 281–297).
- Nethercote, N., & Seward, J. (2007). Valgrind: A framework for heavyweight dynamic binary instrumentation. *SIGPLAN Not.*, 42(6), 89–100. doi:[10.1145/1273442.1250746](https://doi.org/10.1145/1273442.1250746)
- Ng, R., & Han, J. (2002). CLARANS: A method for clustering objects for spatial data mining. *Knowledge and Data Engineering, IEEE Transactions on*, 14, 1003–1016. doi:[10.1109/TKDE.2002.1033770](https://doi.org/10.1109/TKDE.2002.1033770)
- Novikov, A. V., & Benderskaya, E. N. (2014a). Oscillatory neural networks based on the kuramoto model for cluster analysis. *Pattern Recognit. Image Anal.*, 24(3), 365–371. doi:[10.1134/S1054661814030146](https://doi.org/10.1134/S1054661814030146)

- Novikov, A. V., & Benderskaya, E. N. (2014b). SYNC-som double-layer oscillatory network for cluster analysis. In *Proceedings of the 3rd international conference on pattern recognition applications and methods, ICPRAM 2014* (pp. 305–309). Portugal: SCITEPRESS - Science; Technology Publications, Lda. doi:[10.5220/0004906703050309](https://doi.org/10.5220/0004906703050309)
- Novikov, A., & Benderskaya, E. (2015). Oscillatory network based on kuramoto model for image segmentation. In *Proceedings of the 13th international conference on parallel computing technologies - volume 9251* (pp. 210–221). New York, NY, USA: Springer-Verlag New York, Inc. doi:[10.1007/978-3-319-21909-7_20](https://doi.org/10.1007/978-3-319-21909-7_20)
- Pelleg, D., & Moore, A. W. (2000). X-means: Extending k-means with efficient estimation of the number of clusters. In *Proceedings of the seventeenth international conference on machine learning, ICML '00* (pp. 727–734). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. Retrieved from <http://dl.acm.org/citation.cfm?id=645529.657808>
- Rousseeuw, P. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.*, 20(1), 53–65. doi:[10.1016/0377-0427\(87\)90125-7](https://doi.org/10.1016/0377-0427(87)90125-7)
- Samet, H. (1990). *The design and analysis of spatial data structures*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc. doi:[https://doi.org/10.1016/0098-3004\(91\)90117-V](https://doi.org/10.1016/0098-3004(91)90117-V)
- Schikuta, E., & Erhart, M. (1998). BANG-clustering: A novel grid-clustering algorithm for huge data sets. In A. Amin, D. Dori, P. Pudil, & H. Freeman (Eds.), *Advances in pattern recognition* (pp. 867–874). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:[10.1007/BFb0033313](https://doi.org/10.1007/BFb0033313)
- Shao, J., He, X., Böhm, C., Yang, Q., & Plant, C. (2013). Synchronization-inspired partitioning and hierarchical clustering. *IEEE Transactions on Knowledge and Data Engineering*, 25(4), 893–905. doi:[10.1109/TKDE.2012.32](https://doi.org/10.1109/TKDE.2012.32)
- Theodoridis, S., & Koutroumbas, K. (2009). *Pattern recognition, fourth edition*. Elsevier.
- Thorndike, R. L. (1953). Who belongs in the family? *Psychometrika*, 18(4), 267–276. doi:[10.1007/BF02289263](https://doi.org/10.1007/BF02289263)
- Wang, D., & Terman, D. (1997). Image segmentation based on oscillatory correlation. *Neural Comput.*, 9(4), 805–836. doi:[10.1162/neco.1997.9.4.805](https://doi.org/10.1162/neco.1997.9.4.805)
- Wu, J., Jiao, L., Li, R., & Chen, W. (2011). Clustering dynamics of nonlinear oscillator network: Application to graph coloring problem. *Physica D: Nonlinear Phenomena*, 240(24), 1972–1978. doi:<https://doi.org/10.1016/j.physd.2011.09.010>
- Zhang, T., Ramakrishnan, R., & Livny, M. (1996). BIRCH: An efficient data clustering method for very large databases. *SIGMOD Rec.*, 25(2), 103–114. doi:[10.1145/235968.233324](https://doi.org/10.1145/235968.233324)