scikit-hubness: Hubness Reduction and Approximate Neighbor Search

Roman Feldbauer¹, Thomas Rattei¹, and Arthur Flexer²

¹ Division of Computational Systems Biology, Department of Microbiology and Ecosystem Science, University of Vienna, Althanstraße 14, 1090 Vienna, Austria
² Austrian Research Institute for Artificial Intelligence (OFAI), Freyung 6/6/7, 1010 Vienna, Austria

DOI: 10.21105/joss.01957

Software
- Review
- Repository
- Archive

Editor: Yuan Tang
Reviewers:
- @ryEllison
- @aozorahime

Submitted: 10 December 2019
Published: 17 January 2020

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

Summary

scikit-hubness is a Python package for efficient nearest neighbor search in high-dimensional spaces. Hubness is an aspect of the curse of dimensionality in nearest neighbor graphs. Specifically, it describes the increasing occurrence of hubs and antihubs with growing data dimensionality: Hubs are objects, that appear unexpectedly often among the nearest neighbors of others objects, while antihubs are never retrieved as neighbors. As a consequence, hubs may propagate their information (for example, class labels) too widely within the neighbor graph, while information from antihubs is depleted. These semantically distorted graphs can reduce learning performance in various tasks, such as classification (Radovanović, Nanopoulos, & Ivanović, 2010), clustering (Schnitzer & Flexer, 2015), or visualization (Flexer, 2015). Hubness is known to affect a variety of applied learning systems (Angiulli, 2018), causing—for instance—overrepresentation of certain songs in music recommendations (Flexer & Stevens, 2018), or improper transport mode detection (Feldbauer, Leodolter, Plant, & Flexer, 2018).

Multiple hubness reduction algorithms have been developed to mitigate these effects (Flexer & Schnitzer, 2013; Hara, Suzuki, Kobayashi, Fukumizu, & Radovanovic, 2016; Schnitzer, Flexer, Schedl, & Widmer, 2012). We compared these algorithms exhaustively in a recent survey (Feldbauer & Flexer, 2019), and developed approximate hubness reduction methods with linear time and memory complexity (Feldbauer et al., 2018).

Currently, there is a lack of fully-featured, up-to-date, user-friendly software dealing with hubness. Available packages miss critical features and have not been updated in years (“Hub-Miner,” 2015), or are not particularly user-friendly (“Hub-Toolbox,” 2019). In this paper we describe scikit-hubness, which provides powerful, readily available, and easy-to-use hubness-related methods:

- hubness analysis (“Is my data affected by hubness?”): Assess hubness with several measures, including k-occurrence skewness (Radovanović et al., 2010), and Robin-Hood index (Feldbauer et al., 2018).
- hubness reduction (“How can we improve neighbor retrieval in high dimensions?”): Mutual proximity, local scaling, and DisSimLocal are currently supported, as they performed best in our survey. Exact methods as well as their approximations are available.
- approximate neighbor search (“Does it work for large data sets?”): Several methods are currently available, including locality-sensitive hashing (Aumüller, Christiani, Pagh, & Vesterli, 2019) and hierarchical navigable small-world graphs (Malkov & Yashunin, 2018).
scikit-hubness builds upon the SciPy stack (Virtanen, Gommers, Oliphant, Haberland, & others, 2019) and is integrated into the scikit-learn environment (Pedregosa, Varoquaux, Gramfort, Michel, & others, 2011), enabling rapid adoption by Python-based machine learning researchers and practitioners. Convenient interfaces to hubness-reduced neighbors-based learning are available in the skhubness.neighbors subpackage. It acts as a drop-in replacement for sklearn.neighbors, featuring all its functionality, and adding support for hubness reduction, where applicable. This includes, for example, the supervised KNeighborsClassifier and RadiusNeighborsRegressor, NearestNeighbors for unsupervised learning, and the general kneighbors_graph.

scikit-hubness is developed using several quality assessment tools and principles, such as PEP8 compliance, unit tests with high code coverage, continuous integration on all major platforms (Linux, MacOS, Windows), and additional checks by LGTM. The source code is available at https://github.com/VarIr/scikit-hubness under the BSD 3-clause license. The online documentation is available at https://scikit-hubness.readthedocs.io/. Install from the Python package index with $ pip install scikit-hubness.

Outlook

Future plans include adaption to significant changes of sklearn.neighbors introduced in version 0.22 in December 2019: The KNeighborsTransformer and RadiusNeighborsTransformer transform data into sparse neighbor graphs, which can subsequently be used as input to other estimators. Hubness reduction and approximate search can then be implemented as Transformers. This provides the means to turn skhubness.neighbors from a drop-in replacement of sklearn.neighbors into a scikit-learn plugin, which will (1) accelerate development, (2) simplify addition of new hubness reduction and approximate search methods, and (3) facilitate more flexible usage.

Acknowledgements

We thank Silvan David Peter for testing the software. This research is supported by the Austrian Science Fund (FWF): P27703 and P31988

References


