The Journal of Open Source Software

SISSO++: A C++ Implementation of the (Sure-Independence Screening and Sparsifying Operator Approach

Thomas A. R. Purcell[1], **Matthias Scheffler**[1], **Christian Carbogno**[1], **and Luca M. Ghiringhelli**[1]

1 NOMAD Laboratory at the Fritz Haber Institute of the Max Planck Society and Humboldt University, Berlin, Germany

DOI: 10.21105/joss.03960

Software

- Review
- Repository
- Archive

Editor: Jarvist Moore Frost

Reviewers:

- @ml-evs
- @AstroBarker

Submitted: 01 October 2021
Published: 09 December 2021

Summary

The sure independence screening and sparsifying operator (SISSO) approach (Ouyang et al., 2018) is an algorithm belonging to the field of artificial intelligence and more specifically a combination of symbolic regression and compressed sensing. As a symbolic regression method, SISSO is used to identify mathematical functions, i.e. the descriptors, that best predict the target property of a data set. Furthermore, the compressed sensing aspect of SISSO, allows it to find sparse linear models using tens to thousands of data points. SISSO is introduced for both regression and classification tasks. In practice, SISSO first constructs a large and exhaustive feature space of trillions of potential descriptors by taking in a set of user-provided *primary features* as a dataframe, and then iteratively applying a set of unary and binary operators, e.g. addition, multiplication, exponentiation, and squaring, according to a user-defined specificaiton. From this exhaustive pool of candidate descriptors, the ones most correlated to a target property are identified via sure-independence screening, from which the low-dimensional linear models with the lowest error are found via an ℓ_0 regularization.

Because symbolic regression generates an interpretable equation, it has become an increasingly popular concept across scientific disciplines (Wang et al., 2019), (Neumann et al., 2020), (Udrescu & Tegmark, 2020). A particular advantage of these approaches are their capability to model complex phenomena using relatively simple descriptors. SISSO has been used successfully in the past to model, explore, and predict important material properties, including the stability of different phases (Bartel et al., 2018), and (Schleder et al., 2020); the catalytic activity and reactivity (Han et al., 2021), (W. Xu et al., 2021), (Andersen et al., 2019), and (Andersen & Reuter, 2021); and glass transition temperatures (Pilania et al., 2019). Beyond regression problems, SISSO has also been used successfully to classify materials into different crystal prototypes (Ouyang et al., 2019), or whether a material crystallizes in its ground state as a perovskite (Bartel et al., 2019), or to determine whether a material is a topological insulator or not (Cao et al., 2020).

The SISSO++ package is an open-source (Apache-2.0 licence), modular, and extensible C++ implementation of the SISSO method with Python bindings. Specifically, SISSO++ applies this methodology for regression, log regression, and classification problems. Additionally, the library includes multiple Python functions to facilitate the post-processing, analyzing, and visualizing of the resulting models.

The Journal of Open Source Software

Statement of need

The main goal of the SISSO++ package is to provide a user-friendly, easily extendable version of the SISSO method for the scientific community. While both a FORTRAN (Ouyang, n.d.) and a Matlab (Gasper, n.d.) implimentation of SISSO exist, their lack of native Python interfaces led to the development of multiple seperate Python wrappers (C. Xu, n.d.), (Waroquiers, n.d.). This package looks to rectify this siutation by providing an implimentation that has native Python bindings and can be used both in a massively parallel environment for discovering the descriptors and on personal computing devices for analyzing and visualizing the results. For this reason, all computationally intensive task are written in C++ and support parallelization via MPI and OpenMP. Additionally, the Python bindings allow one to easily incorporate the methods into computational workflows and postprocess results. Furthermore, this enables integration of SISSO into existing machine-learning frameworks, e.g. scikit-learn (Pedregosa et al., 2011). The code is designed in a modular fashion, which simplifies the process of extending the code for other applications. Finally the project's extensive documentation and tutorials provide a good access point for new users of the method.

Features

The following features are implemented in SISSO++:

- A C++ library for using SISSO to find analytical models for a given problem
- Python bindings to be able to interface with the C++ objects in a Python environment
- Postprocessing tools for visualizing models and analyzing results using Matplotlib (Hunter, 2007)
- Access to solve an *n*-dimensional classification model using a combination of calculating the convex-hull overlap and a linear-support vector machine solver
- The ability to include non-linear paramters within features (e.g. $\exp(\alpha x)$ and $\ln(x + \beta)$) (Purcell et al., 2021)
- Complete API documentation defining all functions of the code
- Tutorials and Quick-Start Guides describing the basic functionality of the code

Code Dependencies

The following libraries are used by SISSO++:

- NLopt (Johnson, n.d.) is used to optimize the non-linear bias and scale paramters within features
- The CLP library from Coin-OR (Forrest et al., n.d.) is used to find the number of points in the convex hull overlap region for classification problems
- LIBSVM (Chang & Lin, 2011) is used to find the linear-SVM model for classification problems
- Scikit-learn (Pedregosa et al., 2011), (Buitinck et al., 2013) is used to update the SVM model for classification problems within Python

- NumPy (Harris et al., 2020) and pandas (team, 2020), (McKinney, 2010) are used to represent the data structures within Python and perform array operations.
- seaborn (Waskom, 2021) and Matplotlib (Hunter, 2007) are used to generate the plots during postprocessing

Acknowledgements

The authors would like to thank Markus Rampp and Meisam Tabriz of the MPCDF for technical support. We would also like to thank Lucas Foppa, Jingkai Quan, Aakash Naik, James Dean, Timur Bazhirov, and Luigi Sbailò for testing and providing valuable feedback. T.P. would like to thank the Alexander von Humboldt Foundation for their support through the Alexander von Humboldt Postdoctoral Fellowship Program. This project was supported by TEC1p (the European Research Council (ERC) Horizon 2020 research and innovation programme, grant agreement No. 740233), BiGmax (the Max Planck Society's Research Network on Big-Data-Driven Materials-Science), and the FAIRmat consortium of the NFDI and FAIR-DI e.V. association.

References

Andersen, M., Levchenko, S. V., Scheffler, M., & Reuter, K. (2019). Beyond Scaling Relations for the Description of Catalytic Materials. *ACS Catal.*, *9*(4), 2752–2759. https://doi.org/10.1021/acscatal.8b04478

Andersen, M., & Reuter, K. (2021). Adsorption Enthalpies for Catalysis Modeling through Machine-Learned Descriptors. *Acc. Chem. Res.*, *54*(12), 2741–2749. https://doi.org/10.1021/acs.accounts.1c00153

Bartel, C. J., Millican, S. L., Deml, A. M., Rumptz, J. R., Tumas, W., Weimer, A. W., Lany, S., Stevanović, V., Musgrave, C. B., & Holder, A. M. (2018). Physical descriptor for the Gibbs energy of inorganic crystalline solids and temperature-dependent materials chemistry. *Nat. Commun.*, *9*(1), 1–10. https://doi.org/10.1038/s41467-018-06682-4

Bartel, C. J., Sutton, C., Goldsmith, B. R., Ouyang, R., Musgrave, C. B., Ghiringhelli, L. M., & Scheffler, M. (2019). New tolerance factor to predict the stability of perovskite oxides and halides. *Sci. Adv.*, *5*(2). https://doi.org/10.1126/sciadv.aav0693

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., & Varoquaux, G. (2013). API design for machine learning software: Experiences from the scikit-learn project. *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 108–122.

Cao, G., Ouyang, R., Ghiringhelli, L. M., Scheffler, M., Liu, H., Carbogno, C., & Zhang, Z. (2020). Artificial intelligence for high-throughput discovery of topological insulators: The example of alloyed tetradymites. *Phys. Rev. Mater.*, *4*(3), 034204. https://doi.org/10.1103/PhysRevMaterials.4.034204

Chang, C.-C., & Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, *2*, 27:1–27:27.

Forrest, J. J., Vigerske, S., Ralphs, T., Hafer, L., Fasano, J. P., Santos, H. G., Saltzman, M., Gassmann, H. I., Kristjansson, B., & King, A. (n.d.). *Coin-or/clp: Version 1.17.6*. http://dx.doi.org/10.5281/zenodo.3748677

Gasper, P. (n.d.). https://github.com/NREL/SISSORegressor_MATLAB

Han, Z. K., Sarker, D., Ouyang, R., Mazheika, A., Gao, Y., & Levchenko, S. V. (2021). Single-atom alloy catalysts designed by first-principles calculations and artificial intelligence. *Nat. Commun.*, *12*(1), 1–9. https://doi.org/10.1038/s41467-021-22048-9

Harris, C. R., Millman, K. J., Walt, S. J. van der, Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Kerkwijk, M. H. van, Brett, M., Haldane, A., Río, J. F. del, Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

Johnson, S. G. (n.d.). *The NLopt nonlinear-optimization package*. http://github.com/stevengj/nlopt

McKinney, Wes. (2010). Data Structures for Statistical Computing in Python. In Stéfan van der Walt & Jarrod Millman (Eds.), *Proceedings of the 9th Python in Science Conference* (pp. 56–61). https://doi.org/10.25080/Majora-92bf1922-00a

Neumann, P., Cao, L., Russo, D., Vassiliadis, V. S., & Lapkin, A. A. (2020). A new formulation for symbolic regression to identify physico-chemical laws from experimental data. *Chem. Eng. J.*, *387*, 123412. https://doi.org/10.1016/j.cej.2019.123412

Ouyang, R. (n.d.). *GitHub - rouyang2017/SISSO: A data-driven method combining symbolic regression and compressed sensing toward accurate & interpretable models*. Retrieved September 2, 2021, from https://github.com/rouyang2017/SISSO

Ouyang, R., Ahmetcik, E., Carbogno, C., Scheffler, M., & Ghiringhelli, L. M. (2019). Simultaneous learning of several materials properties from incomplete databases with multi-task SISSO. *J. Phys. Mater.*, *2*(2), 024002. https://doi.org/10.1088/2515-7639/ab077b

Ouyang, R., Curtarolo, S., Ahmetcik, E., Scheffler, M., & Ghiringhelli, L. M. (2018). SISSO: A compressed-sensing method for identifying the best low-dimensional descriptor in an immensity of offered candidates. *Phys. Rev. Mater.*, *2*(8), 083802. https://doi.org/10.1103/PhysRevMaterials.2.083802

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Pilania, G., Iverson, C. N., Lookman, T., & Marrone, B. L. (2019). Machine-Learning-Based Predictive Modeling of Glass Transition Temperatures: A Case of Polyhydroxyalkanoate Homopolymers and Copolymers. *J. Chem. Inf. Model.*, *59*(12), 5013–5025. https://doi.org/10.1021/acs.jcim.9b00807

Purcell, T. A. R., Scheffler, M., Ghiringhelli, L. M., & Carbogno, C. (2021). Accelerating Material-Space Exploration by Mapping Materials Properties via Artificial Intelligence: The Case of the Lattice Thermal Conductivitys. *In Preparation*.

Schleder, G. R., Acosta, C. M., & Fazzio, A. (2020). Exploring Two-Dimensional Materials Thermodynamic Stability via Machine Learning. *ACS Appl. Mater. Interfaces*, *12*(18), 20149–20157. https://doi.org/10.1021/acsami.9b14530

team, T. pandas development. (2020). *Pandas-dev/pandas: pandas* (latest) [Computer software]. Zenodo. https://doi.org/10.5281/zenodo.3509134

Udrescu, S. M., & Tegmark, M. (2020). AI Feynman: A physics-inspired method for symbolic regression. *Sci. Adv.*, *6*(16). https://doi.org/10.1126/sciadv.aay2631

Wang, Y., Wagner, N., & Rondinelli, J. M. (2019). Symbolic regression in materials science. *MRS Commun.*, *9*(3), 793–805. https://doi.org/10.1557/mrc.2019.85

Waroquiers, D. (n.d.). *Python interface to the SISSO (sure independence screening and sparsifying operator) method.* https://github.com/Matgenix/pysisso

Waskom, M. L. (2021). Seaborn: Statistical data visualization. *Journal of Open Source Software*, *6*(60), 3021. https://doi.org/10.21105/joss.03021

Xu, C. (n.d.). *SISSOkit*. https://github.com/chuanqixu/SISSOkit

Xu, W., Andersen, M., & Reuter, K. (2021). Data-Driven Descriptor Engineering and Refined Scaling Relations for Predicting Transition Metal Oxide Reactivity. *ACS Catal.*, *11*(2), 734–742. https://doi.org/10.1021/acscatal.0c04170