Thesis Project description

Thesis topic

Efficient molecular machine learning with the Laplacian kernel

Objective of the thesis

Virtual design of new materials and drugs is one of the currently heavily investigated fields in which the use of computational methods promises to strongly support science discovery. One potential part of this design process is the prediction of quantum chemical properties (e.g. ground or excited state energies) of molecules that are part of the new materials or drugs. Often, we are interested to do such a prediction for trajectories of molecules, hence sequences of identical molecules whose atoms are however slightly shifted due to some external process. Given a subset of a trajectory of molecules and calculated properties for each molecule in this subset trajectory, the objective is then to find a machine learning model that can predict these properties for any remaining molecule on that trajectory or other similar trajectories.

An important class of machine learning models for this application are kernel-based models. These include at least Kernel Ridge Regression [1] (KRR) and Gaussian Process Regression [2] (GPR). These models often have low prediction error even for moderately small training sets, hence have a good prediction error. In particular, it is observed that the so-called Laplacian kernel has clearly better prediction error than the often used Gaussian/RBF kernel. At the same time, benchmarks with scikit-learn suggest a significantly increased computation time for the Laplacian kernel in contrast to the Gaussian kernel.

The objective of this project is to investigate this phenomenon and to find a proposal how to overcome the challenge of the higher runtime. In particular, this will require an in-depth analysis of the differences in the implementation in scikit-learn, a benchmark of various other libraries for KRR or GPR, among others QML [3], GPy [4], Gpflow [5], pyro [6], pymc [7], etc., and maybe a better re-implementation of kernel-based machine learning using the Laplacian kernel.

Sources

[1] Rupp, Matthias. (2015). Machine learning for quantum mechanics in a nutshell. International Journal of Quantum Chemistry.

[2] Carl Edward Rasmussen and Christopher K. I. Williams. 2005. Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning). The MIT Press.

[3] Web page of "QML: A Python Toolkit for Quantum Machine Learning" https://github.com/qmlcode/qml, last access 2022/12/07.

[4] Web page of “GPy - A Gaussian Process (GP) framework in Python”, https://gpy.readthedocs.io/en/deploy/, last access: 2022/12/07

[5] Alexander G. de G. Matthews et al., GPflow: A Gaussian process library using TensorFlow, Journal of Machine Learning Research, 18(40), 1-6, 2017.

[6] Eli Bingham, Jonathan P. Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletsos, Rohit Singh, Paul Szerlip, Paul Horsfall, and Noah D. Goodman. 2019. Pyro: deep universal probabilistic programming. J. Mach. Learn. Res. 20, 1 (January 2019), 973–978.

[7] Salvatier J., Wiecki T.V., Fonnesbeck C. (2016) Probabilistic programming in Python using PyMC3. PeerJ Computer Science 2:e55

Deliverables

D1. Implementation of benchmarking codes for scikit-learn and several other KRR / GPR libraries

D2. Application of benchmarks to provided data sets.

D3. Maybe: Re-Implementation of kernel-based ML for the Laplacian kernel

D4. All source codes for D1-D3 are provided including a reasonable amount of comments in the source code.

D5. Thesis based on the BSc/MSc thesis template.