First status update report
Theoretical Foundations of Data Science

2021-05-11

1 Problem statement

Recurrent neural networks are widely used on time series data, yet such models often ignore the underlying physical structures in such sequences. A new class of physics-based methods related to Koopman theory has been introduced, offering an alternative for processing non-linear dynamical systems.

We focus on dynamical systems that can be described by a time-invariant model

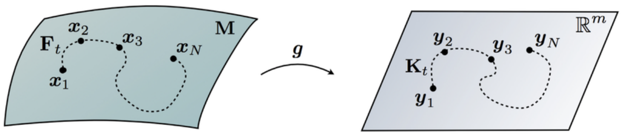
$$x_{t+1} = F_t(x_t), \ \ x \in M \subset \mathbb{R}^m, \tag{1}$$

where x_t denotes the state of the system at time $t \in \mathbb{N}$. The map $F_t : M \to M$ is a (potentially non-linear) update rule on a finite-dimensional manifold M, pushing states from time t to time $t+1$. The above model assumes that future states depend only on the current state x_t and not on information from a sequence of previous states.

Koopman's theory postulates that any non-linear dynamical system can be lifted into the space of observable functions g, in which the system evolves linearly. Specifically, it can be shown that there always exists a K such that:

$$g(x_t) = y_t = Ky_{t-1} = Kg(x_{t-1}) \tag{2}$$

with K commonly referred to as the Koopman operator. Note that the Koopman operator and, therefore y_t might be infinite-dimensional and that g and K are usually unknown.



Even if K and g obtained known, forecasting x_t would still not be trivial because even though one could evolve y_t into the future, transforming knowledge of future values of y_t into the knowledge of x_k is complicated. This is why in the past, g was often assumed to be invertible. Additionally, it was assumed that g could be approximated by a function (as opposed to a functional).

2 Main challenges

The main challenge lies in overcoming the problem that the dimension can be infinite, and it is better to find the operator's eigenfunctions. Many approaches seek to identify eigenfunctions of the Koopman operator directly, satisfying:

$$g(x_{t+1}) = Kg(x_t) = \lambda g(x_t) \tag{3}$$

Eigenfunctions are guaranteed to span an invariant sub-space, and the Koopman operator will yield a matrix when restricted to this subspace. In practice, Koopman eigenfunctions may be more difficult to obtain than the solution of (1); however, this is a one-time up-front cost that yields a compact linear description. The challenge of identifying and representing Koopman eigenfunctions provides strong motivation for the use of powerful emerging deep learning methods.

3 Baseline solution

For short-term forecasting, the following algorithm [**1**] demonstrated the state-of-the-art. The work requires the consistency of the system, that is, the ability to make predictions in both directions. This minimizes the following loss

$$\varepsilon = \lambda_{id}\varepsilon_{id} + \lambda_{fwd}\varepsilon_{fwd} + \lambda_{bwd}\varepsilon_{bwd} + \lambda_{con}\varepsilon_{con},$$

where ε_{id} is a reconstruction loss. $\varepsilon_{fwd}, \varepsilon_{bwd}$ – k steps forward (backward) prediction error. ε_{con} – consistency loss.
For long-term forecasting, there was proposed Spectral Methods usage with Koopman theory [**2**], and a comparison with Fourier transform is made.

4 Roles for the participants (preliminary)

Nikita Balabin (50%) – Refactoring the two methods with PyTorch Lightning as the main framework for the library. Introduction of Ray to optimize all hyperparameters. + Main implementations.

Oleg Maslov (50%) – Introduction the unit testing and provide a test coverage of 70% of the codebase. Creation of the necessary documentation for the API using readthedocs. Providing notebooks with examples on how to run each method. + Main implementations.

5 Repository structure

```
.
├── from_fourier_to_koopman
      ├── examples.py
      ├── fourier_koopman
            ├── fourier.py
            ├── __init__.py
            └── koopman.py
      ├── imgs
            ├── fourier_koopman_objectives.png
            └── youtube_thumb.png
      ├── LICENSE
      ├── README.rst
      └── unknown_phase_problem.ipynb
├── koopmanAE
      ├── driver.py
      ├── model.py
      ├── plot
            └── pred_pendulum.png
      ├── plot_pred_error.py
      ├── read_dataset.py
      ├── README.md
      ├── tools.py
      ├── training_parms.txt
      └── train.py
├── peer-reviews
      ├── first_peer_review_Prophet.pdf
      ├── Firtst_peer_review_DVT.pdf
      ├── Firtst_peer_review_Feature_selection.pdf
      └── README.md
├── README.md
└── reports
    └── first_report.pdf
```

6 Link to the GitHub repository

https://github.com/adasegroup/koopman_forecasting

References

[1] Azencot O, Erichson NB, Lin V, Mahoney M. Forecasting sequential data using consistent Koopman autoencoders. InInternational Conference on Machine Learning 2020 Nov 21 (pp. 475-485). PMLR.

[2] Lange H, Brunton SL, Kutz JN. From fourier to koopman: Spectral methods for long-term time series prediction. Journal of Machine Learning Research. 2021;22(41):1-38.