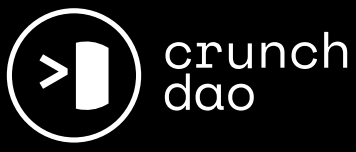


Machine Learning Methods for Nonlinear Reduced-order Modeling of the Thermospheric Density Field

Vahid Nateghi, Matteo Manzi

2022/09/15

Machine Learning Methods for Nonlinear Reduced-order Modeling of the Thermospheric Density Field

abstract

Accurate prediction of the thermospheric density field has recently been gaining a lot of attention, due to an outstanding increase in space operations in Low-Earth Orbit, in the context of the NewSpace. In order to model such high-dimensional systems, Reduced-Order Models (ROMs) have been developed against existing physics-based density models. In general, the data-driven reconstruction of dynamical systems usually consists of two steps: compression and prediction. In this paper, we focus on the compression step and assess state-of-the-art order reduction methodologies such as autoencoders, linear, and nonlinear Principal Component Analysis (PCA). Kernel-based PCA, which is a nonlinear variation of PCA, outperforms Neural Networks in representing the model in low-dimensional space in terms of accuracy, computational time, and energy consumption for both the Lorenz system, a chaotic dynamical system developed in the context of atmospheric modeling, and the thermospheric density.

Plain Language Summary

In the context of the commercial activities performed in Low-Earth Orbit, a region of space of a few hundred kilometers of altitude, because the space traffic is increasing, it is important for us to obtain a model of the density field of the atmosphere, as the motion of the satellites in this orbital regime is strongly influenced by atmospheric drag, which is a function of the atmospheric density. While such models, based on first principles, already exist, they are complex and require a lot of computations; at the same time, more empirical models are less accurate. The trade-off proposed in this work, called reduced-order modeling, enables us to obtain a compressed representation of the density field, which can be used to construct predictive models, perform uncertainty quantification, and estimate the position of spacecraft in the future taking into account our knowledge of the environment and our availability of observation data. We here focus on non-linear methods, to perform the compression, using Machine Learning Methods. In particular, the use of Neural Networks is compared with the use of Support Vector Machine Methods. Interestingly, for the datasets investigated, the latter technique is not only much more efficient but also more accurate.

1. Introduction

As a result of new advancements in satellite design and low-cost launches, we are witnessing an outstanding increase in the number of satellites in Low Earth Orbit (LEO). The dynamics of satellites in such orbital regimes are influenced by certain sources of perturbations among which thermospheric drag has the largest impact on orbital motion uncertainty. To determine the drag force, accurate knowledge of the density field is required; however, because of the complex physical interactions of space weather on the density field in extreme events (Berger et al. 2020), it is needed to perform uncertainty quantification and real-time calibration. Previous works highlighted for instance the importance of nonlinearity in the field as a response to solar storms (Licata et al. 2021).

Due to the lack of accuracy in empirical density models and computational inefficiency of physics-based models of density (Gondelach and Linares 2020), recent works have proposed Reduced-Order Modeling (ROM) techniques as a trade-off between the existing models. The ROMs are constructed usually by a compression step in which the system is represented by a lower number of states, compared to the observed one, and a prediction step in which the model describing the time evolution of the reduced-order states is reconstructed.

Previous works attempted at compressing the thermospheric density field using Linear Methods (Licata et al. 2022) and Neural Networks (Manzi and Vasile 2021; Nateghi, Manzi, and Vasile 2021), which have been successfully used to compress and predict dynamical systems in general (Champion et al. 2019, Luchtenburg_2021). Depending on the dataset, autoencoders have been shown to perform better than linear methods. However, for the compression of the thermospheric density field, Principal Component Analysis (PCA) performs better than autoencoders. In (Licata and Metha 2020), the importance of hyper-optimization is highlighted; in (Turner et al. 2020) different datasets lead to different comparisons between Neural Networks and Linear Methods. Machine Learning can be powerful in dealing with dynamical systems (Karniadakis et al. 2021), particularly because of the functional analysis framework in which it is embedded: however, while autoencoders “might seem preferable to a detailed nonlinear analysis, the resulting neural network models require extensive tuning, lack physical interpretability, generally perform poorly outside their training range and tend to be unnecessarily complex" (Cenedese et al. 2022).

Inspired by the successful use of Kernel Methods in the context of Koopman Operator Theory (Williams, Rowley, and Kevrekidis 2014), (Klus, Nüske, and Hamzi 2020) and finance (LeFloch et al. 2021), this work aims at using alternative generalizations of Principal Component Analysis: Kernel PCA (Schölkopf, Smola, and Müller 1997). KPCA is a nonlinear version of PCA that is developed for compression and denoising applications (Bakır, Weston, and Schölkopf 2004). KPCA has been proven as a powerful method for data reconstruction in general cases (Mika et al. 1998) and particularly Earth observation data (Bueso, Piles, and Camps-Valls 2020). In this work, we aim at analysing KPCA along with the other nonlinear dimensionality reduction methods.

This article is organized as follows. In Section 2, we elaborate on the different methodologies that are followed by Section 3, containing the numerical results for a toy case as well as the thermospheric density model by making a comparison on the accuracy, computation time and energy consumption among the methodologies. Conclusions and the ideas for the future work are reported in Section 4.

2. Methodology - Dimensionality Reduction

The first step before applying any data-driven technique to derive any dynamical model for a high-dimensional system is to reduce the order of the problem with the least loss of information to improve the computation time and performance of the regression, generally speaking. In this section, the state-of-the-art linear and nonlinear algorithms for dimensionality reduction are discussed.

Linear Reduced-Order Modeling: PCA

A commonly used and straightforward linear method for order reduction is PCA which aims at representing variable $x \in \mathbb{R}^d$ by a set of linear basis functions of dimension k, being $k \ll d$. In this regard, the zero-mean variable x can be approximated by the first r dominant orthogonal spatial modes:

$$x(s,t) \approx \sum_{i=1}^{r} c_i(t)\Phi_i(s) \tag{1}$$

in which s is the spatial grid, Φ_i are the spatial modes, and c_i are the corresponding time variant coefficients. The spatial modes Φ_i can be computed by diagonalizing the centered covariance matrix of a data, defined $C = XX^T$ or alternatively by using Singular Value Decomposition (SVD) of the matrix X of size $d \times n_s$ containing n_s samples of x. By performing SVD, the matrix X can be factorized by two unit matrices $U \in \mathbb{R}^{d\times d}$ and $V \in \mathbb{R}^{n_s\times n_s}$ and a diagonal matrix $\Sigma \in \mathbb{R}^{d\times n_s}$ in the form:

$$X = U\Sigma V^T \tag{2}$$

The reduced-order states as a result are computed:

$$z = U_r^T x \tag{3}$$

where U_r is defined as the first r columns of U matrix. The output of this analysis is the n orthogonal directions with biggest variance containing as much information as possible.

Nonlinear Reduced-Order Modeling with Kernel PCA

Although PCA is a strong method for dimensionality reduction, in a large number of cases, a d-dimensional variable x, can not be expressed in a linear manifold without significant loss of information. Among the various proposed nonlinear dimensionality reduction algorithms, Kernel-based PCA is widely used for its simplicity and ease of implementation.

In case of having no lower-dimensional linear subspace on which x can lie, KPCA proposes to bring the system to an even higher dimension D through an arbitrary transformation $\Phi : \mathbb{R}^d \rightarrow \mathbb{R}^D$. Although it may sound counter-intuitive in a dimensionality reduction problem, the transformation is designed such that the modified variable $\tilde{x}$ can be described by linear low-dimensional manifold by performing the PCA.

As one guesses, producing such a very large-dimensional space and performing any transformation in this space is computationally costly. The *kernel trick* is a solution for this problem. The essence of this trick is that one does not need to explicitly apply the transformation Φ and represent the data with generating the higher-dimensional space with the new transformed coordinates, and instead, it is sufficient to perform a pairwise similarity comparisons between the original data and compute $\kappa(.,.)$ such that:

$$\kappa(x^i, x^j) = \langle \Phi(x^i), \Phi(x^j) \rangle \tag{4}$$

It should be noted that there is a difficulty in this method which is the selection of the kernel function or Φ transformation. This transformation can differ by case depending on how the data shape. There are a number of typical kernels such as Gaussian, radial basis functions (RBF) and polynomials. According to the data of interest in this work, RBF, a generalized case of Gaussian, has been selected:

$$\kappa(x^i, x^j) = exp(-\beta ||x^i - x^j||^2) \tag{5}$$

Once the kernel is defined, the Gram matrix $K \in \mathbb{R}^{n_s \times n_s}$ of the kernel space is formed:

$$K_{i,j} = \kappa(x^i, x^j) \;\; for \;\; i, j = 1, 2, ..., n_s. \tag{6}$$

Note that kernel must be selected such that K matrix is positive definite. Now that K is known, its SVD can be computed:

$$K = V \Sigma V^T \tag{7}$$

The reduced-order states can be computed the same as introduced in PCA:

$$z = V_r^T k(x) \tag{8}$$

where V_r is obtained by the first r columns of V and $k(x)$ is the i-th column of K matrix corresponding

to the i-th sample reduced state.

Note that for improving the efficiency of PCA, the Gram matrix K needs to be centered:

$$K_{centered} = (I - 1_{n_s})K(I - 1_{n_s}) \tag{9}$$

where I is the identity matrix and 1_{n_s} is a $n_s \times n_s$ matrix whose elements are all equal to $1/n_s$. More details on centering the kernel and the proof can be found in (García-González et al. 2020).

KPCA is a general case of PCA, that if we introduce a linear kernel by which $\Phi(x) = x$, KPCA corresponds to standard PCA in a higher dimension space. Thus, the only added computational complexity to the problem is just the calculation of inner products that is hardly changed if the selected kernel is easy to compute(Schölkopf, Smola, and Müller 1998).

As a result of KPCA being a generalization of PCA, according to what introduced and proved in (Schölkopf and Smola 2001), the first r principal components computed via KPCA are still uncorrelated and carry the highest variance among any set of orthogonal directions.

KPCA backward mapping problem

In the standard PCA, the k-dimensional linear manifold spanning by the first r modes is defined in the original space $\mathbb{R}^d$ and therefore, the backward mapping from the reduced-order states into the original one is simply $\tilde{x} = U_r z$. This is not the case for KPCA and it is well-known as *pre-image* problem which tries to find a corresponding $x \in \mathbb{R}^d$, given a point Ψ in the kernel space with kernel function Φ such that $\Psi = \Phi(x)$.

The first attempt is to find an exact pre-image. If an exact pre-image exists and if the chosen kernel is invertible, one can compute the pre-image analytically. However, this is not practical as it does not usually exist an input data in the original space corresponding to the any point in the kernel space.

In case the exact pre-image does not exist, the pre-image $\tilde{x}$ is *approximated*:

$$\tilde{x} = \arg\min_x ||\Psi - \Phi(x)|| \tag{10}$$

To solve the minimization problem, some proposes iterative schemes to find the pre-images (Kwok and Tsang 2004). That can be an option but the scheme highly depends on the particular type of kernels and does not necessarily provide the global minimum.

To find the optimal solution, kernelized ridge regression is proposed (Bakır, Weston, and Schölkopf 2004). For each $\boldsymbol{x} \in \mathbb{R}^d$ and $\boldsymbol{X} \in \mathbb{R}^{n_s \times d}$, there exist vector $\boldsymbol{\omega} = [\omega_1, \omega_2, ..., \omega_{n_s}] \in \mathbb{R}^{n_s}$ such that:

$$\boldsymbol{x} = \boldsymbol{\omega} \boldsymbol{X} \tag{11}$$

To find the vector $\boldsymbol{\omega}$, the kernel ridge regression can be written as:

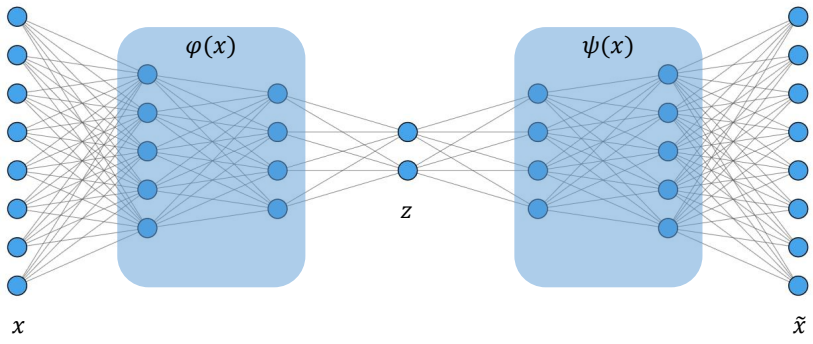
$$J_{\omega} = \arg\min_{\omega} ||\Psi - \Phi(\boldsymbol{\omega}\boldsymbol{X})||^2 + \lambda||\boldsymbol{\omega}||^2 \tag{12}$$

where the second term is a L^2 norm regularizer. The optimal solution of the dual problem of such minimization problem is given by (Murphy 2012):

$$\boldsymbol{\omega} = X^T(XX^T + \lambda I_{n_s})^{-1}\Psi \tag{13}$$

Nonlinear Reduced-Order Modeling with Autoencoders

New advancements in the area of machine learning made them appealing to various applications, namely dimensionality reduction. Autoencoders are Neural Networks (NN) consisted of an encoder that takes an input vector $x^{N\times 1}$ and outputs a vector $z^{r\times 1}$ and a decoder that takes the encoder's output and provides a vector $\tilde{x}^{N\times 1}$. These NNs are trained such that their input is copied into their output. In the application of dimensionality reduction, according to the Figure, if $r < N$, the inner layer has less neurons compared to the input and output and makes the network summarizes the information in a lower dimension vector while keeping the reconstruction error as low as possible.



To perform the training of the autoencoders throughout the work, the mean squared error is used as a loss function according to the input vector x and $\tilde{x}$ and functions $\Phi(x)$ and $\Psi(z)$ for the encoder and decoder, respectively:

$$L^2(x, \Psi(\Phi(x))) = ||x - \tilde{x}||^2 \tag{14}$$

The activation functions in such network can be nonlinear that makes it a powerful tool for the learning and reconstruction of a nonlinear manifold. Assuming only one layer for the encoder and decoder with linear activation functions, the network resembles a standard PCA as shown in the figure.

3. Results

To have a comparison on the performance of the algorithms for dimensionality reduction, they are used for the reduced-order modeling of the Lorenz system as a toy example of chaotic dynamical system and the thermospheric density.

Lorenz System

We start assessing the described methodologies to a dynamical system whose differential equations are explicitly known. We do so in order to have an accurate measure on the reconstruction error, as the dynamics is known and can be compared with the result of the compression associated to each algorithm.

In this regard, we follow (Champion et al. 2019) in building an high dimensional system upon the chaotic Lorenz system. The nonlinear dynamics of the Lorenz system is described by the following set of equations:

$$\begin{aligned} \dot{z}_1 &= \sigma(z_2 - z_1) \\ \dot{z}_2 &= z_1(\rho - z_3) - z_2 \\ \dot{z}_3 &= z_1 z_2 - \beta z_3 \end{aligned} \tag{15}$$

Based on this system, the high-dimensional system can be constructed by a mapping from $\mathbb{R}^3$ to $\mathbb{R}^{128}$, using six fixed spatial modes associated to Legendre polynomials.

The relatively high-dimensional system now can be used as the input for the order-reduction algorithms whose performance are accurately assessed given the true three-dimensional Lorenz system.

To fully show the potential of the algorithms, we consider nonlinear mapping, using cubic terms:

$$x(t) = \boldsymbol{u}_1 z_1(t) + \boldsymbol{u}_2 z_2(t) + \boldsymbol{u}_3 z_3(t) + \boldsymbol{u}_4 z_1(t)^3 + \boldsymbol{u}_5 z_2(t)^3 + \boldsymbol{u}_6 z_3(t)^3 \tag{16}$$

where the components of $\boldsymbol{u}_n$ are the 128 evaluations of the n^{th} Legendre polynomial. The dataset we are using for Lorenz system in this work is made of 256000 snapshots as training set and 2500 snapshots as validation set which is used in PCA and KPCA.

Considering the input/output size of 128 and the latent space dimension of 3, we make use of fully connected layers to build the network. In order to deal with the nonlinear structure of the system, we make use of five layers with hyperbolic tangent activation function. The network is summarized in the Table below.

Input
Encoder
Dense layer, size 128, hyperbolic tangent activation function
Dense layer, size 64, hyperbolic tangent activation function
Dense layer, size 3, hyperbolic tangent activation function
Decoder
Dense layer, size 64, hyperbolic tangent activation function
Dense layer, size 128, hyperbolic tangent activation function

The next table shows the summary of the reconstruction error for the aforementioned dimensionality reduction algorithms for Lorenz system.

Method	Error
PCA	3.4715
KPCA	0.1556
Autoencoder	2.2779

As one expects, due to nonlinear intrinsic of the system, nonlinear methods should perform better than the standard PCA. The results, moreover, are showing that when dealing with such high dimensional system, KPCA outperforms autoencoder. It should be noted that the parameter γ in the definition of RBF kernel in Equation 5 is optimized using *Nelder-Mead* method for an ensemble of equally spaced set of initial guesses.

Thermospheric Density

Now that the applicability of KPCA is demonstrated for a toy case, we can apply the proposed methodologies for thermospheric density field. To initiate the algorithms for thermospheric density, we took the density data from the database of High Accuracy Satellite Drag Model (Tobiska et al. 2021), because of its decent temporal and spatial resolution. The Table below, shows the characteristics of the data. By preprocessing the data, we end up having a dataset consist of 2920 snapshots in a space with 1920

features. Note that 2000 snapshots are used as training set and the rest is assumed for the validation set which is also used in PCA and KPCA algorithms.

	Domain	Resolution
Local solar time [hr]	[0, 24[	3
Latitude [deg]	[-90, 90]	10
Longitude [deg]	[0, 360[	15
Altitude [km]	[500, 600[	25

The tuning of autoencoder here has been optimized for this specific dataset. We made use of KerasTuner (O'Malley et al. 2019), a Python package for hyperparameter optimization. We fixed the activation function and optimized the architecture of the network according to the following parameters:

	Domain
Number of layers	$1, 2, 3$
Number of neurons per layer	$2^4, 2^5, ..., 2^{10}$
Network bottleneck size	$5, 6, ..., 10$

It is worth noting that the number of neurons in layers are selected such that the structure of the network is preserved according to the Figure.

After tuning the autoencoder, the network is now made of 3 fully connected layers for the encoder and decoder and latent space size of 10 shown in 1.

Table 1: Structure of the fully connected autoencoder making use of HASDM data for the modeling of thermospheric density field.

Input
Encoder
Dense layer, size 1920, ReLu activation function
Dense layer, size 1024, ReLu activation function
Dense layer, size 512, ReLu activation function
Dense layer, size 256, ReLu activation function
Dense layer, size 10, ReLu activation function
Decoder
Dense layer, size 256, ReLu activation function
Dense layer, size 512, ReLu activation function
Dense layer, size 1024, ReLu activation function
Dense layer, size 1920, ReLu activation function

In order to have a more accurate reconstruction in both linear and nonlinear ROM methods, the data have been normalized and mean-subtracted, making use of the maximum density value in the space and time of interest.

Table 2 shows the relative error in reconstruction of the thermospheric density field.

Table 2: Reconstruction error for thermospheric density field

Method	L^2 norm of the error	L^∞ norm of the error
PCA	1.2293	0.0074
KPCA	0.2862	0.0080
Autoencoder	21.2226	0.1915

The results show that KPCA could provide one with a set of reduced-order states with the highest accuracy in reconstruction of density field compared to the other methods. To signify the superiority of the method, the L^∞ of the error is also calculated to be sure that the error is bounded throughout the entire validation dataset. It is shown that PCA and KPCA are two orders of magnitude more accurate than Autoencoder, though the L^∞ norm of the error of PCA is comparable to KPCA for this specific dataset.

Here again, the parameter γ in the definition of kernel is optimized. To give an impression on the behaviour of the system with respect to any variation in parameter γ, the reconstruction error is calculated for a wide range of γ and the result is shown in Figure 1.

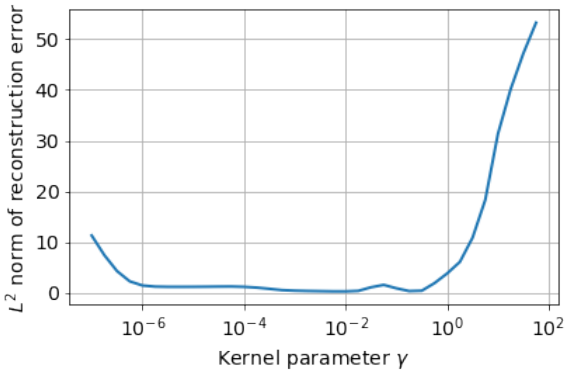


Figure 1: L^2 norm reconstruction error for different kernel parameter γ

The Figure suggests that by decreasing (increasing) the kernel parameter, we are widening (narrowing) the kernel distribution function which results in underfitting (overfitting) the data.

To investigate the sensitivity of the algorithm to the size of the latent space, we apply the three algorithms for different size of reduced order states (The network in the case of Autoencoder has been trained for different size of latent space accordingly). Figure 2 shows the trend of reconstruction error for different order reduction methodologies with respect to the number of states in the latent space.

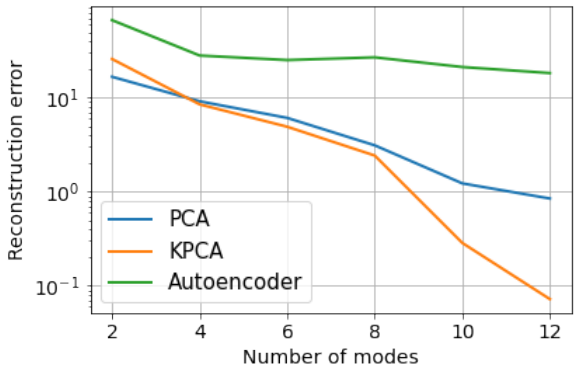


Figure 2: L^2 norm reconstruction error for different latent space size

To have a better understanding on the result of the reconstruction, the input density field and the corresponding reconstruction errors for each method are depicted in Figure 3 on the 1^{th} of January 2013 at midnight for the different algorithms.

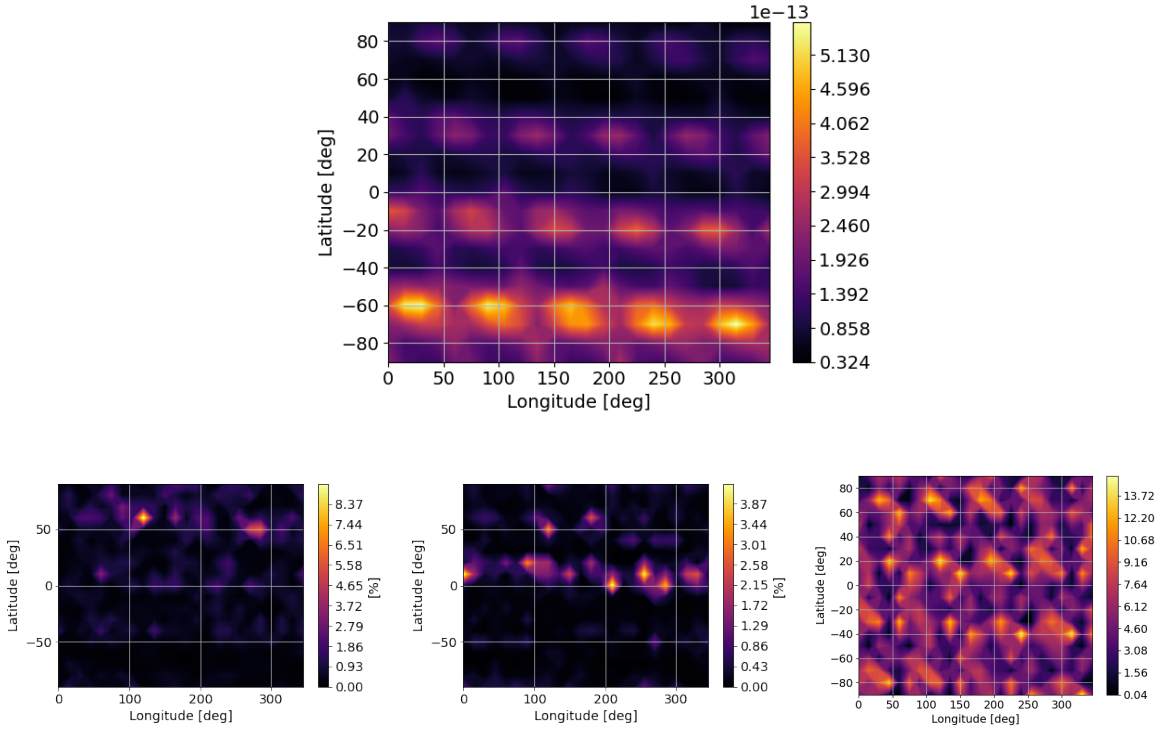


Figure 3: (a) Reference density field, (b) PCA reconstruction error, (c) KPCA reconstruction field, (d) Autoencoder reconstruction field

The result for the reconstruction of density has been shown for a specific year of 2013. To show the robustness of the method and support the drawn result, the three methods have been tested for the density data for the altitude between $500 - 600Km$ for the years between $2011 - 2019$. Figure 4 shows that regardless of datasets, KPCA outperforms the other methods consistently.

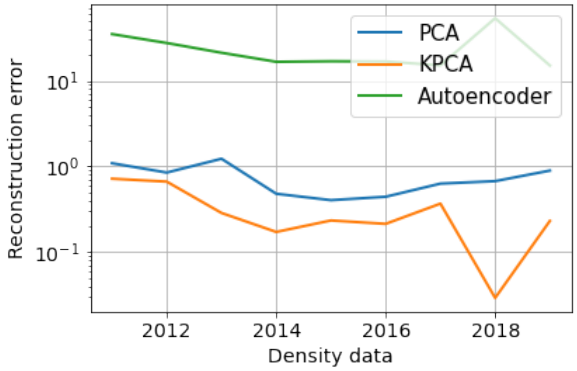


Figure 4: L^2 norm reconstruction error for different datasets

Computation Time and Energy Consumption

Beside benchmarking all the proposed approached for reduced order modeling problems with respect to computational time, inspired by the work by PowerAPI, focused on greener computing, we here also discuss their energy consumption. The general goal is address the environmental issues associated also with the growing carbon footprint of neural network training (Patterson et al. 2021).

The computational cost and the energy consumption of the three routines discussed in this work (Table 3) have been computed using the pyJoules library. Note that the amount of time for the optimization of the parameter γ in KPCA is also included in the execution time.

Table 3: Costs associated to different methodologies

Method	Execution duration [s]	Power Consumption [uJ]
PCA	0.3923	$17.9077 \cdot 10^6$
KPCA	24.6358	$1.3325 \cdot 10^9$
Autoencoder	1298.7059	$39.8931 \cdot 10^9$

This table, combined with the results given in the previous section, show that, at least for the two problems considered in this work, the choice of the model is not a consequence of a trade-off between

accuracy and computational cost.

4. Conclusions and Recommendations

It was shown that, at least for some datasets, Kernel methods are a competitive choice, compared to Neural Networks, particularly in contexts in which efficiency is crucial. Moreover, the modes associated to KPCA have a hierarchy: the accuracy of the compression is proportional to the number of modes used. The same is not guaranteed for autoencoders.

In the context of previous works on atmospheric density estimation in particular, the efficiency of kernel methods is also an important feature, as the propagation requires the decoding of the reduced order state.

This work can still be improved: here, the kernel function has been selected via trial and error and the literature, not from a rigorous optimization; the same can be said about the choice of autoencoders over recent variants of deep and convolutional autoencoders.

Data Availability Statement

The HASDM density database now resides in a SQL database with open community access for scientific studies and can be accessed at this link: https://spacewx.com/hasdm/ and at the Zenodo link https://zenodo.org/record/7046622.

Acknowledgments

The authors would like to thank Stefan Klus for introducing them to Kernel Methods in the context of data-driven dynamical systems and Feliks Nüske for valuable comments that improved the manuscript.

The SET HASDM density data are provided for scientific use courtesy of Space Environment Technologies.

References

Bakır, Gökhan H, Jason Weston, and Bernhard Schölkopf. 2004. “Learning to Find Pre-Images.” *Advances in Neural Information Processing Systems* 16: 449–56.

Berger, Thomas E, M. J. Holzinger, Marcus J. Holzinger, Eric K. Sutton, and Jeffrey P. Thayer. 2020. "Flying Through Uncertainty." *Space Weather-the International Journal of Research and Applications*. https://doi.org/10.1029/2019sw002373.

Bueso, Diego, Maria Piles, and Gustau Camps-Valls. 2020. "Nonlinear Pca for Spatio-Temporal Analysis of Earth Observation Data." *IEEE Transactions on Geoscience and Remote Sensing* 58 (8): 5752–63.

Cenedese, Mattia, Joar Axås, Bastian Bäuerlein, Kerstin Avila, and George Haller. 2022. "Data-Driven Modeling and Prediction of Non-Linearizable Dynamics via Spectral Submanifolds." *Nature Communications* 13 (1). https://doi.org/10.1038/s41467-022-28518-y.

Champion, Kathleen P., Bethany Lusch, J. Nathan Kutz, and Steven L. Brunton. 2019. "Data-Driven Discovery of Coordinates and Governing Equations." *arXiv: Other Statistics*. https://doi.org/10.1073/pnas.1906995116.

García-González, Alberto, Antonio Huerta, Sergio Zlotnik, and Pedro Díez. 2020. "A Kernel Principal Component Analysis (kPCA) Digest with a New Backward Mapping (Pre-Image Reconstruction) Strategy." *arXiv: Numerical Analysis*. https://doi.org/10.21203/rs.3.rs-126052/v1.

Gondelach, David J, and Richard Linares. 2020. "Real-Time Thermospheric Density Estimation via Two-Line Element Data Assimilation." *Space Weather* 18 (2): e2019SW002356.

Karniadakis, George, Yannis Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. 2021. "Physics-Informed Machine Learning," May, 1–19. https://doi.org/10.1038/s42254-021-00314-5.

Klus, Stefan, Feliks Nüske, and Boumediene Hamzi. 2020. "Kernel-Based Approximation of the Koopman Generator and Schrödinger Operator." *Entropy* 22 (7): 722. https://doi.org/10.3390/e22070722.

Kwok, JT-Y, and IW-H Tsang. 2004. "The Pre-Image Problem in Kernel Methods." *IEEE Transactions on Neural Networks* 15 (6): 1517–25.

LeFloch, Philippe G., Philippe G. LeFloch, Philippe G. LeFloch, Mercier Jean-Marc, and Shohruh Miryusupov. 2021. "Codpy - Advanced Tutorial." *Null*. https://doi.org/10.2139/ssrn.3769804.

Licata, Richard J., Piyush M. Mehta, W. Kent Tobiska, and S. Huzurbazar. 2021. "Machine-Learned Hasdm Model with Uncertainty Quantification." arXiv. https://doi.org/10.48550/ARXIV.2109.07651.

Licata, Richard, Piyush Mehta, W Kent Tobiska, and S. Huzurbazar. 2022. "Machine-Learned Hasdm

Thermospheric Mass Density Model with Uncertainty Quantification." *Space Weather* 20 (April). https://doi.org/10.1029/2021SW002915.

Licata, Richard, and Pyush Metha. 2020. "Physics-Informed Machine Learning with Autoencoders and Lstm for Probabilistic Space Weather Modeling and Forecasting." *17th Conference on Space Weather - 100th AMS Annual Meeting*.

Manzi, Matteo, and Massimiliano Vasile. 2021. "AUTOENCODER-Based Thermospheric Density Model for Uncertainty Quantification and Real-Time Calibration." *8th European Conference on Space Debris*.

Mika, Sebastian, Bernhard Schölkopf, Alex Smola, Klaus-Robert Müller, Matthias Scholz, and Gunnar Rätsch. 1998. "Kernel Pca and de-Noising in Feature Spaces." *Advances in Neural Information Processing Systems* 11.

Murphy, Kevin P. 2012. *Machine Learning: A Probabilistic Perspective*. MIT press.

Nateghi, Vahid, Matteo Manzi, and Massimiliano Vasile. 2021. "AUTOENCODER-Based Thermospheric Density Estimation Using Gps Tracking Data." *72nd International Astronautical Congress (IAC)*.

O'Malley, Tom, Elie Bursztein, James Long, François Chollet, Haifeng Jin, Luca Invernizzi, and others. 2019. "KerasTuner." https://github.com/keras-team/keras-tuner.

Patterson, David, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. 2021. "Carbon Emissions and Large Neural Network Training." arXiv. https://doi.org/10.48550/ARXIV.2104.10350.

Schölkopf, Bernhard, and Alexander J. Smola. 2001. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press.

Schölkopf, Bernhard, Alexander Smola, and Klaus-Robert Müller. 1997. "Kernel Principal Component Analysis." In *Artificial Neural Networks — Icann'97*, edited by Wulfram Gerstner, Alain Germond, Martin Hasler, and Jean-Daniel Nicoud, 583–88. Berlin, Heidelberg: Springer Berlin Heidelberg.

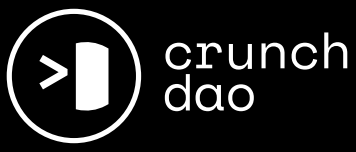
———. 1998. "Nonlinear Component Analysis as a Kernel Eigenvalue Problem." *Neural Computation* 10 (5): 1299–1319.

Tobiska, W. Kent, Bruce R. Bowman, S. David Bouwer, Dave Bouwer, Alfredo Cruz, Kaiya Wahl, Marcin Pilinski, Piyush M. Mehta, and Richard J. Licata. 2021. "The Set Hasdm Density Database." *Social Work*.

https://doi.org/10.1029/2020sw002682.

Turner, Herbert, Maggie Zhang, David Gondelach, and Richard Linares. 2020. "Machine Learning Algorithms for Improved Thermospheric Density Modeling." In *Dynamic Data Driven Applications Systems: Third International Conference, Dddas 2020, Boston, Ma, Usa, October 2-4, 2020, Proceedings*, 143–51. Berlin, Heidelberg: Springer-Verlag. https://doi.org/10.1007/978-3-030-61725-7_18.

Williams, Matthew O., Clarence W. Rowley, and Ioannis G. Kevrekidis. 2014. "A Kernel-Based Approach to Data-Driven Koopman Spectral Analysis." arXiv. https://doi.org/10.48550/ARXIV.1411.2260.


crunch
dao