MACHINE
LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
21 May 2020

REVISED
7 July 2020

ACCEPTED FOR PUBLICATION
28 July 2020

PUBLISHED
22 October 2020

PAPER

Structure-property maps with Kernel principal covariates regression

Benjamin A Helfrecht, Rose K Cersonsky, Guillaume Fraux and Michele Ceriotti

Laboratory of Computational Science and Modeling, IMX, École Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland

E-mail: michele.ceriotti@epfl.ch

Keywords: machine learning, structure-property maps, kernel methods, materials science, physical chemistry

Supplementary material for this article is available online

Abstract

Data analyses based on linear methods constitute the simplest, most robust, and transparent approaches to the automatic processing of large amounts of data for building supervised or unsupervised machine learning models. Principal covariates regression (PCovR) is an underappreciated method that interpolates between principal component analysis and linear regression and can be used conveniently to reveal structure-property relations in terms of simple-to-interpret, low-dimensional maps. Here we provide a pedagogic overview of these data analysis schemes, including the use of the kernel trick to introduce an element of non-linearity while maintaining most of the convenience and the simplicity of linear approaches. We then introduce a kernelized version of PCovR and a sparsified extension, and demonstrate the performance of this approach in revealing and predicting structure-property relations in chemistry and materials science, showing a variety of examples including elemental carbon, porous silicate frameworks, organic molecules, amino acid conformers, and molecular materials.

1. Introduction

Over the past decade, there has been a tremendous increase in the use of data-driven and machine learning (ML) methods in materials science, ranging from the prediction of materials properties [1–4], to the construction of interatomic potentials [5–8] and searches for new candidate materials for a particular application [9–12]. Broadly speaking, these methods can be divided into two categories: those that are focused on predicting the properties of new materials (supervised learning), and those that are focused on finding or recognising patterns, particularly in atomic structures (unsupervised learning). While supervised methods are useful for predicting properties of materials with diverse atomic configurations, they are not as well-suited for classifying structural diversity. Conversely, unsupervised methods are useful for finding structural patterns, but often fail to directly predict materials properties. Moreover, it can be difficult to validate motifs identified by an unsupervised learning algorithm, as the results obtained from the clustering algorithm depend on the choice of the structural representation and can therefore be biased by preconceived expectations on what the most relevant features should be [13].

Methods that combine the predictive power of supervised ML and the pattern recognition capabilities of unsupervised ML stand to be very useful in materials informatics, making it possible to increase data efficiency and more clearly reveal structure-property relations. A number of statistical methods have been developed for augmenting regression models to incorporate information about the structure of the input data, including principal component regression [14], partial least squares regression [15], cluster-wise regression [16], continuum regression [17], and principal covariates regression (PCovR) [18–21]. Among these, PCovR is particularly appealing, because it transparently combines linear regression (LR; a supervised learning method) with principal component analysis (PCA; an unsupervised learning method). The method has found previous applications in climate science [22], macroeconomics [23], social science [20], and bio-informatics [24, 25], but has yet to be widely adopted. A handful of extensions have been developed for PCovR, including a combination with cluster-wise regression [26] and regularised models [22, 24].

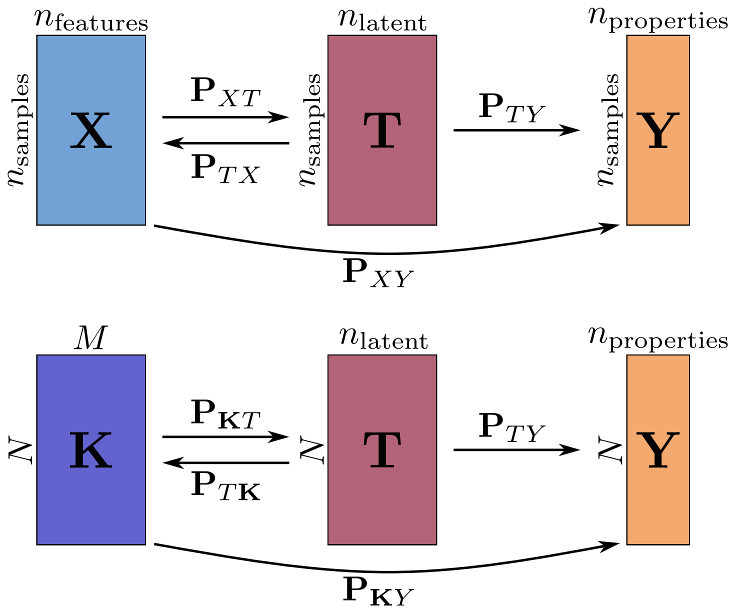

Figure 1. A schematic representation of the different linear operations that can be performed to model structure-property relations in terms of a matrix of features **X** or an associated kernel matrix **K** that represents the input samples.

In this paper, we propose a kernel-based variation on the original PCovR method, which we call *Kernel Principal Covariates Regression* (KPCovR), with the aim of making it even more versatile for statistics and machine learning applications. We begin by summarising the required background concepts and constituent methods used in the construction of linear PCovR in addition to the kernel trick, which can be used to incorporate an element of non-linearity in otherwise linear methods. We then introduce KPCovR, both for full and sparse kernels, and demonstrate their application to several different classes of materials and chemical systems.

2. Background methods

We start by giving a concise but complete overview of established linear methods for dimensionality reduction and regression, as well as their kernelized counterparts. This is done to set a common notation and serve as a pedagogic introduction to the problem, complemented by a set of interactive Jupyter notebooks [27]. Expert readers can skip this section and proceed to section 3, where we introduce kernelized PCovR methods. Throughout this section, we demonstrate the methods on the CSD-1000r dataset [28], which contains the NMR chemical shielding of nuclei in a collection of 1 000 organic crystals comprising C, H, N, and O and their 129 580 atomic environments, of which we use 25 600 in this study. To simplify this into a more illustrative example, we classify and predict simultaneously the chemical shieldings of all nuclei, even though in actual applications one usually would deal separately with each element. As the input features, we use the SOAP power spectrum vectors containing 2 520 features, which discretise a three-body correlation function including information on each atom, its relationships with neighbouring atoms, and the relationships between sets of neighbours [29, 30].

2.1. Notation

In the following, we assume that the input data has been processed in such a way that the nature of each sample (e.g. the composition and structure of a molecule) is encoded as a row of a feature matrix **X**. Each sample is therefore a vector $\mathbf{X}_i$ of length n_{features}, so that **X** has the shape $n_{\text{samples}} \times n_{\text{features}}$. Similarly, the properties associated with each sample are stored in a property matrix **Y**, which has the shape $n_{\text{samples}} \times n_{\text{properties}}$. We denote the data in latent space (i.e. a low-dimensional approximation of **X**) as **T**. We denote each projection matrix from one space to another as $\mathbf{P}_{AB}$, where A is the original space and B is the projected space. As such, the projection matrix from the input space to **T** is $\mathbf{P}_{XT}$, and $\mathbf{P}_{TX}$ is the projection matrix from **T** to the input space. Note that in general projectors $\mathbf{P}_{AB}$ are not assumed to be orthogonal nor full-rank. A graphical summary of the mappings that we consider in this paper is depicted in figure 1.

To simplify notation and to work with unit-less quantities, we assume in our derivations that both **X** and **Y** are centred according to their respective column means and are scaled such that the square of their Frobenius norms are equal to n_{samples} and that the variance of each individual property in **Y** is equal to

$1/n_{\text{properties}}$. A similar centring and scaling procedure is also applied when working with kernels [31]. Centring and scaling is discussed in more detail in appendix A, and demonstrated in the companion Jupyter notebooks [27]. To make notation less cumbersome, variables names are *not* defined uniquely across the entirety of the paper. We re-use variable names for common elements among the different subsections—for example, using $\mathbf{T}$ to represent a low-dimensional latent space in all methods—but the precise definitions of the re-used variables may differ between subsections and should not be confused with one another.

We also use throughout a few additional conventions: (1) we write an approximation or truncation of a given matrix $\mathbf{A}$ as $\hat{\mathbf{A}}$; (2) we use $\tilde{\mathbf{A}}$ to signify an augmented version of $\mathbf{A}$; that is, $\tilde{\mathbf{A}}$ is defined differently from $\mathbf{A}$, but occupies the same conceptual niche (3) we represent the eigendecomposition of a symmetric matrix as $\mathbf{A} = \mathbf{U}_\mathbf{A}\mathbf{\Lambda}_\mathbf{A}\mathbf{U}_\mathbf{A}^T$, where $\mathbf{\Lambda}_\mathbf{A}$ is a diagonal matrix containing the eigenvalues and $\mathbf{U}_\mathbf{A}$ the matrix having the corresponding eigenvectors as columns; (4) we use throughout the Frobenius norm $\|\mathbf{A}\| = \sqrt{\mathrm{Tr}\,(\mathbf{A}^T\mathbf{A})}$; and (5) we define and report the values of the different losses normalised by the number of sample points, but we omit such normalisation in derivations to declutter equations.

2.2. Linear methods

We begin by discussing models of the form:

$$\mathbf{B} = \mathbf{A}\mathbf{P}_{AB} \tag{1}$$

where $\mathbf{B}$ is a target quantity (e.g. a property that one wants to predict or an alternative, lower-dimensional representation of the feature matrix $\mathbf{A}$), and $\mathbf{P}_{AB}$ is a linear projection that maps the features to the target quantity [32, 33].

2.2.1. Principal component analysis

In principal component analysis [34, 35], the aim is to reduce the dimensionality of the feature matrix $\mathbf{X}$ by determining the orthogonal projection $\mathbf{T} = \mathbf{X}\mathbf{P}_{XT}$ which incurs minimal information loss. More formally, we wish to minimise the error ℓ of reconstructing $\mathbf{X}$ from the low-dimensional projection:

$$\ell_{\text{proj}} = \|\mathbf{X} - \mathbf{T}\mathbf{P}_{TX}\|^2/n_{\text{samples}}. \tag{2}$$

The requirement that $\mathbf{P}_{XT}$ is orthonormal implies that $\mathbf{P}_{TX} = \mathbf{P}_{XT}^T$. Using the properties of the Frobenius norm, ℓ can be rewritten as

$$\ell = \mathrm{Tr}\left(\mathbf{X}\left(\mathbf{I} - \mathbf{P}_{XT}\mathbf{P}_{XT}^T\right)\mathbf{X}^T\right) \tag{3}$$

which is minimised when the similarity

$$\rho = \mathrm{Tr}(\mathbf{P}_{XT}^T\mathbf{X}^T\mathbf{X}\mathbf{P}_{XT}) \tag{4}$$

is maximised. Given the orthogonality constraint on $\mathbf{P}_{XT}$, the similarity is maximised when $\mathbf{P}_{XT}$ corresponds to the eigenvectors of the covariance $\mathbf{C} = \mathbf{X}^T\mathbf{X}$ that are associated with the n_{latent} largest eigenvalues. We introduce the eigendecomposition $\mathbf{C} = \mathbf{U}_\mathbf{C}\mathbf{\Lambda}_\mathbf{C}\mathbf{U}_\mathbf{C}^T$, where $\mathbf{U}_\mathbf{C}$ is the matrix of the eigenvectors and $\mathbf{\Lambda}_\mathbf{C}$ the diagonal matrix of the eigenvalues, so that

$$\mathbf{T} = \mathbf{X}\hat{\mathbf{U}}_\mathbf{C}, \tag{5}$$

where we use the notation $\hat{\mathbf{U}}$ to indicate the matrix containing only the top n_{latent} components. The outcomes of a PCA with $n_{\text{latent}} = 2$ of the CSD-1000r dataset are shown in figure 2(a). The atomic environments are split clearly according to the nature of the atom sitting at the centre of the environment, reflecting the prominence of this information in the SOAP features we use.

2.2.2. Multidimensional scaling

A reduction in the dimensionality of the feature space can also be achieved with a different logic that underlies several methods grouped under the label of multidimensional scaling (MDS) [36]. In MDS, the latent space is chosen to preserve the pairwise distances of the original feature space, corresponding to the loss

$$\ell = \frac{1}{n_{\text{samples}}}\sum_{i<j}\left(\left|\mathbf{X}_i - \mathbf{X}_j\right|^2 - \left|\mathbf{t}_i - \mathbf{t}_j\right|^2\right)^2, \tag{6}$$

where $\mathbf{X}_i$ and $\mathbf{t}_i$ refer to the full and projected feature vector of the i^{th} sample. In general, equation (6) can be applied to all sorts of measures of feature dissimilarity, and requires an iterative optimisation. When the

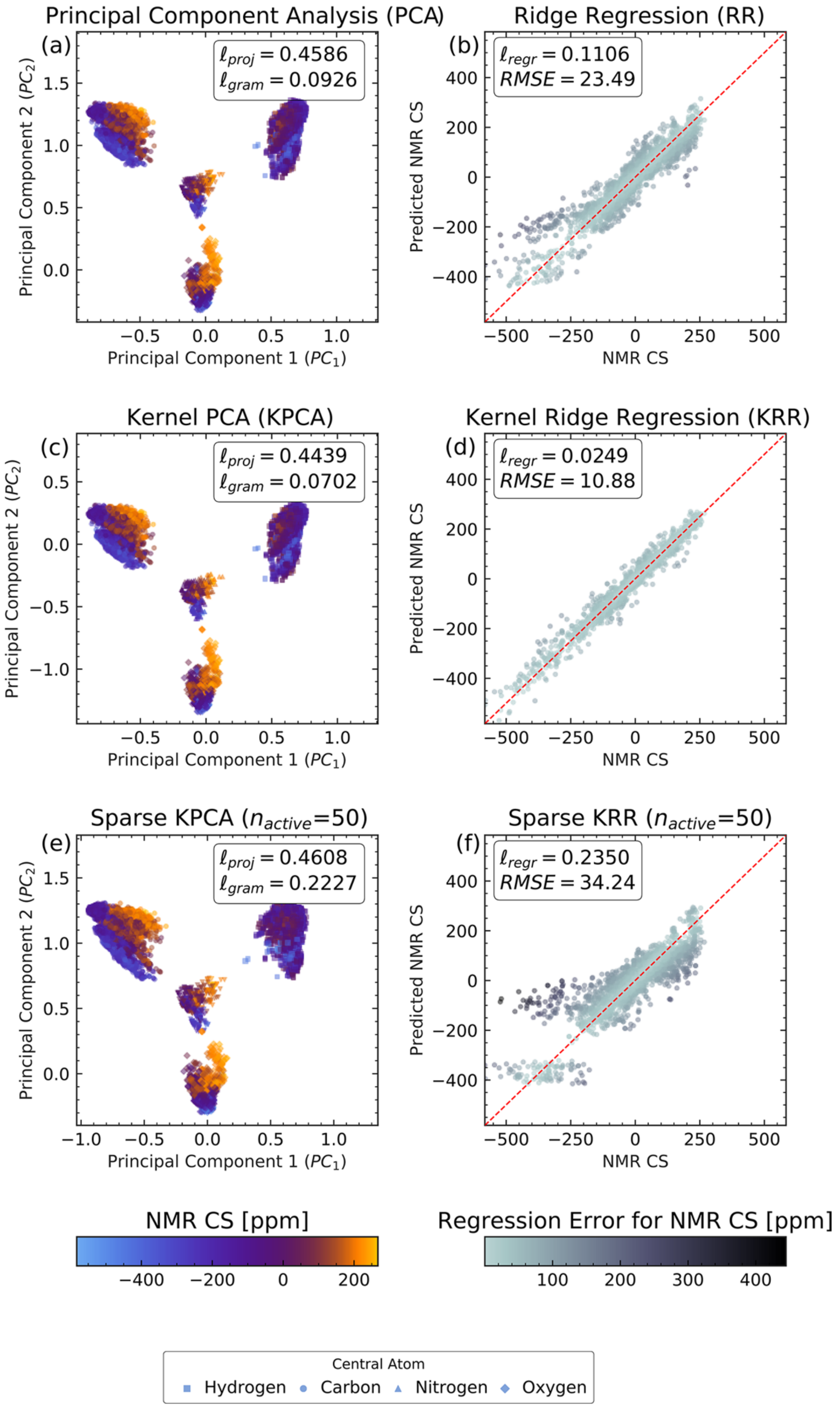

Figure 2. Projection and Regression Models of CSD-1000r. In each projection, the property values are denoted by marker colour and the marker symbol denotes the central atom of each environment, which corresponds to the cluster in the projection. In each regression, the target is denoted by a dotted line. colours denote absolute error of predicted properties, and inset includes the values of $\ell_{\text{proj}} = \|\mathbf{X} - \mathbf{TP}_{TX}\|^2/n_{\text{samples}}$, $\ell_{\text{regr}} = \|\mathbf{Y} - \mathbf{TP}_{TY}\|^2/n_{\text{samples}}$, and root-mean-square error (RMSE), where appropriate. Projections: (a) Principal Components Analysis (PCA) and Multidimensional Scaling (MDS), (c) Kernel PCA, and (e) Sparse Kernel PCA, with $n_{active} = 50$. Regressions: (b) Ridge Regression, (d) Kernel Ridge Regression (KRR), and (f) Sparse KRR, with $n_{active} = 50$. It is important to note that the regressions performed in (b), (d), and (f) are computed on the full input space, and greatly outperform regressions performed on the corresponding latent-space projections in (a), (c), and (e).

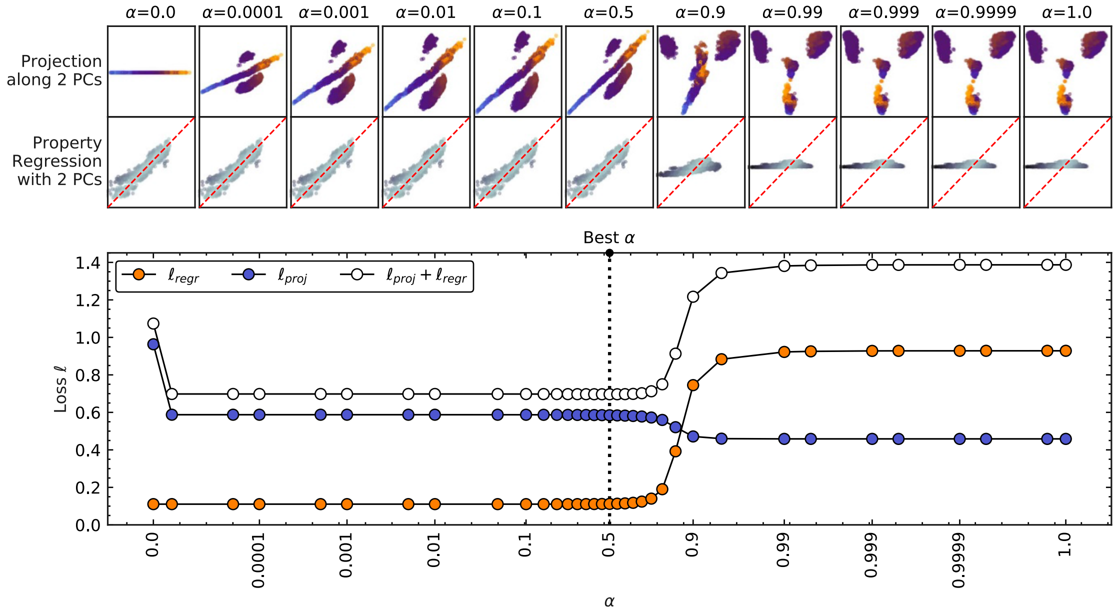

Figure 3. Principal Covariates Regression of CSD-1000r. Combining ridge regression (far left) and PCA (far right) with mixing parameter α, PCovR can minimise the total loss $\ell = \ell_{\text{proj}} + \ell_{\text{regr}} = \|\mathbf{X} - \mathbf{TP}_{TX}\|^2/n_{\text{samples}} + \|\mathbf{Y} - \mathbf{TP}_{TY}\|^2/n_{\text{samples}}$, as denoted in white in the figure. The upper panels show the resulting projections and regressions at the indicated α value, aligned with the horizontal axis in the lower plot. Colour mappings correspond to those in figure 2 and figure 5.

distance between features is the Euclidean distance, the link between the metric and the scalar product suggests using the alternative loss

$$\ell_{\text{gram}} = \|\mathbf{K} - \mathbf{TT}^T\|^2/n_{\text{samples}}. \tag{7}$$

Minimizing this loss (a procedure that is referred to as 'classical MDS') is not equivalent to minimising equation (6), as the two losses concur only if one can find a solution that zeroes ℓ. If the eigenvalue decomposition of the Gram matrix reads $\mathbf{K} = \mathbf{XX}^T = \mathbf{U_K\Lambda_K U_K^{\mathit{T}}}$, ℓ is minimised when $\mathbf{TT}^T$ is given by the singular value decomposition of $\mathbf{K}$, that is by taking

$$\mathbf{T} = \hat{\mathbf{U}}_\mathbf{K}\hat{\mathbf{\Lambda}}_\mathbf{K}^{1/2} \tag{8}$$

restricted to the largest n_{latent} eigenvectors. However, $\mathbf{C}$ and $\mathbf{K}$ have the same (non-zero) eigenvalues, and the (normalised) eigenvectors are linked by $\mathbf{U_K} = \mathbf{XU_C\Lambda_C}^{-1/2}$. Hence, one sees that $\mathbf{T} = \mathbf{X}\hat{\mathbf{U}}_\mathbf{C}$, consistent with equation (5). Thus, classical MDS yields the same result as PCA in figure 2(a).

2.2.3. Linear regression

In linear regression, one aims to determine a set of weights $\mathbf{P}_{XY}$ to minimise the error between the true properties $\mathbf{Y}$ and the properties predicted via $\hat{\mathbf{Y}} = \mathbf{XP}_{XY}$, which is equivalent to minimising the loss

$$\ell_{\text{regr}} = \|\mathbf{Y} - \mathbf{XP}_{XY}\|^2/n_{\text{samples}} \tag{9}$$

In the following, we consider the case of an $\mathcal{L}^2$ regularised regression with regularisation parameter λ, i.e. ridge regression [32]. The loss to be minimised is

$$\ell = \|\mathbf{Y} - \mathbf{XP}_{XY}\|^2 + \lambda\|\mathbf{P}_{XY}\|^2. \tag{10}$$

Minimising the loss with respect to $\mathbf{P}_{XY}$ yields the solution $\mathbf{P}_{XY} = \left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\mathbf{X}^T\mathbf{Y}$. If one chooses to perform the regression using the low-dimensional latent space $\mathbf{T} = \mathbf{XP}_{XT}$ and approximate $\mathbf{Y}$ with $\mathbf{TP}_{TY}$, then $\mathbf{P}_{TY} = \left(\mathbf{T}^T\mathbf{T} + \lambda\mathbf{I}\right)^{-1}\mathbf{T}^T\mathbf{Y}$.

The ridge regression of the CSD-1000r dataset is shown in figure 2(b). Given the small train set size, and the difficulty of fitting simultaneously different elements with shieldings across a large range (≈800 ppm), the model achieves a very good accuracy, with an RMSE below 23 ppm.

2.3. Principal covariates regression

Principal covariates regression (PCovR) [18] utilises a combination between a PCA-like and an LR-like loss, and therefore attempts to find a low-dimensional projection of the feature vectors that simultaneously minimises information loss and error in predicting the target properties using only the latent space vectors **T**. A mixing parameter α determines the relative weight given to the PCA and LR tasks,

$$\ell = \frac{\alpha}{n_{\text{samples}}}\|\mathbf{X} - \mathbf{X}\mathbf{P}_{XT}\mathbf{P}_{TX}\|^2 + \frac{(1-\alpha)}{n_{\text{samples}}}\|\mathbf{Y} - \mathbf{X}\mathbf{P}_{XT}\mathbf{P}_{TY}\|^2. \tag{11}$$

The derivation we report here, albeit in our notation, follows closely that in the original article [18]. PCovR can be implemented in a way that diagonalises a modified Gram matrix (sample-space PCovR) or in a way that requires computing and diagonalising a modified covariance (feature-space PCovR). The two approaches yield the same latent-space projections, and which one should be used depends on the relative magnitudes of n_{samples} and n_{features}.

2.3.1. Sample-space PCovR

It is easier to minimise equation (11) by looking for a projection $\tilde{\mathbf{T}}$ in an auxiliary latent space for which we enforce orthonormality, $\tilde{\mathbf{T}}^T\tilde{\mathbf{T}} = \mathbf{I}$, known as a *whitened* projection.

This allows us to write $\mathbf{P}_{\tilde{T}X} = \tilde{\mathbf{T}}^T\mathbf{X}$ and $\mathbf{P}_{\tilde{T}Y} = \tilde{\mathbf{T}}^T\mathbf{Y}$. By definition $\tilde{\mathbf{T}} = \mathbf{X}\mathbf{P}_{X\tilde{T}}$, thus we can express the loss as

$$\ell = \alpha\|\mathbf{X} - \tilde{\mathbf{T}}\tilde{\mathbf{T}}^T\mathbf{X}\|^2 + (1-\alpha)\|\mathbf{Y} - \tilde{\mathbf{T}}\tilde{\mathbf{T}}^T\mathbf{Y}\|^2. \tag{12}$$

This loss is minimised by maximising the associated similarity

$$\rho = \operatorname{Tr}\left(\alpha\tilde{\mathbf{T}}\tilde{\mathbf{T}}^T\mathbf{X}\mathbf{X}^T + (1-\alpha)\tilde{\mathbf{T}}\tilde{\mathbf{T}}^T\hat{\mathbf{Y}}\hat{\mathbf{Y}}^T\right) \tag{13}$$

$$= \operatorname{Tr}\left(\alpha\tilde{\mathbf{T}}\tilde{\mathbf{T}}^T\mathbf{X}\mathbf{X}^T + (1-\alpha)\tilde{\mathbf{T}}\tilde{\mathbf{T}}^T\mathbf{X}\mathbf{P}_{XY}\mathbf{P}_{XY}^T\mathbf{X}^T\right), \tag{14}$$

where we have substituted **Y** with the regression approximation $\hat{\mathbf{Y}} = \mathbf{X}\mathbf{P}_{XY}$, given that a linear approximation of **Y** in the latent space can only, at best, reproduce the part of the properties that can be represented in the full feature space. If we define the modified Gram matrix

$$\tilde{\mathbf{K}} = \alpha\mathbf{X}\mathbf{X}^T + (1-\alpha)\mathbf{X}\mathbf{P}_{XY}\mathbf{P}_{XY}^T\mathbf{X}^T, \tag{15}$$

we can further write the similarity as

$$\rho = \operatorname{Tr}\left(\tilde{\mathbf{T}}^T\tilde{\mathbf{K}}\tilde{\mathbf{T}}\right). \tag{16}$$

The latent space projections $\tilde{\mathbf{T}}$ that maximise the similarity correspond to the principal eigenvectors of the matrix $\tilde{\mathbf{K}}$, $\tilde{\mathbf{T}} = \hat{\mathbf{U}}_{\tilde{\mathbf{K}}}$. By analogy with multidimensional scaling—and to ensure that in the limit of $\alpha \to 1$ we obtain the same latent space as in classical MDS—one can obtain de-whitened projections $\mathbf{T} = \hat{\mathbf{U}}_{\tilde{\mathbf{K}}}\hat{\mathbf{\Lambda}}_{\tilde{\mathbf{K}}}^{1/2} = \tilde{\mathbf{K}}\hat{\mathbf{U}}_{\tilde{\mathbf{K}}}\hat{\mathbf{\Lambda}}_{\tilde{\mathbf{K}}}^{-1/2}$, reminiscent of equation (8). The projector from feature space to the latent space is then given by

$$\mathbf{P}_{XT} = \left(\alpha\mathbf{X}^T + (1-\alpha)\mathbf{P}_{XY}\mathbf{P}_{XY}^T\mathbf{X}^T\right)\hat{\mathbf{U}}_{\tilde{\mathbf{K}}}\hat{\mathbf{\Lambda}}_{\tilde{\mathbf{K}}}^{-1/2}. \tag{17}$$

The projector matrix from the latent space to the properties **Y** can be computed from the LR solution

$$\mathbf{P}_{TY} = \left(\mathbf{T}^T\mathbf{T} + \lambda\mathbf{I}\right)^{-1}\mathbf{T}^T\mathbf{Y} \underset{\lambda\to 0}{=} \hat{\mathbf{\Lambda}}_{\tilde{\mathbf{K}}}^{-1/2}\hat{\mathbf{U}}_{\tilde{\mathbf{K}}}^T\mathbf{Y}. \tag{18}$$

2.3.2. Feature-space PCovR

Rather than determining the optimal PCovR projections by diagonalising the equivalent of a Gram matrix, one can tackle the problem in a way that more closely resembles PCA by instead diagonalising a modified covariance matrix. Given that $\mathbf{I} = \tilde{\mathbf{T}}^T\tilde{\mathbf{T}} = \mathbf{P}_{X\tilde{T}}^T\mathbf{X}^T\mathbf{X}\mathbf{P}_{X\tilde{T}} = \mathbf{P}_{X\tilde{T}}^T\mathbf{C}\mathbf{P}_{X\tilde{T}}$, we see that $\mathbf{C}^{1/2}\mathbf{P}_{X\tilde{T}}$ is orthogonal. We can thus rewrite the similarity function from equation (16) as

$$\rho = \operatorname{Tr}\left(\mathbf{P}_{X\tilde{T}}^T\mathbf{C}^{1/2}\tilde{\mathbf{C}}\mathbf{C}^{1/2}\mathbf{P}_{X\tilde{T}}\right), \tag{19}$$

introducing

$$\begin{aligned}\tilde{\mathbf{C}} =&\mathbf{C}^{-1/2}\mathbf{X}^T\tilde{\mathbf{K}}\mathbf{X}\mathbf{C}^{-1/2}\\ =&\alpha\mathbf{C}+(1-\alpha)\mathbf{C}^{-1/2}\mathbf{X}^T\hat{\mathbf{Y}}\hat{\mathbf{Y}}^T\mathbf{X}\mathbf{C}^{-1/2}\\ =&\mathbf{U}_{\tilde{\mathbf{C}}}\mathbf{\Lambda}_{\tilde{\mathbf{C}}}\mathbf{U}_{\tilde{\mathbf{C}}}^T.\end{aligned} \tag{20}$$

The similarity is maximised when the orthogonal matrix $\mathbf{C}^{1/2}\mathbf{P}_{X\tilde{T}}$ matches the principal eigenvalues of $\tilde{\mathbf{C}}$, i.e. $\mathbf{P}_{X\tilde{T}}=\mathbf{C}^{-1/2}\hat{\mathbf{U}}_{\tilde{\mathbf{C}}}$. In general $\mathbf{P}_{X\tilde{T}}\mathbf{P}_{\tilde{T}X}=\mathbf{C}^{-1/2}\hat{\mathbf{U}}_{\tilde{\mathbf{C}}}\hat{\mathbf{U}}_{\tilde{\mathbf{C}}}^T\mathbf{C}^{1/2}$ is not a symmetric matrix, and so it is not possible to define an orthonormal $\mathbf{P}_{XT}$ such that $\mathbf{P}_{TX}=\mathbf{P}_{XT}^T$. Consistently with the case of sample-space PCovR, we obtain

$$\begin{aligned}\mathbf{P}_{XT} =&\mathbf{C}^{-1/2}\hat{\mathbf{U}}_{\tilde{\mathbf{C}}}\hat{\mathbf{\Lambda}}_{\tilde{\mathbf{C}}}^{1/2}\\ \mathbf{P}_{TX} =&\hat{\mathbf{\Lambda}}_{\tilde{\mathbf{C}}}^{-1/2}\hat{\mathbf{U}}_{\tilde{\mathbf{C}}}^T\mathbf{C}^{1/2}\\ \mathbf{P}_{TY} =&\hat{\mathbf{\Lambda}}_{\tilde{\mathbf{C}}}^{-1/2}\hat{\mathbf{U}}_{\tilde{\mathbf{C}}}^T\mathbf{C}^{-1/2}\mathbf{X}^T\mathbf{Y},\end{aligned} \tag{21}$$

which minimise the PCovR loss in equation (11). These projections reduce to PCA as $\alpha\to 1$ and—if the dimension of the latent space is at least as large as the number of target properties in $\mathbf{Y}$—reduce to LR as $\alpha\to 0$.

Figure 3 demonstrates the behaviour of PCovR when applied to the analysis of the CSD-1000r dataset. Here we plot ℓ_{proj} and ℓ_{regr} as a function of α. The thumbnails above the losses correspond to the projections $\mathbf{T}=\mathbf{X}\mathbf{P}_{XT}$ and regressions $\hat{\mathbf{Y}}=\mathbf{T}\mathbf{P}_{TY}$ for the indicated α below.

For $\alpha=0$, we recover the accuracy of pure LR in predicting the values of the chemical shielding, but obtain a latent space that misses completely the structure of the dataset. The first principal component reflects the LR weight vector $\mathbf{P}_{XY}$, and the second carries no meaningful information. For $\alpha=1$, we recover the PCA projection, which separates clearly the environments based on the nature of the central atom. A linear model built in the two-dimensional latent space, however, performs very poorly, because there is no *linear* correlation between the position in latent space and the shielding values. Intermediate values of α yield a projection that achieves the best of both worlds. The regression error is close to that of pure LR, but the error in the reconstruction of the input data from the latent space is now only marginally increased compared to pure PCA.

The PCovR map that corresponds to this 'optimal' value of α achieves $\ell_{\mathrm{proj}}=0.585$ (comparable to the PCA value of 0.460) and $\ell_{\mathrm{regr}}=0.112$ (comparable to the LR value of 0.111). Considering the poor performance of PCA in regression ($\ell_{\mathrm{regr}}=0.928$) and LR in projection ($\ell_{\mathrm{proj}}=0.963$), it is clear that the latent-space description of the dataset achieves a more versatile representation of structure-property relations. There is still a recognisable clustering of the environments according to central atom species, but the O cluster, that exhibits the largest variance in the values of the shielding, is spread out diagonally to achieve maximal correlation between the position in latent space and value of the target properties. We propose that an optimal value of α can be obtained looking for the minimum in $\ell_{\mathrm{proj}}+\ell_{\mathrm{regr}}$, but depending on the specifics of the problem at hand, one can also give more emphasis to either of the terms or choose an entirely different criterion to select the most suitable α.

2.4. Kernel methods

While linear methods have the beauty of simplicity, they rely on the knowledge of a sufficient number of informative features that reflect the relation between inputs and properties. Kernel methods introduce a possibly non-linear relation between samples in the form of a positive-definite kernel function $k(\mathbf{X},\mathbf{X}')$ (e.g. the Gaussian kernel $\exp(-\|\mathbf{X}-\mathbf{X}'\|^2)$, or the linear kernel $\mathbf{X}\cdot\mathbf{X}'$), and use it to define a higher-dimensional space in which data points serve effectively as an adaptive basis [37]. Unless otherwise specified, here we use a radial basis function (RBF) kernel, $\exp(-\gamma\|\mathbf{X}-\mathbf{X}'\|^2)$, with the hyperparameter γ optimized for each data set, as shown in the SI. Doing so can help uncover non-linear relationships between the samples, resulting ultimately in a more effective determination of a low-dimensional latent space and increased regression performance.

Mercer's theorem [38] guarantees that, given a positive-definite kernel, there is a linear operator $\phi(\mathbf{X})$ that maps input features into a (possibly infinite-dimensional) reproducing kernel Hilbert space (RKHS) [37] whose scalar product generates the kernel, i.e. $\phi(\mathbf{X})\cdot\phi(\mathbf{X}')=k(\mathbf{X},\mathbf{X}')$. $\phi(\mathbf{X})$ is not necessarily known explicitly, but as we will see it can be approximated effectively for a given dataset, and we will use the notation $\mathbf{\Phi}$ to indicate the feature matrix that contains the (approximate) values of the kernel features for all of the sample points. We indicate with $\mathbf{K}=\mathbf{\Phi}\mathbf{\Phi}^T$ the $n_{\mathrm{samples}}\times n_{\mathrm{samples}}$ matrix that contains as entries the values of the kernel function between every pair of samples. In the case of a linear kernel, this is simply the Gram matrix computed for the input features, while for a non-linear kernel its entries can be computed by evaluating the kernel between pairs of samples, $K_{ij}=k(\mathbf{X}_i,\mathbf{X}_j)$. Analogously to what we did for linear methods, we centre and normalise all the kernels we use in this work. Some subtleties connected to the centring operation are discussed in appendix A.

2.4.1. Kernel principal component analysis

Kernel principal component analysis [31] proceeds parallel to classical MDS, to which it corresponds exactly when a linear kernel is used. To construct a KPCA decomposition, one computes the eigendecomposition of the kernel matrix $\mathbf{K} = \mathbf{U}_\mathbf{K}\mathbf{\Lambda}_\mathbf{K}\mathbf{U}_\mathbf{K}^T$ and defines the projections as the principal components $\mathbf{T} = \hat{\mathbf{U}}_\mathbf{K}\hat{\mathbf{\Lambda}}_\mathbf{K}^{1/2}$[1]. The projections can also be computed as $\mathbf{T} = \mathbf{K}\mathbf{P}_{KT} = \mathbf{K}\hat{\mathbf{U}}_\mathbf{K}\hat{\mathbf{\Lambda}}_\mathbf{K}^{-1/2}$, and this second expression can be used to project new data (in place of $\mathbf{K}$ we use the matrix containing the values of the kernel matrix between new and reference points) in the approximate RKHS defined by the original samples. One can also approximate the kernel using the projector $\mathbf{P}_{TK} = \hat{\mathbf{\Lambda}}_\mathbf{K}^{-1}\mathbf{T}^T\mathbf{K}$. As shown in figure 2(c), for this dataset there is little qualitative difference between what we obtained with plain PCA and the KPCA projection. This is because SOAP features exhibit very clear correlations with the nature of the central environments, which is already well represented with a linear model. While it is possible to compute the loss $\ell_\text{proj} = \|\mathbf{X} - \mathbf{T}\mathbf{P}_{TX}\|^2/n_\text{samples}$ associated with the approximation of $\mathbf{X}$ based on $\mathbf{T}$, KPCA aims to approximate the kernel, and it is more appropriate to judge the method 's performance based on a Gram loss $\ell_\text{gram} = \|\mathbf{K} - \mathbf{T}\mathbf{T}^T\|^2/n_\text{samples}$, which reduces to equation (7) for linear kernels. Alternatively, one can compute a projection loss based on the approximation of the RHKS features, $\ell_\text{proj} = \|\mathbf{\Phi} - \mathbf{T}\mathbf{P}_{T\Phi}\|^2/n_\text{samples}$, as detailed in appendix B.

2.4.2. Kernel ridge regression

Kernel ridge regression [39, 40] is analogous to ridge regression, except that the kernel feature space vectors $\mathbf{\Phi}$ are substituted for the original input data $\mathbf{X}$, giving the loss

$$\ell = \|\mathbf{Y} - \mathbf{\Phi}\mathbf{P}_{\Phi Y}\|^2 + \lambda\|\mathbf{P}_{\Phi Y}\|^2, \tag{22}$$

so that the optimal weights are

$$\begin{aligned}\mathbf{P}_{\Phi Y} &= \left(\mathbf{\Phi}^T\mathbf{\Phi} + \lambda\mathbf{I}\right)^{-1}\mathbf{\Phi}^T\mathbf{Y}\\ &= \mathbf{\Phi}^T\left(\mathbf{\Phi}\mathbf{\Phi}^T + \lambda\mathbf{I}\right)^{-1}\mathbf{Y}.\end{aligned} \tag{23}$$

Predicted properties $\hat{\mathbf{Y}}$ can then be evaluated with $\hat{\mathbf{Y}} = \mathbf{\Phi}\mathbf{P}_{\Phi Y}$. One can avoid computing explicitly the RKHS features by redefining the weights as $\mathbf{P}_{KY} = \left(\mathbf{\Phi}\mathbf{\Phi}^T + \lambda\mathbf{I}\right)^{-1}\mathbf{Y} = (\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{Y}$ so that $\mathbf{P}_{\Phi Y} = \mathbf{\Phi}^T\mathbf{P}_{KY}$ [41]. We can then write the predicted properties as

$$\hat{\mathbf{Y}} = \mathbf{\Phi}\mathbf{\Phi}^T\mathbf{P}_{KY} = \mathbf{K}\mathbf{P}_{KY}. \tag{24}$$

As shown in figure 2(d), the greater flexibility afforded by a kernel model reduces the error by over 70%.

2.5. Sparse kernel methods

Since the size of kernel matrices grows in n^2 with respect to the number of samples, one wants to avoid computing (and inverting) the whole kernel matrix for large datasets. Instead, we can formulate a low-rank approximation to the kernel matrix through the Nyström approximation [42], using a sub-selection of the data points, the active set, to define an approximate RKHS. These representative points can be selected in a variety of ways; two straightforward methods that have been used successfully in atomistic modelling are farthest point sampling (FPS) [43] and a CUR matrix decomposition [44–46].

Using the subscript N to represent the full set of training data and M to indicate the active set, one can explicitly construct the approximate feature matrix as $\mathbf{\Phi}_{NM} = \mathbf{K}_{NM}\mathbf{U}_{\mathbf{K}_{MM}}\mathbf{\Lambda}_{\mathbf{K}_{MM}}^{-1/2}$, where $\mathbf{U}_{MM}$ and $\mathbf{\Lambda}_{MM}$ are from the eigendecomposition of $\mathbf{K}_{MM}$. All sparse kernel methods can be derived in terms of a linear method based on the RKHS, although it is often possible to avoid explicitly computing $\mathbf{\Phi}_{NM}$. For instance, the approximate kernel matrix takes the form [42],

$$\mathbf{K} \approx \hat{\mathbf{K}}_{NN} = \mathbf{\Phi}_{NM}\mathbf{\Phi}_{NM}^T = \mathbf{K}_{NM}\mathbf{K}_{MM}^{-1}\mathbf{K}_{NM}^T. \tag{25}$$

For the following methods, we consider the approximate feature matrix $\mathbf{\Phi}_{NM}$ to be centred and scaled as discussed in appendix A.

[1] If one retains all the n_samples eigenvectors, $\mathbf{T}$ corresponds to an exact approximation of the kernel features for the given dataset, as $\mathbf{T}\mathbf{T}^T = \mathbf{K} = \mathbf{\Phi}\mathbf{\Phi}^T$.

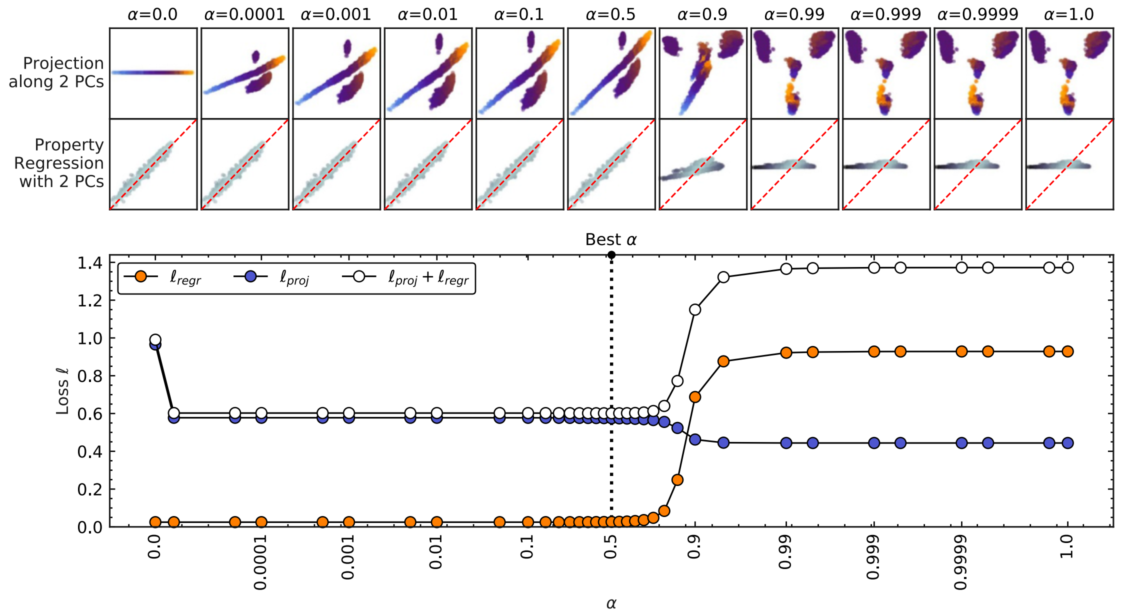

Figure 4. Kernel Principal Covariates Regression of CSD-1000r. Combining kernel ridge regression (far left) and KPCA (far right) with mixing parameter α. The combined loss $\ell = \ell_{\text{proj}} + \ell_{\text{regr}} = \|\mathbf{\Phi} - \mathbf{T}\mathbf{P}_{T\Phi}\|^2/n_{\text{samples}} + \|\mathbf{Y} - \mathbf{T}\mathbf{P}_{TY}\|^2/n_{\text{samples}}$ is given in the lower panel. Similar to figure 3, the upper panels show the resulting projections and regressions at the indicated α value, aligned with the horizontal axis in the lower plot. Colour mappings again correspond to those in figure 2 and figure 5. An animated representation of the top panels can be found in the SI.

2.5.1. Sparse kernel principal component analysis

We can define the covariance in the kernel feature space along with its eigendecomposition,

$$\mathbf{C} = \mathbf{\Phi}_{NM}^T\mathbf{\Phi}_{NM} = \mathbf{U}_\mathbf{C}\mathbf{\Lambda}_\mathbf{C}\mathbf{U}_\mathbf{C}^T, \tag{26}$$

and subsequently compute the projections analogously to standard KPCA

$$\mathbf{T} = \mathbf{\Phi}_{NM}\hat{\mathbf{U}}_\mathbf{C} = \mathbf{K}_{NM}\mathbf{U}_{\mathbf{K}_{MM}}\mathbf{\Lambda}_{\mathbf{K}_{MM}}^{-1/2}\hat{\mathbf{U}}_\mathbf{C} = \mathbf{K}_{NM}\mathbf{P}_{KT}, \tag{27}$$

which effectively determine the directions of maximum variance of the samples in the active RHKS.

Figure 2(e) shows that with an active set size of just 50 samples (out of more than 12 000), selected by FPS [46], one can obtain a KPCA latent projection that matches very well the qualitative features of the full KPCA construction.

2.5.2. Sparse kernel ridge regression

In sparse KRR we proceed as in standard KRR, but use the feature matrix from the Nyström approximation. The corresponding regularised LR loss in the kernel feature space is

$$\ell = \|\mathbf{Y} - \mathbf{\Phi}_{NM}\mathbf{P}_{\Phi Y}\|^2 + \lambda\|\mathbf{P}_{\Phi Y}\|^2 \tag{28}$$

for which the solution is

$$\begin{aligned}\mathbf{P}_{\Phi Y} &= \left(\mathbf{\Phi}_{NM}^T\mathbf{\Phi}_{NM} + \lambda\mathbf{I}\right)^{-1}\mathbf{\Phi}_{NM}^T\mathbf{Y} \\ &= \left(\mathbf{\Phi}_{NM}^T\mathbf{\Phi}_{NM} + \lambda\mathbf{I}\right)^{-1}\mathbf{\Lambda}_{\mathbf{K}_{MM}}^{-1/2}\mathbf{U}_{\mathbf{K}_{MM}}^T\mathbf{K}_{NM}^T\mathbf{Y}.\end{aligned} \tag{29}$$

Alternatively, we can redefine the weights so that

$$\hat{\mathbf{Y}} = \mathbf{\Phi}_{NM}\mathbf{P}_{\Phi Y} = \mathbf{K}_{NM}\mathbf{U}_{\mathbf{K}_{MM}}\mathbf{\Lambda}_{\mathbf{K}_{MM}}^{-1/2}\mathbf{P}_{\Phi Y} = \mathbf{K}_{NM}\mathbf{P}_{KY}, \tag{30}$$

from which we see that

$$\begin{aligned}\mathbf{P}_{KY} &= \mathbf{U}_{\mathbf{K}_{MM}}\mathbf{\Lambda}_{\mathbf{K}_{MM}}^{-1/2}\mathbf{P}_{\Phi Y} \\ &= \mathbf{U}_{\mathbf{K}_{MM}}\mathbf{\Lambda}_{\mathbf{K}_{MM}}^{-1/2}\left(\mathbf{\Phi}_{NM}^T\mathbf{\Phi}_{NM} + \lambda\mathbf{I}\right)^{-1} \\ &\quad \times \mathbf{\Lambda}_{\mathbf{K}_{MM}}^{-1/2}\mathbf{U}_{\mathbf{K}_{MM}}^T\mathbf{K}_{NM}^T\mathbf{Y}.\end{aligned} \tag{31}$$

By writing out explicitly $\mathbf{\Phi}_{NM}^T\mathbf{\Phi}_{NM}$ in terms of $\mathbf{K}_{NM}$ we obtain [40],

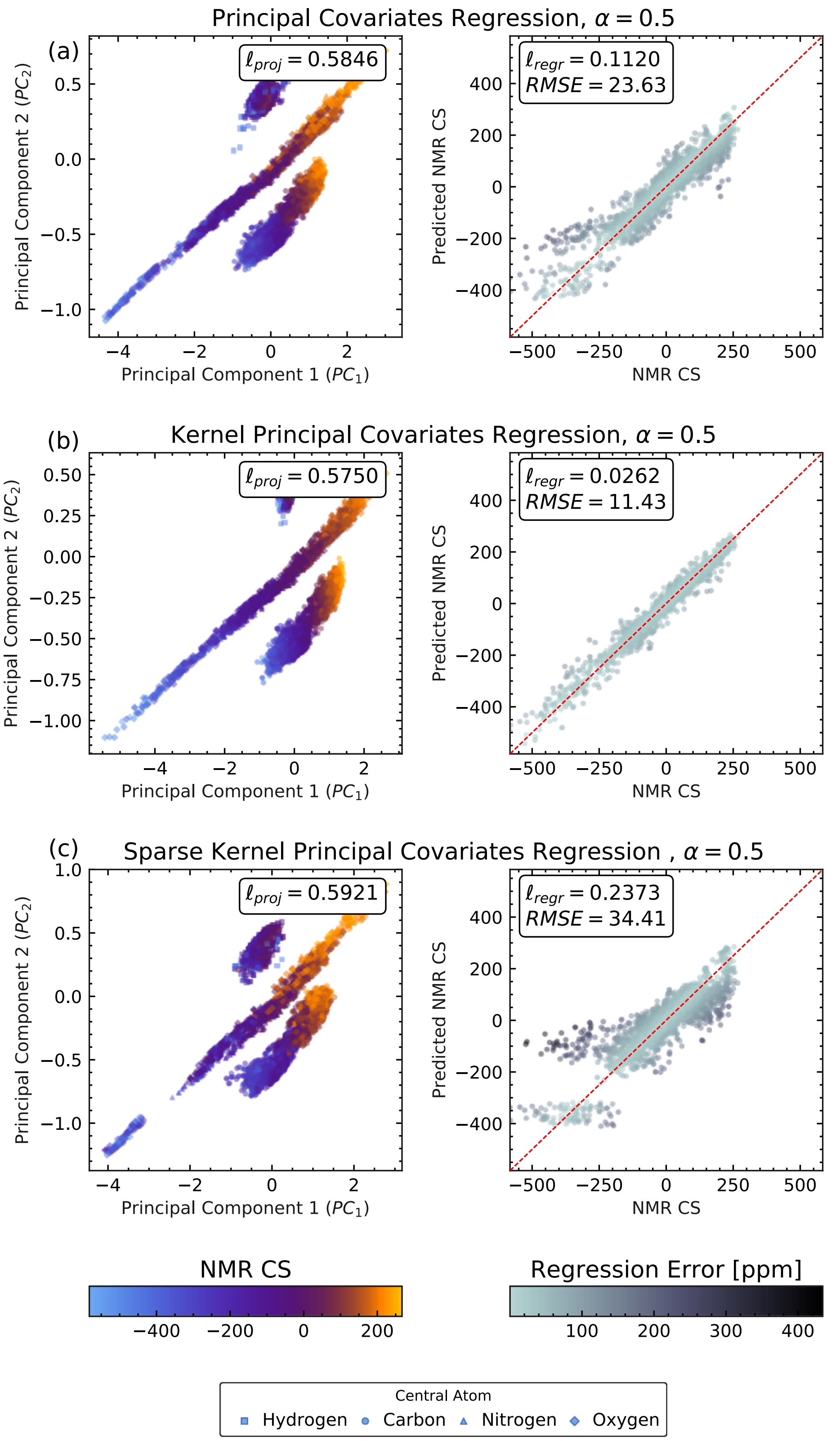

Figure 5. Projections and Regression from PCovR Models. Analogous to figure 2, we show the loss incurred from the projections and regression of the three PCovR models at $\alpha = 0.5$. (a) Principal Covariates Regression (PCovR), (b) Kernel PCovR (KPCovR), and (c) Sparse KPCovR with $n_{active} = 50$. α was chosen to best compare the models, although the ideal α may fluctuate between models, albeit often with a range of suitable α, as in figure 3. The inset includes the ℓ_{proj} and $\ell_{\mathrm{regr}} = \|\mathbf{Y} - \mathbf{T}\mathbf{P}_{TY}\|^2/n_{\mathrm{samples}}$.

$$\mathbf{P}_{KY} = \left(\mathbf{K}_{NM}^T\mathbf{K}_{NM} + \lambda\mathbf{K}_{MM}\right)^{-1}\mathbf{K}_{NM}^T\mathbf{Y}. \tag{32}$$

As shown in figure 2(f), an active set size of 50 is not sufficient to achieve an accurate regression model, and the error is larger than with a linear regression method. However, the error can be reduced systematically by increasing the size of the active set, finding the best balance between accuracy and cost (see SI).

3. Extensions to principal covariates regression

After having summarised existing linear and kernel methods for feature approximation and property prediction, we now introduce kernelized PCovR (KPCovR), as a way to combine the conceptual framework of PCovR and the non-linear features afforded by a kernel method.

3.1. Full kernel PCovR

We start by constructing the augmented kernel matrix as a combination of KPCA and KRR. In particular, we substitute $\mathbf{\Phi}$ for $\mathbf{X}$ and the KRR solution of $\mathbf{Y}$, $\hat{\mathbf{Y}} = \mathbf{K}\left(\mathbf{K}+\lambda\mathbf{I}\right)^{-1}\mathbf{Y}$, for $\mathbf{Y}$, so that we have

$$\tilde{\mathbf{K}} = \alpha\mathbf{K} + (1-\alpha)\hat{\mathbf{Y}}\hat{\mathbf{Y}}^T, \tag{33}$$

where we consider the kernel matrix to be standardised in a way that is equivalent to normalising $\mathbf{\Phi}$ (see appendix A). Just as in PCovR, the unit variance projections $\tilde{\mathbf{T}}$ are given by the top eigenvectors $\hat{\mathbf{U}}_{\tilde{\mathbf{K}}}$ of $\tilde{\mathbf{K}}$, and the non-whitened projections as $\mathbf{T} = \hat{\mathbf{U}}_{\tilde{\mathbf{K}}}\hat{\mathbf{\Lambda}}_{\tilde{\mathbf{K}}}^{1/2} = \tilde{\mathbf{K}}\hat{\mathbf{U}}_{\tilde{\mathbf{K}}}\hat{\mathbf{\Lambda}}_{\tilde{\mathbf{K}}}^{-1/2}$, corresponding to the RKHS $\tilde{\mathbf{\Phi}}$ associated with the PCovR kernel (equation (33)).

Projecting a new set of structures in the kernel PCovR space entails computing the RHKS between the samples that were originally used to determine the KPCovR features and the new samples. Given that one may not want to compute these explicitly, it is useful to define a projection acting directly on the kernel, such that $\mathbf{T} = \mathbf{K}\mathbf{P}_{KT}$:

$$\mathbf{P}_{KT} = \left(\alpha\mathbf{I} + (1-\alpha)\left(\mathbf{K}+\lambda\mathbf{I}\right)^{-1}\mathbf{Y}\hat{\mathbf{Y}}^T\right)\hat{\mathbf{U}}_{\tilde{\mathbf{K}}}\hat{\mathbf{\Lambda}}_{\tilde{\mathbf{K}}}^{-1/2}. \tag{34}$$

We also determine the matrix that enables predictions of properties from the latent space $\mathbf{T}$ through LR, just as in the linear case (equation (18)). Computing the projection loss minimised by KPCovR, $\ell_{\text{proj}} = \|\mathbf{\Phi} - \mathbf{T}\mathbf{P}_{T\Phi}\|^2/n_{\text{samples}}$, is trivial if one computes explicitly a RKHS approximation of $\mathbf{\Phi}$, but it requires some work if one wants to avoid evaluating $\mathbf{\Phi}$ (see appendix B).

As shown in figure 4, the method combines a behaviour similar to linear PCovR with the improved property prediction accuracy afforded by kernel methods. In the low-α regime the regression accuracy approaches that of KRR, and the projection accuracy converges to a KPCA-like behaviour for $\alpha \approx 1$. For the optimal value of α, the latent-space map (shown in figure 5(b)) combines a clear separation of structurally-distinct clusters with a 70% reduction in regression error when compared to linear PCovR ($\ell_{\text{regr}} = 0.026$ vs. $\ell_{\text{regr}} = 0.112$).

3.2. Sparse kernel PCovR

Our derivation of the sparse version of KPCovR can be obtained almost directly from that of feature-space PCovR by taking explicitly the projection of the kernel on the active RKHS $\mathbf{\Phi}_{NM} = \mathbf{K}_{NM}\mathbf{U}_{\mathbf{K}_{MM}}\mathbf{\Lambda}_{\mathbf{K}_{MM}}^{-1/2}$. One can then define the covariance of the active kernel features

$$\begin{aligned}\mathbf{C} &= \mathbf{\Phi}_{NM}^T\mathbf{\Phi}_{NM} && (35)\\ &= \mathbf{\Lambda}_{\mathbf{K}_{MM}}^{-1/2}\mathbf{U}_{\mathbf{K}_{MM}}^T\mathbf{K}_{NM}^T\mathbf{K}_{NM}\mathbf{U}_{\mathbf{K}_{MM}}\mathbf{\Lambda}_{\mathbf{K}_{MM}}^{-1/2}, && (36)\end{aligned}$$

and use it in the definition of the modified KPCovR covariance

$$\begin{aligned}\tilde{\mathbf{C}} =& \alpha\mathbf{C} + (1-\alpha)\mathbf{C}^{1/2}\left(\mathbf{C}+\lambda\mathbf{I}\right)^{-1}\mathbf{\Lambda}_{\mathbf{K}_{MM}}^{-1/2}\mathbf{U}_{\mathbf{K}_{MM}}^T\mathbf{K}_{NM}^T\mathbf{Y}\\ &\times\mathbf{Y}^T\mathbf{K}_{NM}\mathbf{U}_{\mathbf{K}_{MM}}\mathbf{\Lambda}_{\mathbf{K}_{MM}}^{-1/2}\left(\mathbf{C}+\lambda\mathbf{I}\right)^{-1}\mathbf{C}^{1/2}.\end{aligned} \tag{37}$$

With these definitions, the projection matrices onto the (sparse) KPCovR latent space and onto the latent-space-restricted properties, analogous to equation (21), read

$$\begin{aligned}\mathbf{P}_{KT} =& \mathbf{U}_{\mathbf{K}_{MM}}\mathbf{\Lambda}_{\mathbf{K}_{MM}}^{-1/2}\mathbf{C}^{-1/2}\hat{\mathbf{U}}_{\tilde{\mathbf{C}}}\hat{\mathbf{\Lambda}}_{\tilde{\mathbf{C}}}^{1/2}\\ \mathbf{P}_{TY} =& \hat{\mathbf{\Lambda}}_{\tilde{\mathbf{C}}}^{-1/2}\hat{\mathbf{U}}_{\tilde{\mathbf{C}}}^T\mathbf{C}^{-1/2}\mathbf{\Lambda}_{\mathbf{K}_{MM}}^{-1/2}\mathbf{U}_{\mathbf{K}_{MM}}^T\mathbf{K}_{NM}^T\mathbf{Y}.\end{aligned} \tag{38}$$

Table 1. Performance of PCovR and KPCovR for the different examples using ℓ_{regr} and ℓ_{proj}. The α which minimizes the total loss $\ell = \ell_{\text{regr}} + \ell_{\text{proj}}$ is given by α^*. Results of optimal-alpha KPCovR are always comparable or better than those from the linear PCovR version. Values that are improved by more than 10% are highlighted in bold.

				PCovR				KPCovR			
Dataset	n_{latent}	N	ℓ	$\alpha = 0.0$	$\alpha = \alpha^*$	$\alpha = 1.0$	α^*	$\alpha = 0.0$	$\alpha = \alpha^*$	$\alpha = 1.0$	α^*
CSD-1000 R	2	25 600	ℓ_{regr}	0.1106	0.112	0.9285	0.5	0.0249	**0.0262**	0.9285	0.5
			ℓ_{proj}	0.9633	0.5846	0.4586		0.9656	0.575	0.4439	
C-VII	2	10 874	ℓ_{regr}	0.0707	0.0753	0.9447	0.5	0.0133	**0.0172**	0.9445	0.5
			ℓ_{proj}	0.9572	0.4443	0.2384		0.9607	0.4515	0.2398	
Deem (global)	2	4 000	ℓ_{regr}	0.065	0.1178	0.6082	0.5	0.0599	0.1143	0.608	0.5
			ℓ_{proj}	0.5788	0.4288	0.2443		0.5652	0.4104	0.2183	
Deem (local)	2	10 968	ℓ_{regr}	0.365	0.0841	0.7279	0.5	0.0015	**0.0464**	0.7163	0.5
			ℓ_{proj}	0.7629	0.6531	0.4258		0.7652	0.6648	0.4577	
QM9	2	10 000	ℓ_{regr}	0.3298	0.3905	0.4789	0.45	0.3135	0.3891	0.4789	0.5
			ℓ_{proj}	0.7419	0.5634	0.5361		0.6281	**0.3677**	0.3407	
QM9	12	10 000	ℓ_{regr}	0.1212	0.1296	0.2938	0.4	0.0744	**0.083**	0.2938	0.55
			ℓ_{proj}	0.4511	0.3493	0.3287		0.2686	**0.0822**	0.0459	
Arginine Dipeptide	2	4 217	ℓ_{regr}	0.0122	0.0131	0.6067	0.55	0.0034	**0.0045**	0.6058	0.55
			ℓ_{proj}	0.824	0.549	0.435		0.8109	0.5282	0.4083	
Azaphenacenes	2	311	ℓ_{regr}	0.4632	0.5537	0.8929	0.6	0.5181	0.5582	0.8834	0.65
			ℓ_{proj}	0.8295	0.5113	0.3742		0.8583	0.4689	0.3124	

Similar to what we observed for sparse KPCA and KRR, reducing the active space to 50 active samples preserves the qualitative features of the latent-space map, but leads to substantial loss of performance for regression (figure 5(c)). The error, however, is equal to that observed for sparse KRR, which indicates that it is due to the limited active space size, and not by the dimensionality reduction.

4. Examples

Up until this point, we have talked about the definition of KPCovR in an abstract, equations-heavy manner. Here we demonstrate its usage for a wide range of materials science datasets. These datasets have all been already published elsewhere, and we leave to the SI a precise discussion of their structure, content and provenance, as well as a thorough analysis of the behaviour of the different linear and kernel methods applied to each data set. Here we limit ourselves to the most salient observations, and summarise the insights that could be relevant to the application of (K)PCovR to other materials and molecular datasets. We also make available data files at reference [47] that can be viewed with the interactive structure-property explorer chemiscope [48]. We encourage the reader to use them to gain a more interactive support to follow the discussion in this section.

For each dataset, we trained machine learning models on a randomly chosen half of the included samples, and then evaluated these models on the remaining samples. In the following section, we report losses, errors and figures on the validation set of points only. The total number of samples we considered are available in table 1, together with performance metrics that show that PCovR-like methods achieve consistently an excellent compromise between the optimisation of ℓ_{proj} and ℓ_{regr}, and demonstrate that KPCovR outperforms by a large margin its non-kernelized counterpart in terms of regression performance.

4.1. Carbon

We apply KPCovR to the C-VII carbon dataset, which contains roughly 11 000 carbon structures generated using *Ab Initio* Random Structure Searching (AIRSS) at 10GPa [49, 50]. Here, a KRR model predicts the average per-atom energy of each structure with an RMSE of 0.054 eV atom^{-1} (equivalent to $\ell_{\text{regr}} = 0.0133$) yet can only describe 4% of the latent space variance (*i.e.* $\ell_{\text{proj}} = 0.96$). The KPCA model retains 76% of the latent space variance but with an RMSE of 0.46 eV atom^{-1} ($\ell_{\text{regr}} = 0.9445$). By comparison, at the optimal α value, KPCovR sacrifices little in regression accuracy (0.062 eV atom^{-1}, $\ell_{\text{regr}} = 0.0172$) and latent space variance retention (55%).

Additionally, KPCovR provides a more intuitive qualitative picture for understanding the dataset. In the original KPCA projection, the principal components correlate strongly with the dimensionality of the carbon structures, with linear nanowires in the lower right of the projection, sheets and planar structures above to the left, and 3D structures conglomerated in the upper left. The KPCovR projection does not only show a much clearer correlation between the position on the map and the stability of each configuration, but also provides more compact clustering of similar structures (figure 6(a)). The nanowires (typically linear carbon

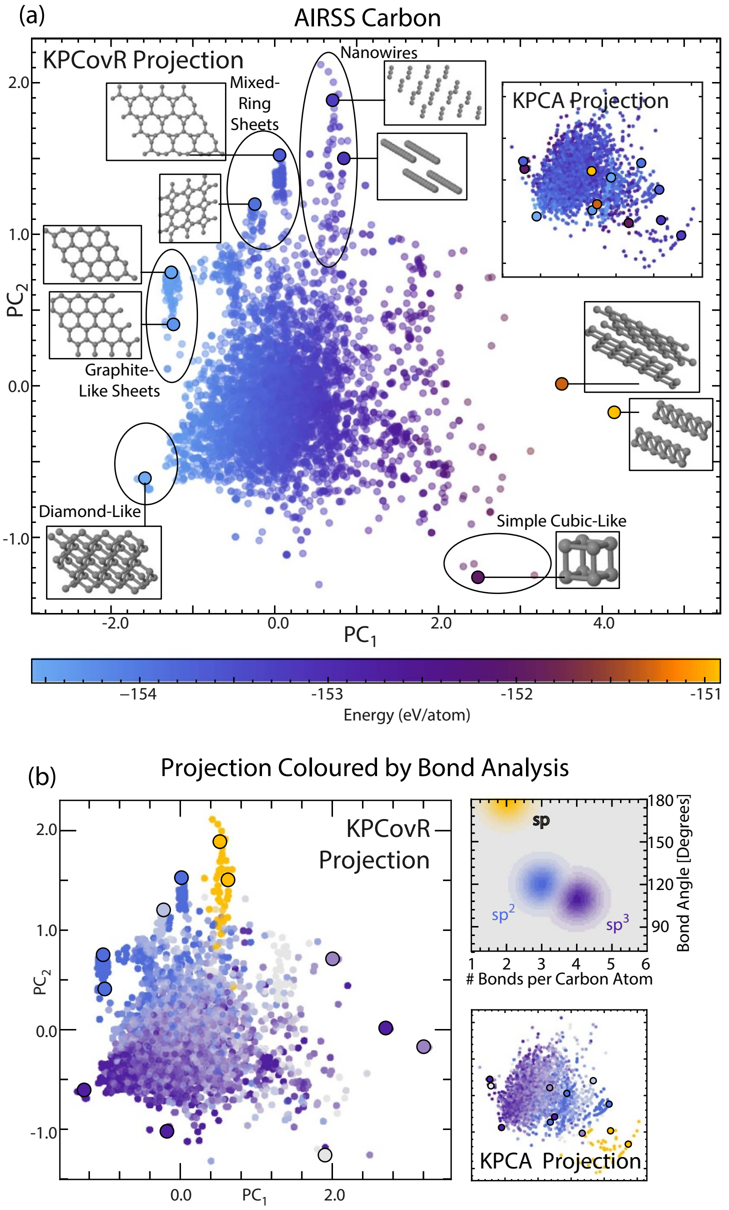

Figure 6. KPCovR for AIRSS Carbon Structures and Energies. (a) Projections coloured by the per-atom energy. KPCovR projection at $\alpha = 0.5$. Clusters identified using *chemiscope* have been labelled with representative snapshots provided. The KPCA projection is given in the upper right inset, with highlighted structures in the KPCovR projection denoted by enlarged points. (b) Projections in (a) recoloured by the bond analysis, where yellow, blue and purple denote similarity to *sp*, *sp*^{2}, and *sp*^{3} geometries, respectively, and grey signifying no resemblance.

bonds) are located in the upper centre, the planar structures are partitioned in smaller clusters to the upper left, with clear sub-clusters associated with different ring patterns, and 3D crystals are distributed throughout the lower left and centre, with a well-separated cluster for tetrahedral carbon. The structural homogeneity of the clusters is confirmed by bond analysis—typically, the most common environments found in low energy 1D, 2D, and 3D carbon structures correspond to *sp*, *sp*^{2}, and *sp*^{3} geometry. Angles between neighbouring bonds typically serve as a good proxy for detecting these environments, where the ideal values are 180°, 120°, 109.5°, respectively. Both KPCA and KPCovR detect clusters delineated by the number of bonds and bond angles (figure 6(b)), with these clusters arranged right-to-left in the KPCA projection, and top-to-bottom in the KPCovR projection. However, in the KPCovR projection, there is another gradient visible, with structures to the left of the projection more strongly coinciding with the ideal *sp*, *sp*^{2}, and *sp*^{3}, and those to the right being increasingly distorted[2]. Thus, bond analysis reveals that, in addition to the clear delineation between structure dimensionality provided by KPCA, the inclusion of an energy regression criterion in the

[2] The two notable exceptions–a dark purple cluster middle centre and a grey cluster upper left—can be shown to be associated with bond angle distributions which are multimodal, leading to incorrect classification by a criterion based on the *mean* bond angles.

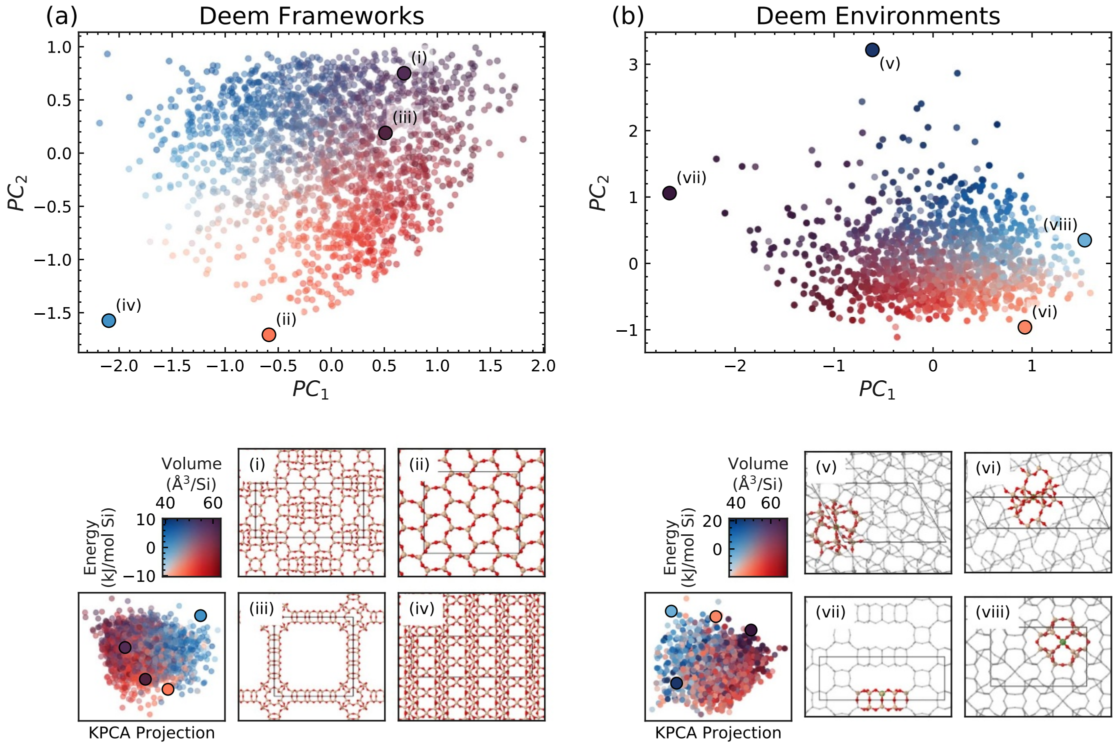

Figure 7. KPCovR for Zeolite Structures and Environments. Two-dimensional KPCovR projections at the optimal α for (a) 2 000 zeolite structures and (b) 5 484 zeolite environments from the Deem SLC PCOD database. The points in the projections are coloured by the molar energy and volume according to the 2D colour map below the projection. For clarity, energies are centred relative to the mean energy per Si over all 10 000 structures from reference [51]. The frameworks and environments with the highest and lowest volumes and energies are also pictured below their respective projections. Below each KPCovR projection is an analogous KPCA to facilitate a qualitative comparison with the KPCovR projections.

KPCovR loss leads to a map that more closely coincides with the conventional understanding of stable structures as those that have low distortion relative to the ideal carbon bonding geometries.

4.2. Zeolites

We apply KPCovR to a subset of the Deem data set of hypothetical silica zeolites [51], where the use of KPCA to construct an atlas of the building blocks of several thousand zeolites was previously demonstrated by some of the authors [52]. By construction all frameworks in the dataset are based on tetrahedrally coordinated SiO_4 units, yet they differ considerably in terms of molar energies and volumes. A KRR model based on an additive combination of environment Gaussian kernels, built using atom-centred SOAP features with a cutoff of 6.0 Å, achieves an excellent accuracy in predicting the lattice energy (with an error around 0.79 kJ mol^{-1} Si), and molar volume (with an error around 1.88 Å^3/Si atom), with $\ell_{\mathrm{regr}} = 0.0599$ for the zeolite frameworks. However, the first two KPCA components correlate rather weakly with these properties. A data representation based on those components provides information on the structural diversity, but describes only qualitatively structure-property relations. As shown in table 1 KPCovR at the optimal α provides a much more effective description; the latent space covers 59% of the structural diversity, while providing enough information to predict accurately lattice energy (1.31 kJ mol^{-1} Si) and molar volume (2.50 Å^3/Si atom), with $\ell_{\mathrm{regr}} = 0.11$. The map naturally orders structures between regular frameworks that have intermediate densities and low lattice energy (figure 7(a, ii)), toward the bottom, to frameworks with very large pores that have very low density and usually intermediate to high lattice energies. While no clear clusters emerge (which is not unexpected given the origin of the dataset as a high-throughput, random search) one can often observe that nearby structures exhibit similar structural motifs. For instance, most of the structures on the top right side of the map are associated with large 1D channels, as shown in figure 7(a, iii).

For a system exhibiting a combinatorial number of metastable structures, such as silica frameworks, an analysis based on the structural building blocks is often more insightful than the analysis of the overall structures. When using atom-centred features, or additive kernels built on them, it is natural to regard additive properties such as volume or energy as arising from a sum of environmental contributions, and to use these atom-centred environments as the building blocks to rationalise structure-property relations. As discussed in appendix C, the construction of a regression model for the framework properties yields as a side-effect a data-driven partitioning that can be used for a (K)PCovR analysis of such building blocks. The resulting representation (figure 7(b)) shows an excellent correlation between the position in a 2D representation and the predicted contributions to lattice energy and molar volume (site energy RMSE: 1.33 kJ/mol Si, site volume RMSE: 2.33 Å^3/Si atom, 34% of structural variance). As observed in reference [51], and consistent with what is seen in the framework analysis, the thorough search of potential frameworks that underlies the construction of this dataset is reflected in the lack of substantial clustering of the environments. The bulk of the latent space is uniformly covered, and one does not see an obvious qualitative relation between properties and the local topology of the framework. Regular structures are mapped side-by-side to disordered environments. Only at the extremes can one recognise clearer patterns. (A few extremal structures and their location in the KPCovR projection are highlighted in figure 7(b)). High-energy (and hence poorly stable) building blocks can be distinguished between high-density structures that contain highly strained three-member rings, and low-density structures, associated with the surfaces of large cavitites, and to ‘pillar-like’ motifs that are present in the most highly porous frameworks (figure 7(b, vii)). Low-energy structures, in the lower side of the map, tend to have low and intermediate volume, and are predominantly associated with six- and four-member rings.

These are however weak correlations, and in general there are no apparent patterns that correlate the framework topology and the position on the map. This confirms, in a more direct manner, the structure-property insights that were inferred by *separate* application of supervised and unsupervised algorithms in reference [52]—namely that, for four-coordinated silica frameworks, the topology of the network is a good predictor of energy and density only for extremes. For the bulk of the possible binding motifs, a low-dimensional representation is not sufficient to capture the extreme structural diversity and to rationalise the multitude of alternative building blocks that give rise to similar macroscopic materials properties.

4.3. QM9

Our next case study regards the QM9 dataset, which contains over 133 000 molecules consisting of carbon, hydrogen, nitrogen, oxygen, and fluorine [53, 54], of which we use 10 000 for this study. To demonstrate the application of KPCovR to multi-target learning, we construct our models using all 12 properties available in the dataset: internal energy at 0 K and 298.15 K, free energy at 298.15 K, enthalpy at 298.15 K, highest occupied molecular orbital (HOMO), lowest unoccupied molecular orbital (LUMO), HOMO-LUMO gap, heat capacity, zero point vibrational energy (ZPVE), dipole moment, isotropic polarisability, and electronic spatial extent (ESE).

Here, there is not a large qualitative difference between the two-dimensional KPCovR projection at intermediate α and that constructed via KPCA, with the former retaining 63% of the variance compared to the latter’s 66%. The energetic properties, which are well-represented by these first two principal components, are also well-correlated with the degree of unsaturation, and thus the structural diversity of the dataset. This is summarised in figure 8(a), where the projections for $\alpha = 0.5$ and $\alpha = 1.0$ are shown, coloured by the degree of unsaturation[3] and with representative molecules highlighted. Besides the left-to-right saturated-to-unsaturated trend, the map position also correlates with the presence of O, N, F atoms, that increase from bottom to top.

Due to the large number of properties used as targets, a low-dimensional latent space cannot achieve the same prediction accuracy as (kernel) ridge regression, for $\alpha = 0$ and by extension at intermediate values of α. It is necessary to retain a larger number of latent space components to obtain a model capable of effective regression, as seen in table 1, where ℓ_{regr} goes from 0.31 with $n_{\text{latent}} = 2$ to 0.07 with $n_{\text{latent}} = 12$, and seen in figure 8(b). For a given value of α, both regression and projection errors are bound to decrease when retaining a larger number of PCs. The optimal value of α, however, is not necessarily the same for increasing numbers of PCs particularly in datasets where ℓ_{regr} and ℓ_{proj} have a magnitude that varies with n_{latent} in a different way. In this case, however, the optimal α is nearly constant, as shown in figure 8(b). The figure also shows a sudden drop in ℓ_{proj} for $\alpha > 0$. This discontinuity in the variance suggests that insufficient information is contained in the 12 properties to construct an orthogonal set of 12 principal components, and thus some properties must be highly correlated.

Models constructed with fewer principal components can provide insight into the nature of the properties included, particularly in the cases weighted towards regression as $\alpha \to 0$. In figure 8(c), we show the regression errors of the individual properties as a function of n_{latent}. For each property, the decay of ℓ_{regr} when incorporating a new principal component indicates how strongly the new feature and the property are correlated, and gives indirect information on the correlation between properties. For instance, we can see

[3] The degree of unsaturation, defined as $d = C - \frac{H}{2} - \frac{X}{2} + \frac{N}{2} + 1$, where X is a halogen, estimates the number of rings and π bonds in the molecule.

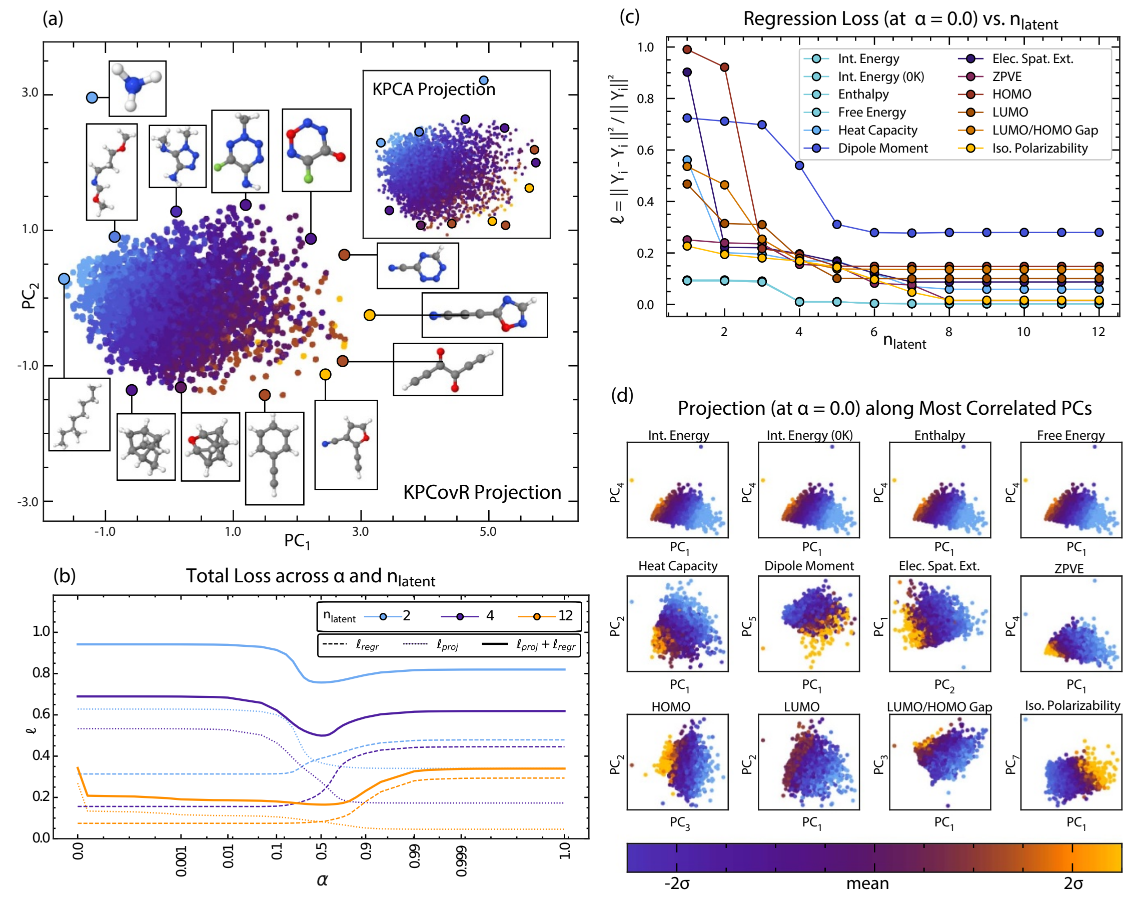

Figure 8. KPCovR for the QM9 Dataset. (a) Comparison of KPCovR ($\alpha = 0.5$) and KPCA (inset) projections, plotted along the first two principal components. Several molecules have been highlighted in insets. Colour denotes the correlated properties of internal energy (at 0 and 285 K), enthalpy, and free energy in corresponding to the colourbar in (d). (b) Total loss across α and the number of principal components for 2 (blue), 4 (purple), and 12 (orange) principal components. Solid lines denote total loss ℓ, dashed ℓ_{regr} and dotted ℓ_{proj}. (c) Per-property regression loss at $\alpha = 0$ for the 12 properties included. Note that the internal energies, enthalpy, and free energy overlap completely and are given by a single colour. (d) Projections at $\alpha = 0.0$ of the 12 properties along their most correlated principal components, where each projection is coloured by the corresponding property, scaled in accordance with the colourbar below.

that (unsurprisingly) the internal energies, enthalpy and free energy are heavily correlated with each other, as their associated ℓ_{regr} decreases precisely in the same way, indicative of a strong correlation with the first and fourth principal components. Figure 8(d) shows colour-coded maps of the 12 targets, using for each of them the two PCs that lead to the largest decrease in ℓ_{regr}.

4.4. Arginine dipeptide

We also applied KPCovR to the 4 217 arginine dipeptide conformers that are collected in the Berlin amino acid database [56], that was also investigated using a purely unsupervised dimensionality-reduction scheme [57]. The conformer energy was used as the target property to construct the KPCovR model. Figure 9 shows the two-component KPCovR projection of the conformers in the test set at the optimal value of $\alpha = 0.55$ coloured by energy (a), radius of gyration (b), and peptide bond isomerism (c). Several individual conformers are also highlighted, including those with the highest and lowest energy and radius of gyration. For comparison, a KPCA projection of the same conformers is plotted in the inset in the lower right corner of each subplot. The KPCA projection alone represents well the different structural features (peptide bond isomerism and radius of gyration), but leads to rather poor energy regression ($\ell_{\mathrm{regr}} = 0.61$ as opposed to $\ell_{\mathrm{regr}} = 0.005$ for KPCovR at optimal α). The KPCovR projection separates more clearly a group of high-energy conformers, to the left, and a cluster of very stable configurations, to the upper right. The former are characterized by having both peptide bonds in the *cis* configuration, and by an unfavourable steric interaction between the terminating methyl groups. The stable conformers, on the other hand, all have the naturally-preferred all-*trans* isomerism, and the backbone takes an extended β-strand structure. They only differ by the hydrogen-bonding pattern of the side-chain, that modulates more subtly the conformational stability.

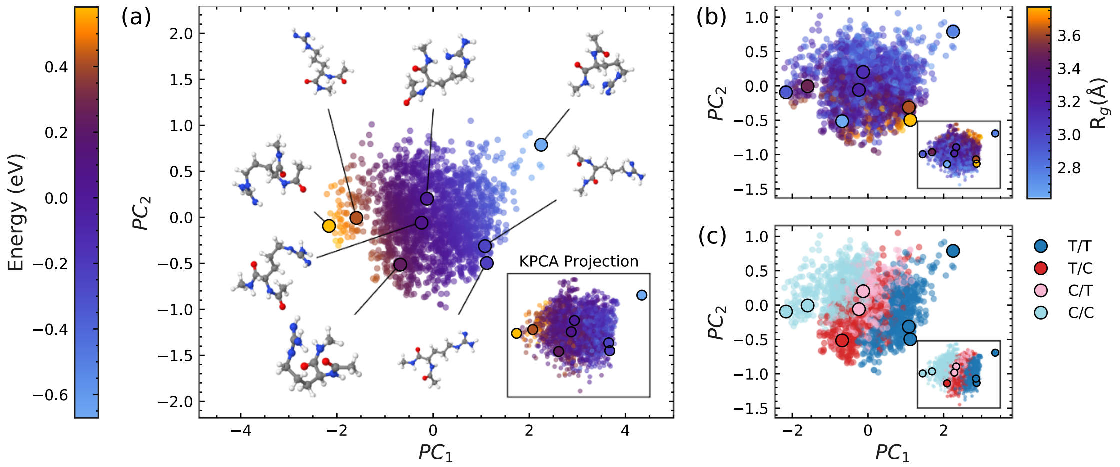

Figure 9. KPCovR of Arginine Dipeptides. Two-dimensional KPCovR projections at the optimal α for bare arginine dipeptides coloured by the (a) structure energies (centred relative to the mean energy over all 4 217 structures), (b) radii of gyration (R_g), and (c) peptide bond isomerism. The peptide bond isomerism is determined by the dihedral angle ω of the two peptide bonds. A particular peptide bond is classified as *cis* (C) if $-\pi/2 < \omega < \pi/2$, and the bond is classified as *trans* (T) if $-\pi < \omega < -\pi/2$ or $\pi/2 < \omega < \pi$. For each KPCovR projection, the inset contains the corresponding KPCA projection. In each KPCovR and KPCA projection, several points are highlighted, and their corresponding atomic snapshots are shown in (a).

The inclusion of an explicit supervised learning component in KPCovR does not only lead to a dimensionality reduction that preserves with much higher accuracy the underlying structure-property relations, but reveals more clearly the molecular motifs that stabilize (or de-stabilize) the different conformers. This kind of analysis is likely to provide valuable insights when investigating the stability of secondary-structure motifs in proteins [58] or the modulation of the stability of conformational space of oligopeptides approaching an inorganic interface [59].

4.5. Azaphenacenes

As a last example, we consider a dataset containing different crystallyne polymorphs of 28 isomers of a pyrrole-based azaphenacene compound, for which total energy and electronic mobility have been previously computed in reference [60]. This dataset present a series of challenges for KPCovR in particular and machine learning techniques in general, and as such is a good test for the new method presented in this paper. First, the dataset contains a very small number of structures (311), half of which were used for training the different models. Second, electronic mobility is an inherently non-local property, which makes it hard to predict it using local descriptors such as SOAP, even when the descriptors are grouped in a *structure* kernel as discussed in appendix C. It provides, therefore, a demonstration of the robustness of KPCovR in presence of target properties that are noisy, or otherwise un-learnable.

We present the results of KPCovR on the validation set for different α values in figure 10, panels (a) and (b). On the pure regression, $\alpha = 0$ side, the prediction of energies is surprisingly good given the very low number of training points. The RMSE of 3.83 kJ mol^{-1} (equivalent to $\ell_{\text{regr}} = 0.038$) is around a quarter of the dataset intrinsic standard deviation of 14.3 kJ mol^{-1}. The prediction of electronic mobility is much harder. We used the logarithm of electronic mobilities instead of the raw values as the prediction target, as detailed in the supplementary information (https://stacks.iop.org/MLST/00/0000001/045021/mmedia). Although this transformation improved our ability to learn electronic mobilities, the regression loss is very high (RMSE of 0.9; $\ell_{\text{regr}} = 0.482$, which is more than 90% of the expected variance of 0.5). We are overall unable to learn electronic mobility for this dataset, similarly to what was already observed using a pure KRR model [60, 61].

Looking now at the optimal $\alpha = 0.65$ (i.e. the value of α minimising the sum of the projection and regression losses), we observe that we are still unable to learn electronic mobility with a relative loss of 0.485 (equivalent to an error that is approximately 95% of the intrinsic variability of the data, and only marginally worse of the error for $\alpha = 0$), and the resulting prediction is visibly skewed in the parity plot. The poor regression performance for the log-mobility is reflected in the lack of a clear correlation between the position in latent space and this target property .

Even if we are unable to predict the electronic mobility, the cohesive energy of the different polymorphs can be learned very effectively, and the optimal $\alpha = 0.65$ corresponds to an excellent balance between ℓ_{regr} and ℓ_{proj}. The latent-space projection separates the data set in two clusters along the vertical axis. This separation is related to the a bimodal energy distribution, with low and high energy structures, which is lost in the limit of pure KPCA at $\alpha = 1$. In figure 10, we show an annotated map of the different crystal stackings,

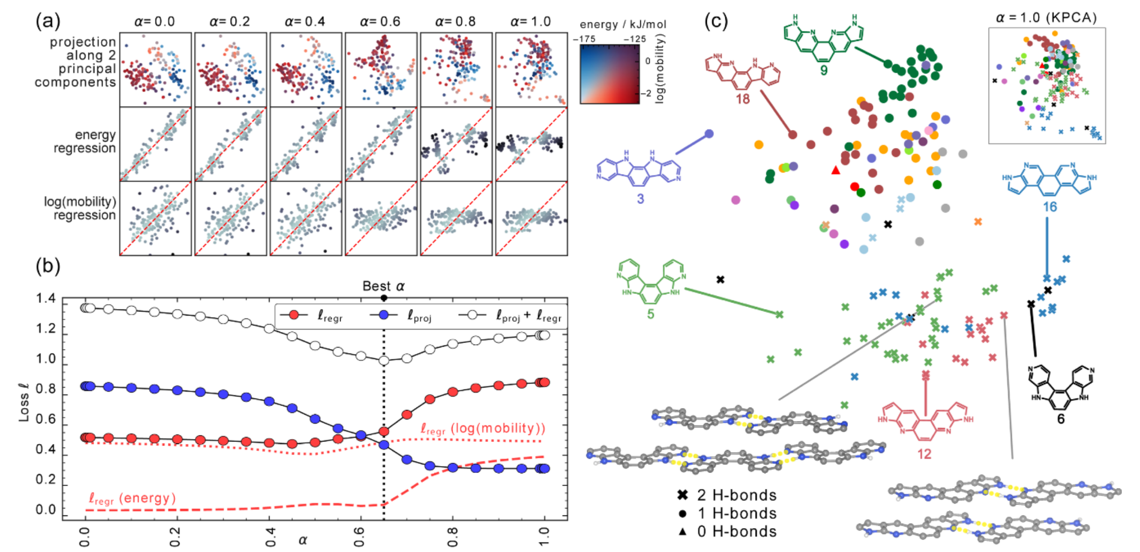

Figure 10. KPCovR of Azaphenacenes. (a) Overview of the projections produced by KPCovR for different α values coloured by both energy and electronic mobility according to the colour map on the side; parity plots for the regression of both energy and log of the electronic mobility over the testing set of Azaphenacenes (b) Evolution of the different losses as a function of alpha ℓ_{regr} is the total regression loss and KPCovR projection loss ℓ_{proj} (c) Annotated map of azaphenacenes as produced by KPCovR at $\alpha = 0.65$. Each point corresponds to a different molecular crystal, with the colours and coloured insets denoting the substituent molecule. Symbols show the number of hydrogen bond per molecule, as computed with PAMM [54]. On the bottom, we show two examples of DNA-style hydrogen bonds between molecules, the images were created using VESTA [55]. The inset in the upper right contains the analogous KPCA projection.

coloured by molecular identity. Different symbols indicate the average number of hydrogen bonds between molecules in the crystals, which we identified using a Probabilistic Analysis of Molecular Motifs (PAMM) [54]. We find that the cluster of low energy stacking contains only structures with two hydrogen bonds per molecule (the maximal possible value), while the high energy cluster contains mostly structures with one hydrogen bond per molecule. Additionally, the majority of structures in the low energy cluster are linked by hydrogen bonds in a DNA-like fashion, i.e. by having pairs of matching hydrogen bond donors and acceptors facing each another. Finally, the low energy cluster only contains crystal created from molecules 5, 6, 12 and 16 (following the notation from the original paper [60]), the 24 other molecules being in the high energy cluster. These four molecules are the only ones with just the right geometry to create matching, DNA-like hydrogen bonding patterns with two bonds per molecules, and high symmetry crystals. Isomer 9 and 18 also contain a similar N-C-NH motif, but cannot form a paired hydrogen bond pattern because of steric hindrance (9) and asymmetry (18).

5. Conclusions

In this paper we provide a comprehensive overview of linear- and kernel-based methods for supervised and unsupervised learning, showing an example of their application to elucidate and predict structure-property relations in solid-state NMR. We also discuss a simple combination of principal component analysis and linear regression, PCovR [18], that has as yet received far less attention than in our opinion it deserves. We derive extensions to PCovR that make it possible to use it in the context of kernel methods (KPCovR and sparse KPCovR), and demonstrate their application to five distinct example datasets of molecules and materials. We also prepared a set of Jupyter notebooks [27] that provide a pedagogic introduction to both traditional and novel methods we discuss, and allow exporting structure-property maps in a format that can be visualised with an interactive tool that we also developed as part of this work [48].

The flexibility afforded by a kernel method allows improving substantially (typically by a factor of two) the regression performance relative to linear PCovR. Compared to kernel PCA, KPCovR maps reflect more explicitly structure-property relations, and—in all the diverse cases we considered—are more revealing, helping to identify the molecular motifs that determine the behaviour of the different structures, and that often reflect intuitive chemical concepts such as hybridisation, chemical composition, and H-bond patterns. This study highlights the promise of combining supervised and unsupervised schemes in the analysis of data generated by atomistic modelling, to obtain clearer insights to guide the design of molecules and materials with improved performance, and to build more effective models to directly predict atomic-scale properties.

Acknowledgment

MC, RKC, BAH acknowledge funding by the from the European Research Council (Horizon 2020 grant agreement no. 677013-HBMAP). GF acknowledge support by the SCCER Efficiency of Industrial Processes, and by the European Center of Excellence MaX, Materials at the Hexascale - GA No. 676598.

Supporting information

The electronic supporting information contains a comprehensive discussions of the parameters used for the analysis of each of the five examples, together with a comprehensive comparison of the performance of different linear, kernel and PCovR-like methods for the five datasets. For each dataset we also provide an interactive map that can be visualized with the online viewer chemiscope [48], archived at reference [47]. A set of Juypyter notebooks that provide a hands-on tutorial for the application of KPCovR is available in a separate repository [27].

Appendix A. Centring and scaling

In this paper, as it is often done in machine learning applications, we centre and scale (or standardise) the original input data, which removes the dependency of results on a trivial shifting or scaling of the data set. Standardisation is of particular importance in PCovR-based methods, as the model can be inherently biased towards the projection or regression if $\mathbf{X}$ and $\mathbf{Y}$ data are of different relative magnitudes. To avoid ambiguity, we centre and scale our raw data $\mathbf{X}'$ and $\mathbf{Y}'$ in the following manner,

$$\mathbf{X} = \frac{\sqrt{n_{\text{train}}}\left(\mathbf{X}' - \bar{\mathbf{X}}'_{\text{train}}\right)}{\|\mathbf{X}'_{\text{train}} - \bar{\mathbf{X}}'_{\text{train}}\|} \tag{A1}$$

$$\mathbf{Y}_i = \frac{\sqrt{n_{\text{train}}}\left(\mathbf{Y}'_i - \bar{\mathbf{Y}}'_{i,\text{train}}\right)}{\sqrt{n_{\text{properties}}}\|\mathbf{Y}'_{i,\text{train}} - \bar{\mathbf{Y}}'_{i,\text{train}}\|}, \tag{A2}$$

where $\mathbf{A}_i$ denotes the i^{th} property (column) of $\mathbf{A}$, $\bar{\mathbf{A}}$ is the columnwise mean of $\mathbf{A}$, and $\mathbf{A}_{\text{train}}$ indicates the subset of samples in $\mathbf{A}$ that belong to the training set. By standardising the data in this manner we ensure that the squared Frobenius norms of $\mathbf{X}_{\text{train}}$ and $\mathbf{Y}_{\text{train}}$ are equal to n_{samples}, and that individual property variances of $\mathbf{Y}_{\text{train}}$ are all equal to $1/n_{\text{properties}}$.

We perform a similar centring and scaling procedure when constructing kernels. Kernel standardization can be viewed as simply centring and scaling the data in the RKHS feature space. If N indicates the dataset that defines the centring (typically the train set) and i, j two data points between which we want to compute the centred kernel,

$$K_{ij} = \frac{n_N\left(\phi_i - \bar{\mathbf{\Phi}}_N\right)^T\left(\phi_j - \bar{\mathbf{\Phi}}_N\right)}{\mathrm{Tr}\left(\left(\mathbf{\Phi}_N - \bar{\mathbf{\Phi}}_N\right)\left(\mathbf{\Phi}_N - \bar{\mathbf{\Phi}}_N\right)^T\right)} \tag{A3}$$

where $\bar{\mathbf{\Phi}}_N$ is the column mean of the training set feature matrix $\mathbf{\Phi}_N$, and is computed once and for all for the train set, together with the normalisation factor. This can be written avoiding to compute explicitly the RKHS features:

$$K_{ij} = \frac{n_{\text{train}}}{\mathrm{Tr}(\mathbf{K}_{NN})}\left(K'_{ij} - \sum_{n\in N}\frac{K'_{in} + K'_{jn}}{n_{\text{train}}} + \sum_{nn'\in N}\frac{K'_{nn'}}{n^2_{\text{train}}}\right). \tag{A4}$$

Centring is achieved by computing column averages of the raw kernels between points i, j and the train set points. Note that kernel matrix elements may refer to different matrices, depending on whether i and j are themselves train set points, or new inputs. The scaling factor $n_{\text{train}}/\mathrm{Tr}(\mathbf{K}_{NN})$ is computed using the *centred* train set kernel.

In sparse KPCA and sparse KPCovR, a slightly different approach is required, as the goal is to ensure that the Nyström approximation to the full kernel matrix is centred and scaled properly—i.e. the active set kernel defines the RKHS, but centring and scaling should be computed based on the training set N. A centred and scaled kernel between an input i and an active point $m \in M$ can then be computed as

$$K_{im} = \sqrt{\frac{n_{\text{train}}}{\mathrm{Tr}\left(\mathbf{K}_{NM}\mathbf{K}_{MM}^{-1}\mathbf{K}_{NM}^T\right)}}\left(K'_{im} - \sum_{j\in N}\frac{K'_{jm}}{n_{\text{train}}}\right), \tag{A5}$$

where once more the trace included at the denominator in the normalisation factor is computed using the centred version of $\mathbf{K}_{NM}$.

Appendix B. Projection loss in kernel methods and KPCovR

Rewriting equation (11) in terms of the RKHS, we get:

$$\ell = \alpha\|\mathbf{\Phi} - \mathbf{T}\mathbf{P}_{T\Phi}\|^2/n_{\text{samples}} + (1-\alpha)\|\mathbf{Y} - \mathbf{T}\mathbf{P}_{TY}\|^2/n_{\text{samples}}. \tag{B6}$$

The latter portion of this equation, ℓ_{regr}, can be written in terms of the KRR loss given in equation (22), where $\mathbf{P}_{\Phi Y}$ also encapsulates the loss incurred from the latent space projection. The former portion, ℓ_{proj}, is straightforward to compute given an explicit RKHS. In case one wants to avoid evaluating the RKHS, however, ℓ_{proj} may be computed in terms of the kernel.

Indicating the kernel between set A and B as $\mathbf{K}_{AB}$, the projection of set A as $\mathbf{T}_A$, and with N and V the train and validation/test set, one obtains

$$\begin{aligned}\ell_{\text{proj}} =& \operatorname{Tr}\left[\mathbf{K}_{VV} - 2\mathbf{K}_{VN}\mathbf{T}_N(\mathbf{T}_N^T\mathbf{T}_N)^{-1}\mathbf{T}_V^T\right.\\ &\left.+\mathbf{T}_V(\mathbf{T}_N^T\mathbf{T}_N)^{-1}\mathbf{T}_N^T\mathbf{K}_{NN}\,\mathbf{T}_N(\mathbf{T}_N^T\mathbf{T}_N)^{-1}\mathbf{T}_V^T.\right]\end{aligned} \tag{B7}$$

When the loss is evaluated on the train set, so that $N \equiv V$, this expression reduces to

$$\ell_{\text{proj}} = \operatorname{Tr}\left(\mathbf{K}_{NN} - \mathbf{K}_{NN}\mathbf{P}_{KT}\mathbf{P}_{TK}\right). \tag{B8}$$

where $\mathbf{P}_{TK} = (\mathbf{T}_N^T\mathbf{T}_N)^{-1}\mathbf{T}_N^T\mathbf{K}_{NN}$. A full derivation of this loss equation can be found in the SI.

Appendix C. Structures and environments

When analyzing molecular or materials structures, there are several possible scenarios, involving the prediction of atom-centred or global properties, and the search for structural correlations between atomic environments or overall structures. Whenever the nature of the property and that of the structural entity match, the formalism we have reviewed in section 2 applies straightforwardly. A common scenario that deserves a separate discussion involves the case in which one seeks to reveal how atomic environments or molecular fragments contribute to global properties of a material. Often, this means that the properties of a structure $\mathbf{y}(\mathcal{A})$ are written as a sum over contributions from the atom-centred environments in each structure, $\mathbf{y}(\mathcal{A}) = \sum_{i\in\mathcal{A}}\mathbf{y}(\mathcal{X}_i)$. For a linear model, this means that the regression loss reads

$$\begin{aligned}\ell_{\text{regr}} =& \frac{1}{n_{\text{samples}}}\sum_{\mathcal{A}}\|\mathbf{y}(\mathcal{A}) - \sum_{i\in\mathcal{A}}\mathbf{X}_i\mathbf{P}_{XY}\|^2\\ =& \frac{1}{n_{\text{samples}}}\sum_{\mathcal{A}}\|\mathbf{y}(\mathcal{A}) - \tilde{\mathbf{X}}(\mathcal{A})\mathbf{P}_{XY}\|^2,\end{aligned} \tag{C9}$$

where we defined $\tilde{\mathbf{X}}(\mathcal{A}) = \sum_{i\in\mathcal{A}}\mathbf{X}_i$. In other terms, one can formulate the regression using features that describe the structures as a sum of the features of their atoms, and then proceed to determine the weights $\mathbf{P}_{XY}$ as in conventional linear regression. A similar expression holds for kernel methods, where the kernels between structures can be built as sums over kernels between environments, resulting in an additive property model. In a (K)PCovR framework, where one is restricted to learning the fraction of the properties that can be approximated as a (kernelized) linear function of $\mathbf{X}$, one should first train a model based on the full structures, and then compute the predictions for individual environments. These are combined to form the approximate property matrix $\hat{\mathbf{Y}}$. The model can then be built as in the homogeneous case of environment features and atom-centred properties.

ORCID iDs

Benjamin A Helfrecht https://orcid.org/0000-0002-2260-7183
Rose K Cersonsky https://orcid.org/0000-0003-4515-3441
Guillaume Fraux https://orcid.org/0000-0003-4824-6512
Michele Ceriotti https://orcid.org/0000-0003-2571-2832

References

[1] Faber F A, Lindmaa A, von Lilienfeld O A and Armiento R 2016 *Phys. Rev. Lett.* **117** 135502

[2] Faber F A *et al* 2017 *J. Chemical Theory Computat.* **13** 5255

[3] Hansen K, Montavon G, Biegler F, Fazli S, Rupp M, Scheffler M, von Lilienfeld O A, Tkatchenko A and Müller K-R 2013 *J. Chemical Theory Computat.* **9** 3404

[4] Rupp M, Tkatchenko A, Müller K-R and von Lilienfeld O A 2012 *Phys. Rev. Lett.* **108** 058301

[5] Deringer V L and Csányi G 2017 *Phys. Rev.* B **95** 094203

[6] Dragoni D, Daff T D, Csányi G and Marzari N 2018 *Phys. Rev. Mater.* **2** 013808

[7] Maillet J-B, Denoual C and Csányi G 2018 *Conf. Proc.* **1979** 050011

[8] Szlachta W J, Bartók A P and Csányi G 2014 *Phys. Rev.* B **90** 104108

[9] Simon C M *et al* 2015 *Energy Environ. Sci.* **8** 1190

[10] Sendek A D, Yang Q, Cubuk E D, Duerloo K-A N, Cui Y and Reed E J 2017 *Energy Environ. Sci.* **10** 306

[11] Kahle L, Marcolongo A and Marzari N 2020 *Energy Environ. Sci.* **13** 928–48

[12] Kirklin S, Meredig B and Wolverton C 2013 *Adv. Energy Mater.* **3** 252

[13] Ceriotti M 2019 *J. Chem. Phys.* **150** 150901

[14] Jolliffe I T 1982 *Phil. Trans. R. Soc.* C **31** 300

[15] Wold S, Sjöström M and Eriksson L 2001 *Chemometr. Intell. Lab. Syst* **58** 109

[16] Späth H 1979 *Computing* **22** 367

[17] Stone M and Brooks R J 1990 *Phil. Trans. R. Soc.* B **52** 237

[18] de Jong S and Kiers H A L 1992 *Chemometr. Intell. Lab. Syst* 14 155

[19] Vervloet M, Van Deun K, Van den Noortgate W and Ceulemans E 2013 *Chemometr. Intell. Lab. Syst.* **123** 36

[20] Vervloet M, Kiers H A L, Van den Noortgate W and Ceulemans E 2015 *J. Stat. Software* **65** 1

[21] Vervloet M, Van Deun K, Van den Noortgate W and Ceulemans E 2016 *Chemometr. Intell. Lab. Syst.* **151** 26

[22] Fischer M J 2014 *J. Geophys. Res. Atmos.* **119** 1266

[23] Heij C, Groenen P J and van Dijk D 2007 *Computat. Stat. Data Anal.* **51** 3612

[24] Van Deun K, Crompvoets E A V and Ceulemans E 2018 *BMC Bioinform.* **19** 104

[25] Taylor M K, Sullivan D K, Ellerbeck E F, Gajewski B J and Gibbs H D 2019 *Public Health Nutrition* **22** 2157–69

[26] Wilderjans T F, Vande Gaer E, Kiers H A L, Van Mechelen I and Ceulemans E 2017 *Psychometrika* **82** 86

[27] A set of utilities and pedagogic notebooks for the use of linear and kernel methods in atomistic modeling (https://github.com/cosmo-epfl/kernel-tutorials/)

[28] Ceriotti M, Emsley L, Paruzzo F, Hofstetter A, Musil F, De S, Engel E A and Anelli A 2019 *Chemical Shifts in Molecular Solids by Machine Learning Datasets Materials Cloud Archive* **2019.0023/v1**

[29] Bartók A P, Kondor R and Csányi G 2013 *Phys. Rev.* B **87** 184115

[30] Willatt M J, Musil F and Ceriotti M 2019 *J. Chem. Phys.* **150** 154110

[31] Schölkopf B, Smola A and Müller K 1998 *Neural Comput.* **10** 1299

[32] Hastie T, Tibshirani R and Friedman J 2008 *Elements of Statistical Learning: Data Mining, Inference and Prediction* 2nd edn (Berlin: Springer Series in Statistics)

[33] Bishop C M 2006 *Pattern Recognition and Machine Learning* Information Science and Statistics (Berlin: Springer)

[34] K P F R S 1901 *London Edinburgh Dublin Phil. Mag. J. Sci.* 2 559

[35] Hotelling H 1933 *J. Educational Psychol.* **24** 417

[36] Torgerson W S 1952 *Psychometrika* **17** 401

[37] Cuturi M 2009 ArXiv: 0911.5367

[38] Mercer J and Forsyth A R 1909 *Phil. Trans. R. Soc.* A **209** 415

[39] Girosi F, Jones M and Poggio T 1995 *Neural Comput.* **7** 219

[40] Smola A J and Schökopf B 2000 *Proc. of the Seventeenth Int. Conf. on Machine Learning* (San Fransisco, CA: Morgan Kaufmann Publishers) 911–18

[41] Murphy K P 2012 *Machine Learning: A Probabilistic Perspective* (Cambridge, MA: MIT Press)

[42] Williams C K I and Seeger M 2001 *Advances in Neural Information Processing Systems* 13 editor ed Leen T K, Dietterich T G and Tresp V (Cambridge, MA: MIT Press) pp 682–8 http://papers.nips.cc/paper/1866-using-the-nystrom-method-to-speed-up-kernel-machines.pdf

[43] Eldar Y, Lindenbaum M, Porat M and Zeevi Y 1997 *IEEE Trans. Image Process.* **6** 1305

[44] Mahoney M W and Drineas P 2009 *Proc. Natl Acad. Sci.* **106** 697

[45] Bartók A P and Csányi G 2015 *Int. J. Quantum Chem.* **115** 1051

[46] Imbalzano G, Anelli A, Giofré D, Klees S, Behler J and Ceriotti M 2018 *J. Chem. Phys.* **148** 241730

[47] Helfrecht B A, Cersonsky R K, Fraux G and Ceriotti M 2020 *Structure-Property Maps With Kernel Principal Covariates Regression* (Materials Cloud Archive) 2020.80

[48] Fraux G, Cersonsky R K and Ceriotti M 2020 *J. Open Source Software* **5** 2117

[49] Pickard C J and Needs R J 2011 *J. Phys. Condens. Matter* **23** 053201

[50] Pickard C J 2020 *AIRSS Data for Carbon at 10GPa and the C+N+H+O System at 1GPa*

[51] Pophale R, Cheeseman P A and Deem M W 2011 *Phys. Chem. Chem. Phys.* **13** 12407. The SLC-PCOD database can be accessed at: www.hypotheticalzeolites.net/DATABASE/DEEM/DEEM_PCOD/index.php.

[52] Helfrecht B A, Semino R, Pireddu G, Auerbach S M and Ceriotti M 2019 *J. Chem. Phys.* **151** 154112

[53] Ramakrishnan R, Dral P O, Rupp M and von Lilienfeld O A 2014 *Sci. Data* **1** 140022

[54] Gasparotto P and Ceriotti M 2014 *J. Chem. Phys.* **141** 174110

[55] Momma K and Izumi F 2011 *J. Appl. Crystallogr.* **44** 1272

[56] Ropo M, Schneider M, Baldauf C and Blum V 2016 *Scientific Data* **3** 1. The Berlin amino acid database can be accessed at: https://aminoaciddb.rz-berlin.mpg.de/

[57] De S, Musil F, Ingram T, Baldauf C and Ceriotti M 2017 *J. Cheminformatics* **9** 1

[58] Helfrecht B A, Gasparotto P, Giberti F and Ceriotti M 2019 *Front. Mol. Biosci.* **6** 1

[59] Maksimov D, Baldauf C and Rossi M 2020 *Int. J. Quantum Chem.* e26369

[60] Yang J, De S, Campbell J E, Li S, Ceriotti M and Day G M 2018 *Chem. Mater.* **30** 4361

[61] Musil F, De S, Yang J, Campbell J E, Day G M and Ceriotti M 2018 *Chem. Sci.* **9** 1289