Data Mining and Knowledge Discovery
https://doi.org/10.1007/s10618-024-01015-0

Interpretable linear dimensionality reduction based on bias-variance analysis

Paolo Bonetti[1] · **Alberto Maria Metelli**[1] · **Marcello Restelli**[1]

Received: 10 January 2023 / Accepted: 21 February 2024

Abstract
One of the central issues of several machine learning applications on real data is the choice of the input features. Ideally, the designer should select a small number of the relevant, nonredundant features to preserve the complete information contained in the original dataset, with little collinearity among features. This procedure helps mitigate problems like overfitting and the curse of dimensionality, which arise when dealing with high-dimensional problems. On the other hand, it is not desirable to simply discard some features, since they may still contain information that can be exploited to improve results. Instead, *dimensionality reduction* techniques are designed to limit the number of features in a dataset by projecting them into a lower dimensional space, possibly considering all the original features. However, the projected features resulting from the application of dimensionality reduction techniques are usually difficult to interpret. In this paper, we seek to design a principled dimensionality reduction approach that maintains the interpretability of the resulting features. Specifically, we propose a bias-variance analysis for linear models and we leverage these theoretical results to design an algorithm, *Linear Correlated Features Aggregation* (LinCFA), which aggregates groups of continuous features with their average if their correlation is "sufficiently large". In this way, all features are considered, the dimensionality is reduced and the interpretability is preserved. Finally, we provide numerical validations of the proposed algorithm both on synthetic datasets to confirm the theoretical results and on real datasets to show some promising applications.

Keywords Dimensionality reduction · Linear regression · Bias-variance tradeoff · Feature aggregation

Responsible editor: Mark Last.

Extended author information available on the last page of the article

Published online: 25 March 2024

1 Introduction

Dimensionality reduction plays a crucial role in applying Machine Learning (ML) techniques in real-world datasets (Sorzano et al. 2014). Indeed, in a large variety of scenarios, data are high-dimensional with a large number of correlated features. For instance, *financial* datasets are characterized by time series representing the trend of stocks in the financial market, and *climatological* datasets include several highly-correlated features that, for example, represent temperature value at different points on the Earth. On the other hand, only a small subset of features is usually significant for learning a specific task, and it should be identified to train a well-performing ML algorithm. In particular, considering many redundant features boosts the model complexity, which increases its variance and the risk of overfitting (Hastie et al. 2009). Furthermore, when the number of features is high, and comparable with the number of samples, the available data become sparse, leading to poor performance (*curse of dimensionality* (Bishop and Nasrabadi 2006)). For this reason, *dimensionality reduction* and *feature selection* techniques are usually applied. Feature selection (Chandrashekar and Sahin 2014) focuses on choosing a subset of features important for learning the target following a specific criterion (e.g., the most correlated with the target, the ones that produce the highest validation score), discarding the others. On the other hand, dimensionality reduction methods (Sorzano et al. 2014) maintain all the features projecting them in a (much) lower dimensional space, producing new features that are linear or non-linear combinations of the original ones. Compared to feature selection, this latter approach has the advantage of reducing the dimensionality without discarding any feature and exploiting all of their contributions to the projections. Moreover, recalling that the variance of a sum of random variables is smaller than or equal to the original one, the features computed with linear dimensionality reduction have smaller variance. However, the reduced features might be less interpretable since they are linear combinations of the original ones with different coefficients.

In this paper, we propose a novel dimensionality reduction method that exploits the information of each feature, without discarding any of them, while preserving the interpretability of the resulting feature set. To this end, we *aggregate* features through their average, and we propose a criterion that aggregates two features when it is beneficial in terms of the bias-variance tradeoff. Specifically, we focus on linear regression, assuming a linear relationship between the features and the target. In this context, the main idea of this work is to identify a group of *aggregable* features and substitute them with their average. Intuitively, in linear settings, two features should be aggregated if their correlation is *large enough*. We identify a theoretical threshold on the minimum correlation for which it is profitable to unify the two features. This threshold is the minimum correlation value between two features for which, comparing the two linear regression models before and after the aggregation, the variance decrease is larger than the increase of bias.

Choosing the average to aggregate the features is to preserve interpretability (the resulting reduced feature is just the average of k original features).

Another advantage is that the variance of the average is smaller than the variance of the original features if they are not perfectly correlated. Indeed, assuming that we unify k standardized features, the variance of their average becomes $var(\bar{X}) = \frac{1}{k} + \frac{k-1}{k}\rho$, with ρ being the average correlation of distinct features (Jacod and Protter 2004). The main restriction of choosing the average to aggregate is that we will only consider continuous features since the mean is not well-defined for categorical features. Moreover, it would be improper to evaluate the mean between heterogeneous features: interpretability is preserved only if the aggregation is meaningful.

Another issue may arise when considering features with a different unit of measurement or scale, for this reason we will consider standardized variables.

Remark 1 (About the linearity assumption and non-linear cases.) The theoretical analysis that lays the foundations of the proposed algorithm is limited to linear assumptions and considers linear regression as ML method. However, the proposed algorithm preserves a relevant significance. Indeed, the theoretical analysis allows to prove that the proposed algorithm is theoretically sound, assuming linearity. Then, the algorithm is designed relying on the linear theoretical result, but it can be applied to any regression problem with continuous features, where the proposed threshold becomes a heuristic quantity. While the theoretical guarantees no longer hold, this claim is supported by the empirical validation of the method on real-world datasets, which have no guarantee of linearity, but show a promising applicability of the proposed method outside linear contexts. Additionally, as usually done also in linear regression, it is possible to consider non-linear transformations of the original features as inputs of the LinCFA algorithm to relax the linearity assumption in some specific contexts.

Remark 2 (About interpretability.) Complex (linear and non-linear) transformations of the original features are usually performed by dimensionality reduction methods. As an example, PCA performs a linear combination of potentially all the original features, each with different weights. This kind of aggregation is already defined in Kovalerchuk et al. (2021) as not completely interpretable, since they define these kind of transformation as *quasi-explainable*. In this context, the LinCFA Algorithm is interpretable, since it only relies on performing the mean of several features, which is a transformation that a domain expert can understand without any additional explanation by ML experts. Lahav et al. (2018) define interpretability as: "the extent to which a ML model can be made understandable to relevant human users, with the goal of increasing users' trust in, and willingness to utilize, the model in practice" and Kovalerchuk et al. (2021) defines interpretability in this terms: "the model is explainable if it is presented only in the domain terms (e.g., medicine) without terms that have no meaning in the domain". The mean of variables known by the domain expert can therefore be considered interpretable in these terms. Additionally, the interpretability of the proposed method is particularly clear when the features have the same unit of measure. In this context, the reduced features are

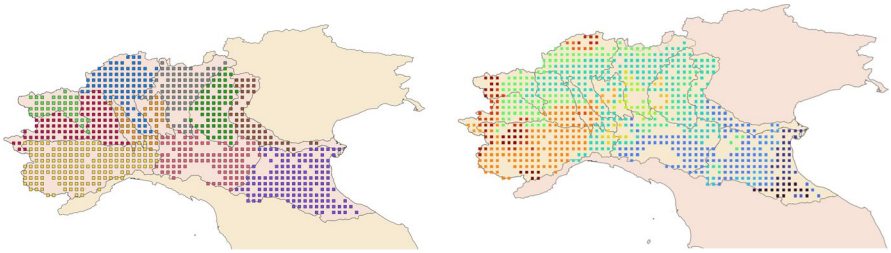

Fig. 1 Figure 1a shows the location of temperature measurements for each of the ten sub-basins of the Po River. Each color identifies the set of locations belonging to the same sub-basin. Figure 1b shows each of the clusters identified by the *LinCFA* algorithm. Each color represents a different set of locations where the algorithm performs an aggregation with the mean (Color figure online)

simply the average of a set of measurements of the same quantity at different locations, with different sensors or at different time frames.

Additionally, depending on the applicative problem considered, the reduction performed with the mean can be particularly meaningful for domain experts. An example with meteorological measurements highlights the main applicative motivation behind the proposed approach. Indeed, a standard preprocessing approach often adopted in ML-based works for Earth science applications consists in computing the mean of a set of neighbouring measurements of the same physical quantity (e.g., temperature measurements in different locations). This method leads to the extraction of features that are average values of quantities over a region. As an example, Fig. 1a shows temperature gridded data related to ten different sub-basins of the Po River. In particular, each colored point in the figure represents one location, where temperature measurements are available. Therefore, each point can be seen as the location of a feature, that represents the temperature in that specific coordinates. To reduce the dimensionality, one may average the measurements of all the points within a sub-basin, following the geographical location of the data (in Fig. 1a each color identifies the set of locations belonging to a specific sub-basin). However, this has no guarantees on the ML performance and does not take into account the relationships between the data. The *LinCFA* algorithm, on the other hand, focuses on the relationships between pairs of points (i.e., features) and their relationship with the target to decide whether to aggregate them. From Fig. 1b it is possible to see the aggregations performed by the *LinCFA* algorithm: the dots having the same color correspond to the locations of the temperature features that the algorithm aggregates with their mean. Therefore, from the figure it is possible to conclude that, in this case, the data-driven approach aggregates the points differently from the geographical boundaries of the sub-basins. This *preserves the interpretability* since it aggregates measurements in different locations in the same way that domain expert does, with the advantage of being a data-driven approach, theoretically motivated.

Outline: The paper is structured as follows. In Sect. 2, we formally define the problem, and we provide a brief overview of the main dimensionality reduction

methods. Section 3 introduces the methodology that will be followed throughout the paper. In Sect. 4, the main theoretical result is presented for the bivariate setting, which is then generalized to D dimensions in Sect. 5. Finally, in Sect. 6, the proposed algorithm, *Linear Correlated Features Aggregation* (LinCFA), is applied to synthetic and real-world datasets to experimentally confirm the result and lead to the conclusions of Sect. 7. The paper is accompanied by supplementary material. Specifically, Appendix A contains the proofs and technical results of the bivariate case that are not reported in the main paper, Appendix B shows an additional finite-samples bivariate analysis, Appendix C elaborates on the bivariate results to be composed only of theoretical or empirical quantities, Appendix D contains the proofs and technical results of the three-dimensional setting, and Appendix E presents in more details the experiments performed.

2 Preliminaries

In this section, we introduce the notation and assumptions employed in the paper (Sect. 2.1) and we survey the main related works (Sect. 2.2).

2.1 Notation and assumptions

Let (X, Y) be random variables with joint probability distribution $P_{X,Y}$, where $X \in \mathbb{R}^D$ is the D-dimensional vector of features and $Y \in \mathbb{R}$ is the scalar target of a supervised learning regression problem. Given N data sampled from the distribution $P_{X,Y}$, we denote the corresponding feature matrix as $\mathbf{X} \in \mathbb{R}^{N \times D}$ and the target vector as $\mathbf{Y} \in \mathbb{R}^N$. Each element of the random vector X is denoted with x_i and it is called a *feature* of the ML problem. We denote as y the scalar target random variable and with σ_y^2 and $\hat{\sigma}_y^2$ its variance and sample variance. For each pair of random variables a, b we denote with σ_a^2, $cov(a, b)$ and $\rho_{a,b}$ respectively the variance of the random variable a and its covariance and correlation with the random variable b. Their estimators are $\hat{\sigma}_a^2$, $\hat{cov}(a, b)$ and $\hat{\rho}_{a,b}$. Finally, the expected value and the variance operators applied on a function $f(a)$ of a random variable a w.r.t. its distribution are denoted with $\mathbb{E}_a[f(a)]$ and $var_a(f(a))$.

A dimensionality reduction method can be seen as a function $\boldsymbol{\phi} : \mathbb{R}^{N \times D} \to \mathbb{R}^{N \times d}$, mapping the original feature matrix $\mathbf{X}$ with dimensionality D into a reduced dataset $\mathbf{U} = \boldsymbol{\phi}(\mathbf{X}) \in \mathbb{R}^{N \times d}$ with $d < D$. The goal of this projection is to reduce the (possibly huge) dimensionality of the original dataset while keeping as much information as possible in the reduced dataset. This is usually done by preserving a distance (e.g., Euclidean, geodesic) or the probability of a point to have the same neighbours after the projection (Zaki and Meira 2014).

In this paper, we assume a *linear* dependence between the features X and the target Y, i.e., $Y = w^T X + \epsilon$, where ϵ is a zero-mean noise, independent of X, and $w \in \mathbb{R}^D$ is the weight vector. Without loss of generality, the expected value of each feature is assumed to be zero, i.e., $\mathbb{E}[x_i] = \mathbb{E}[Y] = 0 \; \forall i \in \{1, \dots, D\}$. Finally, we

consider linear regression as ML method: the i-th estimated coefficient is denoted with $\hat{w}_i$, the estimated noise with $\hat{\epsilon}$ and the predicted (scalar) target with $\hat{y}$.

2.2 Existing methods

This section briefly surveys dimensionality reduction algorithms available in the literature, presenting unsupervised and supervised approaches. More extensive reviews can be found in (Sorzano et al. 2014; Cunningham and Ghahramani 2015; Espadoto et al. 2021; Chao et al. 2019). The algorithm presented in this paper can be considered as a linear supervised dimensionality reduction approach, therefore the focus will be on this topic. However, feature selection also provides a set of reduced features, as discussed in Chapter 1 (the interested reader may refer to literature reviews such as (Li et al. 2017)). Therefore, RReliefF algorithm (Robnik-Sikonja et al. 1997; Kononenko et al. 1997) will also be considered in the empirical evaluation as a supervised feature selection approach.

2.2.1 Unsupervised dimensionality reduction

Classical dimensionality reduction methods can be considered as *unsupervised* learning techniques which, in general, do not take into account the target, but they focus on projecting the dataset $\mathbf{X}$, minimizing a given loss.

The most popular unsupervised linear dimensionality reduction technique is Principal Components Analysis (PCA) (Pearson 1901; Hotelling 1933), a linear method that embeds the data into a linear subspace of dimension d describing as much as possible the variance in the original dataset. One of the main difficulties of applying PCA in real problems is that it performs linear combinations of possibly all the D features, usually with different coefficients, losing the interpretability of each principal component and suffering the curse of dimensionality. To overcome this issue, there exist some variants like svPCA (Ulfarsson and Solo 2011), which forces most of the weights of the projection to be zero. This contrasts with the approach proposed in this paper, which aims to preserve interpretability while exploiting the information yielded by each feature.

There exist several variants to overcome different issues of PCA (e.g., out-of-sample generalization, linearity, sensitivity to outliers) and other methods that approach the problem from a different perspective (e.g., generative approach with Factor Analysis, independence-based approach with Independent Component Analysis, matrix factorization with SVD), an extensive overview can be found in (Sorzano et al. 2014). A broader overview of linear dimensionality reduction techniques can be found in (Cunningham and Ghahramani 2015). Specifically, SVD (Golub and Reinsch 1970) leads to the same result of PCA from an algebraic perspective through matrix decomposition. Factor analysis (Thurstone 1931) assumes that the features are generated from a smaller set of latent variables, called factors, and tries to identify them by looking at the covariance matrix. Both PCA and Factor Analysis can reduce through rotations the number of features that are combined for each reduced component to improve the interpretability, but their coefficients can still be

different and hard to interpret. Finally, Independent Component Analysis (Hyvärinen 1999) is an information theory approach that looks for independent components (not only uncorrelated as PCA) that are not constrained to be orthogonal. This method is more focused on splitting different signals mixed between features than on reducing their dimensionality, which can be done as a subsequent step with feature selection, which would be simplified from the fact that the new features are independent.

Differently from the linear nature of PCA, many non-linear approaches exist (see (Van Der Maaten et al. 2009; Espadoto et al. 2021) for a broader discussion), following the idea that the data can be projected onto non-linear manifolds. Some of them optimize a convex objective function (usually solvable through a generalized eigenproblem) trying to preserve global similarity of data (e.g., Isomap (Tenenbaum et al. 2000), Kernel PCA (Shawe-Taylor and Cristianini 2004), Kernel Entropy Component Analysis (Jenssen 2009), MVU (Weinberger et al. 2004), Diffusion Maps (Lafon and Lee 2006)) or local similarity of data (LLE (Roweis and Saul 2000), Laplacian Eigenmaps (Belkin and Niyogi 2001), LTSA (Zhang and Zha 2004), LPP (He and Niyogi 2003)). Other methods optimize a non-convex objective function with the purpose of rescaling Euclidean distance (Sammon Mapping (Sammon 1969)) introducing more complex structures like neural networks (Multilayer Autoencoders (Hinton and Salakhutdinov 2006)) or aligning mixtures of models (LLC (Teh and Roweis 2002)).

In this paper we assume linearity, therefore in the experimental section we will compare the proposed method with classical PCA and its supervised version, since it is one of the most applied linear unsupervised dimensionality reduction techniques in ML applications. Non-linear techniques for dimensionality reduction (Kernel PCA, Isomap, LLE, LPP) will also be considered to further test the behavior of the *LinCFA* algorithm on real data, where linearity is not guaranteed, together with RReliefF algorithm as nonlinear supervised feature selection approach.

2.2.2 Supervised dimensionality reduction

Supervised dimensionality reduction is a less-known but powerful approach when the main goal is to perform classification or regression rather than learn a data projection into a lower dimensional space. The methods of this subfield are usually based on classical unsupervised dimensionality reduction, adding the regression or classification loss in the optimization phase. In this way, the reduced dataset $\mathbf{U}$ is the specific projection that allows maximizing the performance of the considered supervised problem. This is usually done in classification settings, minimizing the distance within the same class and maximizing the distance between different classes in the same fashion as Linear Discriminant Analysis (Fisher 1936). The other possible approach is to directly integrate the loss function for classification or regression. Following the taxonomy presented in (Chao et al. 2019), these supervised approaches can be divided into PCA-based, NMF-based (mostly linear), and manifold-based (mostly non-linear).

A well-known PCA-based algorithm is Supervised PCA. The most straightforward approach of this kind has been proposed in (Bair et al. 2006), which is a heuristic that applies classical PCA only to the subset of features mostly related to the

target. A more advanced approach can be found in (Barshan et al. 2011), where the original dataset is orthogonally projected onto a space where the features are uncorrelated, simultaneously maximizing the dependency between the reduced dataset and the target by exploiting Hilbert-Schmidt independence criterion. The goal of Supervised PCA is similar to that of the algorithm proposed in this paper. The main difference is that we are not looking for an orthogonal projection, but we aggregate features by computing their means (thus, two projected features can be correlated) to preserve interpretability. Many variants of Supervised PCA exist, e.g., to make it a non-linear projection or to make it able to handle missing values (Yu et al. 2006). Since it is defined in the same context (linear) and has the same final purpose (minimize the mean squared regression error), supervised-PCA will be compared with the approach proposed by this paper in the experimental section. NMF-based algorithms (Jing et al. 2012; Lu et al. 2016) have better interpretability than PCA-based, but they focus on the non-negativity property of features, which is not a general property of linear problems. Manifold-based methods (Ribeiro et al. 2008; Zhang et al. 2018; Zhang 2009; Raducanu and Dornaika 2012), on the other hand, perform non-linear projections with higher computational costs. Therefore, both families of techniques will not be considered in this linear context.

3 Proposed methodology

In this section, we introduce the proposed dimensionality reduction algorithm, named *Linear Correlated Features Aggregation* (LinCFA), from a general perspective. The approach is based on the following simple idea. Starting from the features x_i of the D-dimensional vector X, we build the aggregated features u_k of the d-dimensional vector U. The dimensionality reduction function $\boldsymbol{\phi}$ is fully determined by a partition $\boldsymbol{\mathcal{P}} = \{\mathcal{P}_1, \dots, \mathcal{P}_d\}$ of the set of features $\{x_1, \dots, x_D\}$. In particular, each feature x_i is assigned to a set $\mathcal{P}_k \in \boldsymbol{\mathcal{P}}$ and each feature u_k is computed as the average of the features in the k-th set of $\boldsymbol{\mathcal{P}}$:

$$u_k = \frac{1}{|\mathcal{P}_k|} \sum_{i \in \mathcal{P}_k} x_i. \tag{1}$$

In the following sections, we will focus on finding theoretical guarantees to determine how to build the partition $\boldsymbol{\mathcal{P}}$. Intuitively, two features will belong to the same element of the partition $\boldsymbol{\mathcal{P}}$ if their correlation is larger than a threshold. This threshold is formalized as the minimum correlation for which the Mean Squared Error (*MSE*) of the regression with a single aggregated feature (i.e., the average) is not worse than the *MSE* with the two separated features.[1] In particular, it is possible to decompose the *MSE* as follows (bias-variance decomposition (Hastie et al. 2009)):

[1] For this reason, the approach can be considered *supervised*.

$$\underbrace{\mathbb{E}_{x,y,\mathcal{T}}[(h_{\mathcal{T}}(x)-y)^2]}_{\text{MSE}} = \underbrace{\mathbb{E}_{x,\mathcal{T}}[(h_{\mathcal{T}}(x)-\bar{h}(x))^2]}_{\text{variance}} + \underbrace{\mathbb{E}_{x}[(\bar{h}(x)-\bar{y}(x))^2]}_{\text{bias}} + \underbrace{\mathbb{E}_{x,y}[(\bar{y}(x)-y)^2]}_{\text{noise}}, \tag{2}$$

where x, y are the features and the target of a test sample, $\mathcal{T}$ is the training set, $h_{\mathcal{T}}(\cdot)$ is the ML model trained on dataset $\mathcal{T}$, $\bar{h}(\cdot)$ is its expected value w.r.t. the training set $\mathcal{T}$ and $\bar{y}$ is the expected value of the test output target y w.r.t. the input features x. Decreasing model complexity leads to a decrease in variance and an increase in bias. Therefore, in the analysis, we will compare these two variations and identify a threshold as the minimum value of correlation for which, after the aggregation, the decrease of variance is greater or equal than the increase of bias, so that the *MSE* will be greater or equal than the original one.

4 Two-dimensional analysis

This section introduces the theoretical analysis, performed in the bivariate setting, that identifies the minimum value of the correlation between the two features for which it is convenient to aggregate them with their mean. In particular, Sect. 4.1 introduces the assumptions under which the analysis is performed. Subsection 4.2 computes the amount of variance decreased when performing the aggregation. Then, Sect. 4.3 evaluates the amount of bias increased due to the aggregation. Finally, Sect. 4.4 combines the two results identifying the minimum amount of correlation for which it is profitable to aggregate the two features. In addition, Appendix A contains the proofs and technical results that are not reported in the main paper, Appendix B includes an additional finite-sample analysis, and Appendix C computes confidence intervals that allow stating the results with only theoretical or empirical quantities.

4.1 Setting

In the two-dimensional case ($D = 2$), we consider the relationship between the two features x_1, x_2 and the target y to be linear and affected by Gaussian noise: $y = w_1 x_1 + w_2 x_2 + \epsilon$, with $\epsilon \sim \mathcal{N}(0, \sigma^2)$. As usually done in linear regression (Johnson and Wichern 2007), we assume the training dataset $\mathbf{X}$ to be known. Moreover, recalling the zero-mean assumption ($\mathbb{E}[x_1] = \mathbb{E}[x_2] = 0$), it follows $\mathbb{E}[y] = w_1\mathbb{E}[x_1] + w_2\mathbb{E}[x_2] = 0$ and $\sigma_y^2 = \sigma^2$.

We compare the performance (in terms of bias and variance) of the two-dimensional linear regression $\hat{y} = \hat{w}_1 x_1 + \hat{w}_2 x_2$ with the one-dimensional linear regression, which takes as input the average between the two features $\hat{y} = \hat{w}\frac{x_1+x_2}{2} = \hat{w}\bar{x}$. As a result of this analysis, we will define conditions under which aggregating features x_1 and x_2 in the feature $\bar{x}$ is convenient.

4.2 Variance analysis

In this subsection, we compare the variance of the two models with both an asymptotic and a finite-samples analysis. Since the two-dimensional model estimates two coefficients, it is expected to have a larger variance. Instead, aggregating the two features reduces the variance of the model.

4.2.1 Variance of the estimators

A quantity, necessary to compute the variance of the models that will be compared throughout this subsection, is the covariance matrix of the vector $\hat{w}$ of the estimated regression coefficients w.r.t. the training set. Given the training features $\mathbf{X}$, a known result in a general linear problem with n samples and D features (Johnson and Wichern 2007) (see Appendix A for the computations) is:

$$var_{\mathcal{T}}(\hat{w}|\mathbf{X}) = (\mathbf{X}^T\mathbf{X})^{-1}\sigma^2. \tag{3}$$

The following lemma shows the variance of the weights for the two specific models that we are comparing.

Lemma 1 *Let the real model be linear with respect to the features x_1 and x_2 ($y = w_1x_1 + w_2x_2 + \epsilon$). In the one-dimensional case $\hat{y} = \hat{w}\bar{x}$, we have*:

$$var_{\mathcal{T}}(\hat{w}|\mathbf{X}) = \frac{\sigma^2}{(n-1)\hat{\sigma}^2_{\bar{x}}}. \tag{4}$$

In the two-dimensional case $\hat{y} = \hat{w}_1x_1 + \hat{w}_2x_2$, we have:

$$\begin{aligned} var_{\mathcal{T}}(\hat{w}|\mathbf{X}) = &\frac{\sigma^2}{(n-1)(\hat{\sigma}^2_{x_1}\hat{\sigma}^2_{x_2} - c\hat{o}v(x_1,x_2)^2)} \\ &\times \begin{bmatrix} \hat{\sigma}^2_{x_2} & -c\hat{o}v(x_1,x_2) \\ -c\hat{o}v(x_1,x_2) & \hat{\sigma}^2_{x_1} \end{bmatrix}. \end{aligned} \tag{5}$$

Proof The proof of the two results follows from Equation (3), see Appendix A for the computations. □

4.2.2 Variance of the model

Recalling the general definition of variance of the model from Equation (2), in the specific case of linear regression it becomes:

$$\mathbb{E}_{x,\mathcal{T}}[(h_{\mathcal{T}}(x) - \bar{h}(x))^2] = \mathbb{E}_{x,\mathcal{T}}[(\hat{w}^Tx - \mathbb{E}_{\mathcal{T}}[\hat{w}^Tx])^2]. \tag{6}$$

The following result shows the variance of the two specific models (univariate and bivariate) considered in this section.

Theorem 1 *Let the real model be linear with respect to the two features x_1 and x_2 ($y = w_1 x_1 + w_2 x_2 + \epsilon$). Then, in the one dimensional case $y = \hat{w}\frac{x_1+x_2}{2} = \hat{w}\bar{x}$, we have*:

$$\mathbb{E}_{x,\mathcal{T}}[(h_{\mathcal{T}}(x) - \bar{h}(x))^2 | \mathbf{X}] = \sigma^2_{x_1+x_2} \frac{\sigma^2}{(n-1)\hat{\sigma}^2_{x_1+x_2}}. \tag{7}$$

In the two dimensional case $y = \hat{w}_1 x_1 + \hat{w}_2 x_2$, we have:

$$\begin{aligned} &\mathbb{E}_{x,\mathcal{T}}[(h_{\mathcal{T}}(x) - \bar{h}(x))^2 | \mathbf{X}] \\ &= \frac{\sigma^2(\sigma^2_{x_1}\hat{\sigma}^2_{x_2} + \sigma^2_{x_2}\hat{\sigma}^2_{x_1} - 2cov(x_1,x_2)\hat{cov}(x_1,x_2))}{(n-1)(\hat{\sigma}^2_{x_1}\hat{\sigma}^2_{x_2} - \hat{cov}(x_1,x_2)^2)}. \end{aligned} \tag{8}$$

Proof The proof combines the results of Lemma 1 with the definition of variance for a linear model given in Equation (6). The detailed proof can be found in Appendix A. □

4.2.3 Comparisons

In this subsection, the difference between the variance of the linear regression with two features x_1 and x_2 and the variance of the linear regression with one feature $\bar{x} = \frac{x_1+x_2}{2}$ is shown. We will prove that, as expected, this difference is positive and it represents the reduction of variance when substituting a two-dimensional random vector with the average of its components.

First, the *asymptotic* analysis is performed, obtaining a result that can be applied with good approximation when a large number of samples n is available. Then, the analysis is repeated in the *finite-samples* setting, with an additional assumption on the variance and sample variance of the features x_1 and x_2, that simplify the computations.[2]

Case I: **asymptotic analysis.** The estimators that we are considering are *consistent*, i.e., they converge in probability to the real values of the parameters (e.g., $\text{plim}_{n\to\infty}\hat{\sigma}^2_{x_1} = \sigma^2_{x_1}$). Therefore the following result can be proved.

Theorem 2 *If the number of samples n tends to infinity, let $\Delta^{n\to\infty}_{var}$ be the difference between the variance of the two-dimensional and the one-dimensional linear models, it is equal to*:

$$\Delta^{n\to\infty}_{var} = \frac{\sigma^2}{n-1} \geq 0, \tag{9}$$

[2] The assumption that we will introduce for the finite-samples setting might be restrictive. However, it allows simplifying the computations. A more general finite-sample analysis has also been performed, only assuming unitary variances. This more general analysis leads to more convolute expressions and for this reason it is reported in Appendix B.

that is a positive quantity and tends to zero when the number of samples tends to infinity.

Proof The result follows from the difference between Eqs. 8 and 7, exploiting the consistency of the estimators. □

Case II: finite-samples analysis with equal variance and sample variance. For the finite-samples analysis, we add the following assumption to simplify the computations:

$$\begin{cases} \sigma_{x_1} = \sigma_{x_2} =: \sigma_x \\ \hat{\sigma}_{x_1} = \hat{\sigma}_{x_2} =: \hat{\sigma}_x. \end{cases} \tag{10}$$

Theorem 3 *If the conditions of Equation* (10) *hold, let* Δ_{var} *be the difference between the variance of the two-dimensional and the one-dimensional linear models, it is always non-negative and it is equal to*:

$$\Delta_{var} = \frac{\sigma^2}{n-1} \cdot \frac{\sigma_x^2(1-\rho_{x_1,x_2})}{\hat{\sigma}_x^2(1-\hat{\rho}_{x_1,x_2})}. \tag{11}$$

Proof The proof starts again from the variances of the two models found in Theorem 1 and it performs algebraic computations exploiting the assumption stated in Equation (10). All the steps can be found in Appendix A. □

Remark 3 When the number of samples n tends to infinity, the result of Equation (11) reduces to the asymptotic case, as in Equation (9).

Remark 4 The quantities found in Theorem 2 and 3 are always non-negative, meaning that the variance of the two-dimensional case is always greater or equal than the corresponding one-dimensional version, as expected.

4.3 Bias analysis

In this subsection, we compare the (squared) bias of the two models under examination with both an asymptotic and a finite-samples analysis, as done in the previous subsection for the variance. Since the two-dimensional model corresponds to a larger hypothesis space it is expected to have a lower bias w.r.t. the one-dimensional.

The procedure to derive the difference between biases is similar to the one followed for the variance. The first step is to compute the expected value w.r.t. the training set $\mathcal{T}$ of the vector $\hat{w}$ of the regression coefficients estimates, given the training features $\mathbf{X}$. This is used to compute the bias of the models. In particular, in Equation (2), we defined the (squared) bias as follows:

$$\mathbb{E}_x[(\bar{h}(x) - \bar{y})^2] = \mathbb{E}_x[(\mathbb{E}_{\mathcal{T}}[h(x)] - \mathbb{E}_{y|x}[y])^2]. \tag{12}$$

Starting from this definition, the bias of the one-dimensional case $\hat{y} = \hat{w}\bar{x}$ is computed. Moreover, for the two dimensional case $y = \hat{w}_1 x_1 + \hat{w}_2 x_2$ the model is clearly unbiased. Detailed computations can be found in Appendix A.

After the derivation of the bias of the models, the same asymptotic and finite-samples analysis performed on the variance is repeated in this section for the (squared) bias. Since the two-dimensional model is unbiased, we can conclude that the increase of the bias component of the loss, when the two features are substitute by their mean, is equal to the bias of the one-dimensional model.

Case I: asymptotic analysis. When the number of samples n of the training dataset $\mathcal{T}$ approaches infinity, recalling that the estimators considered converge in probability to the expected values of the parameters, the following result holds.

Theorem 4 *If the number of samples n tends to infinity, let* $\Delta_{bias}^{n\to\infty}$ *be the difference between the bias of the one-dimensional and the two-dimensional models, it is equal to*:

$$\Delta_{bias}^{n\to\infty} = \frac{\sigma_{x_1}^2 \sigma_{x_2}^2 (1 - \rho_{x_1,x_2}^2)(w_1 - w_2)^2}{\sigma_{x_1+x_2}^2} \tag{13}$$

$$= \frac{(1 - \rho_{x_1,x_2})(w_1 - w_2)^2}{2}, \tag{14}$$

where the second equality holds if $\sigma_{x_1} = \sigma_{x_2} = 1$.

Proof The proof starts from the bias of the two models computed in Appendix A and exploits the fact that in the limit $n \to \infty$, it is possible to substitute every sample estimator with the real quantity of the parameters because they are consistent estimators. Details can be found in Appendix A. □

Case II: finite-samples analysis with equal variance and sample variance In the finite-samples case, we provide the same analysis performed for variance, i.e., with the assumptions of Equation (10).

Theorem 5 *If the conditions of Equation* (10) *hold, let* Δ_{bias} *be the difference between the* (*squared*) *bias of the one-dimensional and the two-dimensional linear models, then it has value*:

$$\Delta_{bias} = \frac{\sigma_x^2 (1 - \rho_{x_1,x_2})(w_1 - w_2)^2}{2}. \tag{15}$$

Proof The proof starts from the bias of the two models and performs algebraic computations exploiting the assumptions of Equation (10). All the steps can be found in Appendix A. □

Remark 5 When the number of samples n tends to infinity, the result in Equation (15) reduces to the asymptotic case as in Theorem 4.

Remark 6 Some observations are in order:

- As expected, the quantities found in Theorem 4, 5 are always non-negative, since the hypothesis space of the univariate model is a subset of the one of the bivariate model.
- We observe that $\Delta_{bias} = 0$ if $\rho_{x_1,x_2} = 1$. Indeed, when the two variables are perfectly (positively) correlated their coefficients in the linear regression are equal, therefore there is no loss of information in their aggregation.
- Finally, when the two regression coefficients are equal $w_1 = w_2$ there is no increase of bias due to the aggregation, since it is enough to learn a single coefficient $\bar{w}$ to have the same performance of the bivariate model.

4.4 Correlation threshold

This subsection concludes the analysis with two features by comparing the reduction of variance with the increase of bias when aggregating the two features x_1 and x_2 with their average $\bar{x} = \frac{x_1+x_2}{2}$. In conclusion, the result shows when it is convenient to aggregate the two features with their mean, in terms of mean squared error.

Considering the asymptotic case, the following theorem compares bias and variance of the models.

Theorem 6 *When the number of samples n tends to infinity and the relationship between the features and the target is linear with Gaussian noise, the decrease of variance is greater than the increase of* (*squared*) *bias when the two features* x_1 *and* x_2 *are aggregated with their average if and only if*:

$$\rho^2_{x_1,x_2} \geq 1 - \frac{\sigma^2\sigma^2_{x_1+x_2}}{(n-1)\sigma^2_{x_1}\sigma^2_{x_2}(w_1-w_2)^2}, \tag{16}$$

that, for $\sigma_{x_1} = \sigma_{x_2} = 1$ *becomes*:

$$\rho_{x_1,x_2} \geq 1 - \frac{2\sigma^2}{(n-1)(w_1-w_2)^2}. \tag{17}$$

Proof Computing the difference between Eqs. (9) and (13) the result follows. □

In the finite-samples setting, with the additional assumptions of Eq. (10), the following theorem shows the result of the comparison between bias and variance of the two models.

Theorem 7 *Let the variance and sample variance of the features x_1 and x_2 be equal* (*Eq.* (10)) *and the relationship between the features and then target be linear with Gaussian noise. The decrease of variance is greater than the increase of* (*squared*) *bias when the two features x_1 and x_2 are aggregated with their average if and only if*:

$$\hat{\rho}_{x_1,x_2} \geq 1 - \frac{2\sigma^2}{(n-1)\hat{\sigma}_x^2(w_1 - w_2)^2}, \tag{18}$$

that, for $\hat{\sigma}_x = 1$ becomes:

$$\hat{\rho}_{x_1,x_2} \geq 1 - \frac{2\sigma^2}{(n-1)(w_1 - w_2)^2}. \tag{19}$$

Proof Computing the difference between Equation (11) and (15) the result follows. □

Remark 7 The results of Theorem 6 and 7 comply with the intuition that, in a linear setting with two features, they should be aggregated if their correlation is *large enough*.

Remark 8 Theorem 6 and 7 with unitary sample variances produce the same threshold both in the finite and the asymptotic settings.

In conclusion, the thresholds found in Theorem 6 and 7 show that it is profitable in terms of *MSE* to aggregate two variables in a bivariate linear setting with Gaussian noise if:

- the variance of the noise σ^2 is large, which means that the process is noisy and the variance should be reduced;
- the number of samples n is small, indeed in this case there is little knowledge about the actual model, therefore it is better to learn one parameter rather than two;
- the difference between the two coefficients $w_1 - w_2$ is small, which implies that they are similar, and learning a single coefficient introduces a little loss of information.

5 Generalization: three-dimensional and D-dimensional analysis

In the previous section, we focused on aggregating two features in a bivariate setting. In this section, we extend that approach to three features. Starting from the related results, we will straightforwardly extend them to a general problem with D features, heuristically considering the $D-2$ remaining features as a unique third contribution. Given the complexity of the computations, we focus on asymptotic analysis only. After the analysis, we conclude this section with

the main algorithm proposed in this paper: *Linear Correlated Features Aggregation* (LinCFA).

5.1 Three-dimensional case

In the three-dimensional case ($D = 3$), we consider the relationship between the three features and the target to be linear with Gaussian noise: $y = w_1 x_1 + w_2 x_2 + w_3 x_3 + \epsilon$, $\epsilon \sim \mathcal{N}(0, \sigma^2)$. In accordance with the previous analysis, we assume the training dataset $\mathbf{X} = [\mathbf{x_1}\ \mathbf{x_2}\ \mathbf{x_3}]$ to be known and recalling the zero-mean assumption ($\mathbb{E}[x_1] = \mathbb{E}[x_2] = \mathbb{E}[x_3] = 0$) it follows $\mathbb{E}[y] = w_1\mathbb{E}[x_1] + w_2\mathbb{E}[x_2] + w_3\mathbb{E}[x_3] = 0$, $\sigma_y^2 = \sigma^2$.

In this setting and for the general D dimensional setting of the next subsection, which will be a direct application of this, we compare the performance of the bivariate linear regression $\hat{y} = \hat{w}_i x_i + \hat{w}_j x_j$ of each pair of features x_i, x_j with the univariate linear regression that considers their average $\hat{y} = \hat{w}\frac{x_i + x_j}{2} = \hat{w}\bar{x}$, to decide whether it is convenient to aggregate them or not in terms of *MSE*. Indeed, extending the dimension from $D = 2$ to a general dimension D, and comparing all the possible models where groups of variables are aggregated, is combinatorial in the number of features and it would be impractical. Also, comparing the full D dimensional regression model with the $D - 1$ dimensional model where two variables are aggregated is impractical. Indeed, when the number of features is huge, in addition to a polynomial computational cost, both models suffer issues like the curse of dimensionality and risk of overfitting.

To simplify the exposition, for the theoretical analysis, we will consider $x_i = x_1, x_j = x_2$. Moreover, in the following subsection we will directly report the asymptotic correlation threshold that guarantees the asymptotic decrease of variance to be greater than the increase of bias due to the aggregation of two features. The specific analysis of variance and bias, together with the related proofs, can be found in Appendix D.

5.1.1 Correlation threshold

The result of the following theorem extends the result of Theorem 6 for the three-dimensional setting.

Theorem 8 *In the asymptotic setting, let the relationship between the features and the target be linear with Gaussian noise. Assuming unitary variances of the features* $\sigma_{x_1} = \sigma_{x_2} = \sigma_{x_3} = 1$, *the decrease of variance is greater than the increase of* (*squared*) *bias due to the aggregation of the features* x_1 *and* x_2 *with their average if and only if*:

$$1-(a-b)-\sqrt{a(a-2b)} \leq \rho_{x_1,x_2} \leq 1-(a-b)+\sqrt{a(a-2b)},$$
$$\text{with} \begin{cases} a = \frac{\sigma^2}{(n-1)(w_1-w_2)^2} \\ b = \frac{(\rho_{x_1,x_3}-\rho_{x_2,x_3})w_3}{(w_1-w_2)} \end{cases}. \tag{20}$$

Proof The result follows after algebraic computations on the difference $\Delta_{var}^{n\to\infty} - \Delta_{Bias}^{n\to\infty} \geq 0$, where the expression of the asymptotic difference of variances and biases can be respectively found in Remark 18 and Theorem 18 of Appendix D. □

Remark 9 Equation (20) holds also in the case of generic variance $\sigma_{x_3}^2$ of the feature x_3, with the only difference that b becomes:

$$b = \frac{\sigma_{x_3}(\rho_{x_1,x_3} - \rho_{x_2,x_3})w_3}{(w_1 - w_2)}. \tag{21}$$

Remark 10 The result obtained in this section with three features is more difficult to interpret than the bivariate one. However, if the two features x_1 and x_2 are uncorrelated with the third feature x_3 or they have the same correlation with it ($\rho_{x_1,x_3} = \rho_{x_2,x_3}$), then Equation (20) is equal to the one found in the bivariate asymptotic analysis (Equation (17)).

Remark 11 Since the analysis is asymptotic, the theoretical quantities in Equation (20) can be substituted with their consistent estimators when the number of samples n is *large*.

5.2 D-dimensional case

This last subsection of the analysis shows a generalization from three to D dimensions. In particular, we assume the relationship between the D features $x_1, ..., x_D$ and the target to be linear with Gaussian noise $y = w_1x_1 + ... + w_Dx_D + \epsilon$, with $\epsilon \sim \mathcal{N}(0, \sigma^2)$. As done throughout the paper, we assume the training dataset $\mathbf{X} = [\mathbf{x_1} \; ... \; \mathbf{x_D}]$ to be known and from the zero-mean assumption $\mathbb{E}[y] = 0$ and $\sigma_y^2 = \sigma^2$.

As discussed for the three-dimensional case, we compare the performance (in terms of bias and variance) of the two-dimensional linear regression $\hat{y} = \hat{w}_i x_i + \hat{w}_j x_j$ with the one-dimensional linear regression $\hat{y} = \hat{w}\frac{x_i+x_j}{2} = \hat{w}\bar{x}$ and in the computations we consider $x_i = x_1, x_j = x_2$ without loss of generality.

Considering the linear combination of the remaining features as a unique variable $x = w_3x_3 + ... + w_Dx_D$, we directly extend the three-dimensional analysis of the previous subsection to this general case, considering the model to be

$y = w_1 x_1 + w_2 x_2 + wx + \epsilon$, with $w = 1$ and $x = w_3 x_3 + ... + w_D x_D$. This way, the D-dimensional linear problem is straightforwardly reformulated as a three-dimensional one. However, this analysis can be seen as a heuristic result, in the sense that we do not fully characterize the relationship between the two features under analysis and all the remaining ones, but the focus is only the relationship between the two features under analysis x_1, x_2 and the linear combination $x = w_3 x_3 + ... + w_D x_D$ of the remaining ones.

Recalling that in this case the third feature x has general variance σ_x^2, the following lemma holds.

Lemma 2 *Let* $y = w_1 x_1 + ... + w_D x_D + \epsilon = w_1 x_1 + w_2 x_2 + wx + \epsilon$ *with* $\sigma_{x_1}^2 = \sigma_{x_2}^2 = 1$ *and* $\sigma_x^2 = \sigma_{w_3 x_3 + ... + w_D x_D}^2$. *Then, performing linear regression in the asymptotic setting, the decrease of variance is greater than the increase of bias when aggregating the two features* x_1 *and* x_2 *with their average if and only if the condition on the correlation of Equation* (20) *holds* (*with the parameter b expressed like in Equation* (21) *as* $b = \frac{\sigma_x(\rho_{x_1,x} - \rho_{x_2,x})w}{(w_1 - w_2)}$).

Proof The lemma follows by applying the three-dimensional analysis with general variance of the third feature σ_x^2 (Theorem 8 and Remark 9). □

5.3 D-dimensional algorithm

For the general D-dimensional case, as explained in the previous subsection, the three-dimensional results has be extended considering as third feature the linear combination of the $D - 2$ features not currently considered for the aggregation. A drawback of applying the obtained result in practice is that it requires the knowledge of all the coefficients $w_1, ..., w_D$, which is unrealistic, or to approximate them through an estimate, performing linear regression on the complete D-dimensional dataset. In this case, the computational cost is $\mathcal{O}(n \cdot D^2 + D^3)$—which becomes $\mathcal{O}(n \cdot D^2 + D^{2.37})$ if using the Coppersmith-Winograd algorithm (Coppersmith and Winograd 1990)—and it is impractical with a huge number of features. Therefore, since the equation in the three dimensional asymptotic analysis becomes equal to the bivariate one if the two features have the same correlation with the third (Remark 10), it is reasonable, if they are highly correlated, to assume this to be valid and to apply the asymptotic bivariate result shown in Equation (17) to decide whether the two features should be aggregated or not. In this way, we iteratively try all combinations of two features, with complexity $\mathcal{O}(n + D^2)$ in the worst case, in order to choose the groups of features that is convenient to aggregate with their mean.

Algorithm 1 LinCFA: Linear Correlated Features Aggregation

Input: D-dimensional dataset with randomly shuffled columns $\{x_1, \ldots, x_D\}$; target y; n samples
Output: reduced features $\{\bar{x}_1, \ldots, \bar{x}_d\}$, with $d \leq D$

```
P ← {}                                          ▷ Partition of the features
V ← {}                                          ▷ Set of already considered features
for each i ∈ {1, ..., D} do
    if i ∉ V then
        P ← {i}
        V ← V ∪ {i}
        for each j ∈ {i + 1, ..., D} do
            c ← correlation(x_i, x_j)
            b ← threshold(x_i, x_j, y)
            if c ≥ b then                       ▷ Aggregate the features
                P ← P ∪ {j}
                V ← V ∪ {j}
            end if
        end for
        P ← P ∪ {P}
    end if
end for
d ← |P|
for each k ∈ {1, ..., d} do
    x̄_k = (1/|P_k|) Σ_{i∈P_k} x_i
end for
return {x̄_1, ..., x̄_d}
```

In Algorithm 1 the pseudo-code of the proposed algorithm *Linear Correlated Features Aggregation* (LinCFA) can be found. The proposed dimensionality reduction algorithm creates a d dimensional partition of the indices of the features $\{1, \ldots, D\}$ by iteratively comparing couples of features and adding them to the same subset if their correlation (correlation(x_i, x_j)) is greater than the threshold (threshold(x_i, x_j, y)), obtained from Eq. (17). Then, it aggregates the features in each set k of the partition ($\boldsymbol{\mathcal{P}}$) with their average, producing each output $\bar{x}_k$.

Remark 12 (About theoretical results and the empirical algorithm) The proposed algorithm aggregates sets of features and not only couples of them, as considered in the theoretical analysis. The motivation behind this choice is to perform a single average of a set of features. A possible variation, which aggregates pairs of features as derived by the theory, is to directly aggregate a pair of features with their mean, once they respect the theoretical aggregation condition (e.g., at the first iteration we aggregate x_1, x_2 producing $\bar{x} = \frac{x_1 + x_2}{2}$). Then, considering their mean from there on as a single feature, it would be aggregated with another feature if the condition is

respected (e.g., at the second iteration we aggregate $\bar{x}, x_3$ producing $\hat{x} = \frac{\bar{x}+x_3}{2}$), until no more aggregations are possible. This procedure adheres more with the theoretical results, however it is less interpretable in the sense discussed in Remark 2, since each reduced feature is an iterative mean of means.

Remark 13 (About the ordering of features) The output of the LinCFA algorithm may depend on the ordering of the features. Therefore, in the pseudo-code of Algorithm 1, a random shuffle of the original features is required as input, such that systematic biases due to the ordering are avoided. A greedy approach that removes the dependency of the *LinCFA* algorithm on the ordering of the features would be to introduce an internal ordering among pairs of features. Considering for example correlation, one may consider the pair of the two most correlated features and test if they exceed the threshold. If so, they could be added to a cluster and substituted with their mean. Iteratively proceeding in this way, until all features have been assigned to a set of the partition, produces an algorithm that becomes independent from the initial ordering of features and aggregates only features that exceed the threshold. However, this increases the memory and computational complexity, since all the correlations between each pair of features should be computed and stored.

As a further step, among the possible partitions that can be identified depending on the ordering, there is at least an optimal partition of features, which maximizes the mean squared error. Intuitively, with infinite samples, the MSE is maximized considering each feature independently. This is confirmed by the asymptotic variance analysis, where a term n shows that, with infinite samples, there is no decrease of variance with the aggregation. However, with finite samples, the identification of the optimal partition would be combinatorial, since all the possible partitions should be tested. The proposed algorithm adds one feature at time in a cluster, therefore it has no guarantees of optimality. This is in line with classical machine learning approaches such as forward feature selection, that iteratively selects a promising feature, although a combination of other two features may be more informative.

6 Numerical validation

In this section, the theoretical results obtained in Sects. 4 and 5 are exploited to perform dimensionality reduction on synthetic datasets of two, three and D dimensions. Furthermore, the proposed dimensionality reduction approach *LinCFA* is applied to real datasets and compared with state-of-the-art benchmark methods. To evaluate the performance of the regressions, the results will be evaluated in terms of Mean Squared Error (*MSE*), R-squared (R^2) and Relative *RMSE* (*RRMSE*). Code and datasets can be found at the following link: https://github.com/PaoloBonettiPolimi/PaperLinCFA.

6.1 Two-dimensional application

In the bivariate setting, according to Eq. (17) and (19), it is convenient to aggregate the two features with a small number of samples n, with a small absolute value of the difference between the coefficients of the linear model w_1, w_2 or with a large variance of the noise σ^2. The synthetic experiments (full description in Appendix E) confirm with data the theoretical result. In particular, they are performed with a fixed number of samples $n = 500$, a fixed correlation between the features $\rho_{x_1,x_2} \approx 0.9$, comparing two combinations of weights (at small and large distances) and three different variances of the noise (small, normal, large).

Table 1 shows the results of the experiments (more detailed results can be found in Tables 6,7 of Appendix E). In line with the theory, when the weights in the linear model are consistently distant, only with a huge variance of the noise the threshold is far from 1 and the two features are aggregated, while for a reasonably small amount of variance in the noise they are kept separated. On the other hand, when the weights in the linear model are similar, the threshold of Eq. (17) is small and the conclusion is to aggregate the two features also with a small amount of variance in the noise. The confidence intervals on the R^2 and on the *MSE* confirm that, when the correlation is above the threshold, the performance of the linear model when the two features are aggregated with their average is statistically not worse than the bivariate model where they are kept separate. It is finally important to notice that, knowing the coefficients of the regression, always leads to aggregate the two features or not in all the 500 repetitions of the experiment (row *# aggregations (theo)*). On the contrary, estimating the coefficients from data leads to the same action in most repetitions but not always (row *# aggregations (emp)*), since the limited amount of data introduces noise into the estimates.

6.2 Three-dimensional application

Equation (20) expresses the interval for which it is convenient to aggregate the two features $x_1$and x_2 in the three-dimensional setting. As in the bivariate case, it is related to the number of samples, the difference between weights, and the variance of the noise. In addition, it also depends on the difference of the correlations between each of the two features with the third one x_3 and on the weight w_3.

The experiment performed in this setting is based on synthetic data, computed with the following realistic setting: the weights $w_1 = 0.4$, $w_2 = 0.6$ are closer than $w_3 = 0.2$. Moreover, the two features are significantly correlated: $\rho_{x_1,x_2} \approx 0.88$ (more details can be found in Appendix E).

In this setting, as shown in Table 2, it is convenient to aggregate the two features x_1, x_2 with their average both in terms of *MSE* and R^2, since the aggregation does not worsen the performances. In particular, the aggregation is already convenient with a small standard deviation of the noise ($\sigma = 0.5$).

Table 1 Experiment on synthetic bivariate data for two combinations of weights and three different values of variance of the noise

Quantity	95% Confidence Interval ($w_1 = 0.2, w_2 = 0.8$)			95% Confidence Interval ($w_1 = 0.47, w_2 = 0.52$)		
	$\sigma = 0.5$	$\sigma = 1$	$\sigma = 10$	$\sigma = 0.5$	$\sigma = 1$	$\sigma = 10$
# aggregations (theo)	**0**	**0**	**500**	**500**	**500**	**500**
# aggregations (emp)	**0**	**24**	**332**	**314**	**339**	**346**
R^2 full	0.781 ± 5.2e−5	0.487 ± 1.23e−4	0.010 ± 2.64e−4	0.794 ± 6.4e−5	0.505 ± 9.1e−5	0.012 ± 2.31e−4
R^2 aggregate	0.765 ± 3.0e−5	0.486 ± 7.8e−5	0.011 ± 1.69e−4	0.794 ± 6.0e−5	0.506 ± 7.5e−5	0.015 ± 2.0e−5
MSE full	0.275 ± 6.5e−5	1.000 ± 2.40e−4	103.021 ± 0.027	0.263 ± 8.2e−5	0.949 ± 1.75e−4	94.573 ± 2.209e−3
MSE aggregate	0.295 ± 3.7e−5	1.004 ± 1.52e−4	102.977 ± 0.018	0.262 ± 7.7e−5	0.947 ± 1.45e−4	94.363 ± 1.914e−3

The number of aggregations performed by LinCFA in 500 repetitions is reported in bold

Table 2 Synthetic experiment in the three dimensional setting comparing the full model with three variables with the bivariate model where x_1, x_2 are aggregated with their mean

Quantity	95% Confidence interval
# Aggregations (theo)	**500**
# Aggregations (emp)	**335**
R^2 full	$0.825 \pm 6e{-6}$
R^2 aggregate	$0.825 \pm 5e{-6}$
MSE full	$0.285 \pm 9e{-6}$
MSE aggregate	$0.286 \pm 8e{-6}$

The number of aggregations performed by LinCFA in 500 repetitions is reported in bold

Table 3 Synthetic experiment in the D dimensional setting. The experiment has been repeated twice: considering the theoretical threshold with the exact coefficients (*theo*) and with coefficients estimated from data (*emp*)

Quantity	95% Confidence interval
R^2 full	$0.828 \pm 1.46e{-4}$
R^2 aggregate (theo)	$0.890 \pm 4.8e{-5}$
R^2 aggregate (emp)	$0.881 \pm 1.07e{-4}$
MSE full	157.346 ± 0.120
MSE aggregate (theo)	100.536 ± 0.040
MSE aggregate (emp)	108.725 ± 0.088
Number of reduced variables (theo)	**4**
Number of reduced variables (emp)	**15**

The number of reduced features aggregated by LinCFA algorithm is reported in bold

6.3 *D*-dimensional application

This subsection introduces the D-dimensional synthetic experiment performed 500 times with $n = 500$ samples and $D = 100$ features, reduced with the proposed algorithm *LinCFA* (more details can be found in Appendix E).

The test results shown in Table 3 underline that knowing the real values of the coefficients of the linear model would lead to a reduced dataset of $d = 4$ features and a significant increase of performance (R^2 *aggregate (theo)*, *MSE aggregate (theo)*), while using the empirical coefficients the dimension is reduced to $d = 15$, still with a significant increase of performance both in terms of MSE and R^2 (R^2 *aggregate (emp)*, *MSE aggregate (emp)*). This is a satisfactory result and it is confirmed by the real dataset application described below.

To better understand the performance of the algorithm, in Fig. 1 we consider the number of selected features and the regression scores. From Fig. 2a it is clear that with a small number of samples, both considering theoretical and empirical quantities, the number of reduced features d becomes smaller to prevent overfitting. Moreover, considering the empirical quantities, which are the only ones available in practice, lead to a larger number of reduced features (but still significantly smaller than the original dimension D). Figure 2b, c show the performance of the linear regression

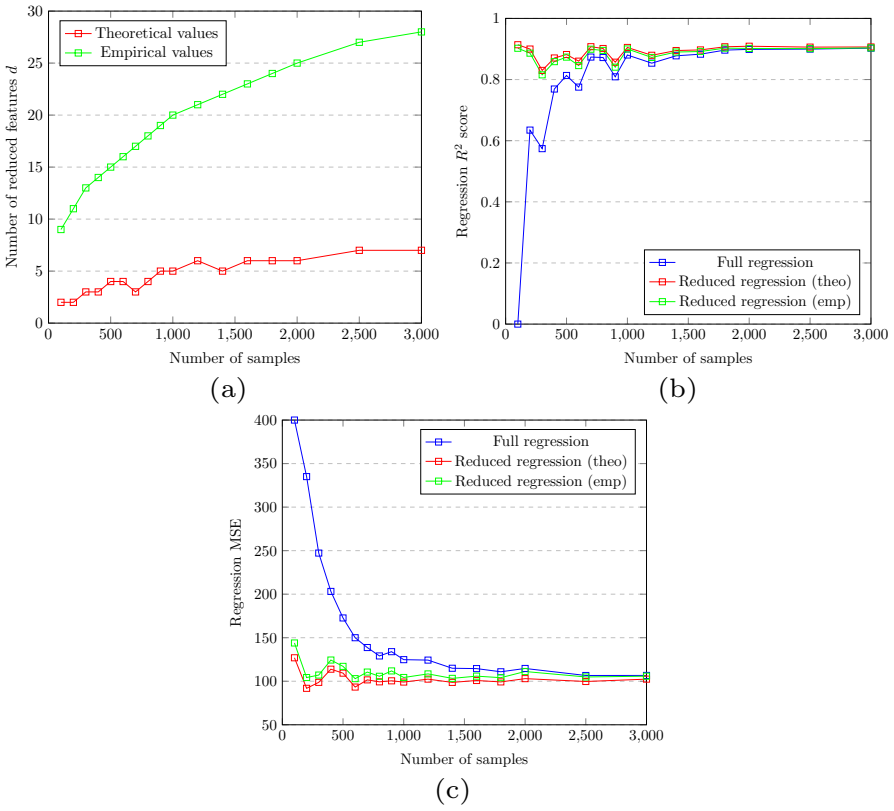

Fig. 2 Figure 2a shows the number of reduced features for a different number of samples. Figure 2b,2c show the regression performance in terms of R^2 and *MSE* for different number of samples. Blue lines refer to the linear regression with all the original features, while red and green lines respectively refer to linear regression on the features reduced by applying the proposed algorithm considering theoretical and empirical quantities (Color figure online)

considering the reduced features compared with the full dataset. When the number of samples is significantly larger than the number of features, the performance of the reduced datasets is only slightly better but, when the number of samples is of the same order of magnitude as the number of features, the reduced datasets (both considering empirical and theoretical quantities) significantly outperform the regression over the full dataset. Moreover, the regression performed with reduced datasets is much robust, since it has a score that is stable for different numbers of samples.

Real-world experiments. The main practical result introduced in this paper (the algorithm *LinCFA*) has been also tested on real datasets. In particular, the results of the application of the dimensionality reduction method introduced in this paper are discussed in comparison with the chosen baselines.

Specifically, the *LinCFA* algorithm has been applied in comparison with *classical (unsupervised) PCA, Supervised PCA, LLE, LPP, Isomap and Kernel PCA*. Additionally, RReliefF has also been considered to take into account a feature selection

method as baseline. The number of components selected for PCA is set to explain 95% of variance, while for Supervised PCA, LLE, LPP, Isomap, Kernel PCA and RReliefF the best result (evaluating from $d = 1$ to $d = 50$ principal components) has been considered. All the other hyperparameters of the methods have been set to default values. Linear, polynomial and sigmoidal kernels have been considered for Kernel and Supervised PCA. The mean squared error (*MSE*), the relative root mean squared error (*RRMSE*) and the coefficient of determination (R^2) of the linear regression, applied on the set of components reduced by the each algorithm under analysis, have been considered as performance measures on the test set. Confidence intervals have been produced bootstrapping the training and validation set with five different seeds. Additionally, the performances of the full dataset applying linear, Ridge (Hoerl and Kennard 1970), and Lasso (Tibshirani 1996) regression have been considered. To further test the *LinCFA* algorithm in comparison with non-linear regression approaches, together with linear regression, support vector machine for regression (Drucker et al. 1996)[3], XGBoost (Chen and Guestrin 2016)[4] and a neural network (Lawrence 1993)[5] have been performed and the related results are available in Appendix E.4. Moreover, in Appendix E.4, also the result of each baseline considering the same number of reduced features selected by the LinCFA algorithm is reported.

Eight datasets with different characteristics have been considered.

The first dataset focuses on the prediction of *life expectancy* from $D = 18$ continuous factors and 1649 samples. The dataset is available on Kaggle[6]. In this case, a reduction of the number of features may be unnecessary, as confirmed by the experimental results, where the full dataset have similar performances w.r.t. the majority of the benchmark methods and the LinCFA algorithm. This experiment provides an example that shows that the algorithm does not reduce too much the dimensionality, when it is not necessary.

The second dataset is a *financial* dataset made of $D = 75$ continuous features, 1299 samples, and a scalar output. The model predicts the cash ratio depending on other metrics, from which it is possible to derive many fundamental indicators. The dataset is available on Kaggle[7]. Given the consistent number of features w.r.t. the number of samples, linear regression with the full dataset provides negative results, while the application of linear regression on the reduced dataset has a significantly high score, and the LinCFA algorithm has one of the best performances among the methods considered.

Then, the algorithm is tested on two *climatological* dataset composed by $D = 136$ (with 1038 samples) and $D = 1991$ (with 981 samples) continuous climatological features and a scalar target, which represents the state of vegetation of a basin of the Po river. This datasets have been composed by the authors merging different sources for the vegetation index, temperature, and precipitation over different basins (see (Didan 2015; Cornes et al. 2018; Zellner and Castelli 2022)), and they are available in the

[3] Considering Scikit-Learn (Pedregosa et al. 2011) implementation.
[4] Implementation available at https://xgboost.readthedocs.io/en/stable/index.html.
[5] Considering Scikit-Learn (Pedregosa et al. 2011) implementation.
[6] https://www.kaggle.com/datasets/kumarajarshi/life-expectancy-who
[7] https://www.kaggle.com/datasets/dgawlik/nyse

repository of this work. With the first climate dataset, considering linear regression, the reduction of features performed by the baselines and the LinCFA algorithm lead again to a significant improvement w.r.t. the full dataset, which has a satisfactory performance only considering XGBoost. The second climatic dataset significantly benefits from the reduction of the dimension in all cases. In particular, the LinCFA algorithm leads to the highest score in combination with linear regression.

Additionally, we further tested the *LinCFA* algorithm on four datasets from the UCI repository. In particular, we considered a simple classical dataset with 13 features (and 506 samples), the Boston Housing dataset (Harrison and Rubinfeld 1978), which confirms that, as discussed for the *life expectancy* dataset, the proposed algorithm does not lose information when the full regression is already able to manage the entire set of features. Then, a more complex dataset related to superconductivity (Hamidieh 2018), with 81 features, provides an example with many samples (21263), showing the possibility to apply the LinCFA algorithm also in this case.

Additionally, we considered the Cifar-10 dataset (Krizhevsky et al. 2009), transformed into a regression problem by considering each pixel of each of the three color layers as a feature and removing a pixel, considered as target. This provides a significant case with 6000 features and 3071 samples, where the LinCFA algorithm provides the best absolute score w.r.t. the full dataset and the considered linear and non-linear methods.

Finally, a *Gene Expression* (Fiorini 2016) dataset composed of 801 samples and 19133 features has been considered, where the gene expression of one gene is the target variable and the gene expression of the other genes available are considered as features. Similarly to the climate and the Cifar-10 dataset, with many highly correlated features and the need to reduce them to gain both interpretability and performance, this dataset allows to test the LinCFA algorithm on a dataset with a large number of features and a relatively small number of samples. The results show once again that the LinCFA algorihtms obtains high scores w.r.t. the other dimensionality reduction methods and the regression on the full dataset.

Tables 4 and 5 show the *MSE*, *RRMSE* and R^2 coefficients obtained with linear regression applied on the full dataset, on the dataset reduced by LinCFA, and on the dataset reduced by the best performing baseline. Additionally, the results related to Ridge and Lasso regression are reported. The extensive results for each dataset with all the baselines and regression methods considered can be found in Appendix E, Tabl es 9, 10, 11, 12, 13, 14, 15, 16. Additionally, in the appendix, Tables 17 and 18 report the results associated to the repetition of the experiments, imposing the same number of reduced features as the one identified by LinCFA to each dimensionality reduction method. Finally, an empirical example of computational time is reported in Table 19.

From the results, as already mentioned during the description of the datasets, it is possible to notice that, when the number of features is low, the results are similar between the full regression and the regression on the reduced dataset applying the baselines or *LinCFA*. On the other hand, when the algorithms are applied to the large dimensional data, the algorithm that we propose always obtains similar or better performances than the other methods. Therefore, the LinCFA Algorithm is able to reduce the dimensionality of the input features improving (or not-worsening) the performance of the linear model, preserving the interpretability of the reduced features.

Table 4 Experiments on four real datasets. The total number of samples n has been divided into train (66% of data) and test (33% of data) sets

Quantity	Life Exp	Financial	Climatological I	Climatological II
# samples n	1649	1299	1038	981
Full dim (# features D)	18	75	136	1991
Reduced dim. best baseline	16.6 ± 0.4	45.6 ± 1.8	47.6 ± 1.8	38.8 ± 5.1
Reduced dim. LinCFA (**ours**)	13.8 ± 1.7	14.6 ± 0.9	35.2 ± 3.9	37.0 ± 3.6
R^2 full	0.8309 ± 0.0031	-4.6094 ± 4.2851	0.2934 ± 0.0859	0.7529 ± 0.0230
R^2 Ridge	0.8302 ± 0.0030	0.8939 ± 0.0067	0.5559 ± 0.0347	0.7885 ± 0.0177
R^2 Lasso	0.7810 ± 0.0045	0.8671 ± 0.0051	0.5043 ± 0.0233	0.9032 ± 0.0046
R^2 best baseline	0.8313 ± 0.0034	$\mathbf{0.8972 \pm 0.0029}$	$\mathbf{0.6317 \pm 0.0201}$	0.8551 ± 0.0151
R^2 LinCFA (**ours**)	$\mathbf{0.8317 \pm 0.0027}$	0.8838 ± 0.0018	0.5727 ± 0.0435	$\mathbf{0.9203 \pm 0.0073}$
MSE full	0.1836 ± 0.0033	8.5071 ± 6.4986	0.2891 ± 0.0351	0.2341 ± 0.0209
MSE Ridge	0.1843 ± 0.0032	0.1690 ± 0.0102	0.1812 ± 0.0142	0.1926 ± 0.0162
MSE Lasso	0.2377 ± 0.0049	0.2016 ± 0.0077	0.2022 ± 0.0095	0.0882 ± 0.0042
MSE best baseline	0.1832 ± 0.0037	$\mathbf{0.1558 \pm 0.0045}$	$\mathbf{0.1503 \pm 0.0082}$	0.1318 ± 0.0138
MSE LinCFA (**ours**)	$\mathbf{0.1828 \pm 0.0029}$	0.1762 ± 0.0028	0.1743 ± 0.0178	$\mathbf{0.0725 \pm 0.0067}$
RRMSE full	0.4712 ± 0.0084	0.7050 ± 0.2453	0.7697 ± 0.0090	0.5865 ± 0.0308
RRMSE Ridge	0.4749 ± 0.0085	0.3383 ± 0.0093	0.4695 ± 0.0267	0.4556 ± 0.0311
RRMSE Lasso	0.4948 ± 0.0042	0.3926 ± 0.0057	0.5032 ± 0.0507	0.3095 ± 0.1291
RRMSE best baseline	0.5035 ± 0.0236	$\mathbf{0.3328 \pm 0.0105}$	$\mathbf{0.4229 \pm 0.0113}$	0.4946 ± 0.0635
RRMSE LinCFA (**ours**)	$\mathbf{0.4697 \pm 0.0074}$	0.3609 ± 0.0083	0.4514 ± 0.0352	$\mathbf{0.2999 \pm 0.0107}$

The best score for each dataset and metric is reported in bold

7 Conclusion and future work

This paper presents a dimensionality reduction algorithm in linear settings with the theoretical guarantee to produce a reduced dataset that does not perform worse than the full dataset in terms of *MSE*, with a decrease of variance that is larger than the increase of bias due to the aggregation of features. The main strength of the proposed approach is that it aggregates features through their mean, which reduces the dimension meanwhile preserving the interpretability of each feature, which is not common in traditional dimensionality reduction approaches like PCA. Moreover, the complexity of the proposed algorithm is lower than performing a linear regression on the full original dataset. The main weaknesses of the proposed method are that all the computations have been done assuming the features to be continuous and the relationship between the target and the features to be linear, which is a strong assumption in real-world applications. However, the empirical results show an increase in performance and a significant reduction of dimensionality when applying the proposed algorithm to real-world datasets. Therefore, as detailed in Remark 1, the algorithm is designed relying on the linear theoretical result, but it can be applied to any regression problem with continuous features, where the proposed threshold

Table 5 Experiments on four additional real datasets. The total number of samples n is divided into train (66% of data) and test (33% of data) sets

Quantity	Boston Housing	Superconductivity	Cifar-10	Gene expression
# samples n	506	21263	6000	801
Full dim (# features D)	13	81	3071	19133
Reduced dim. best baseline	11.4 ± 1.1	49.8 ± 0.4	76	35.2 ± 9.0
Reduced dim. LinCFA (**ours**)	7.2 ± 0.9	49.8 ± 2.9	75.6 ± 6.5	19.6 ± 1.7
R^2 full	0.7027 ± 0.0070	0.7290 ± 0.0013	0.1374 ± 0.6204	0.5166 ± 0.0041
R^2 Ridge	0.7027 ± 0.0069	0.7284 ± 0.0012	0.8575 ± 0.0996	0.5167 ± 0.0041
R^2 Lasso	0.6557 ± 0.0082	0.5844 ± 0.0032	0.9439 ± 0.0095	0.5839 ± 0.0095
R^2 best baseline	$\mathbf{0.7059 \pm 0.0121}$	$\mathbf{0.7470 \pm 0.0031}$	0.8236 ± 0.0192	0.5289 ± 0.0151
R^2 LinCFA (**ours**)	0.6541 ± 0.0206	0.6912 ± 0.0046	$\mathbf{0.9626 \pm 0.0260}$	$\mathbf{0.5990 \pm 0.0121}$
MSE full	0.2552 ± 0.0060	0.2674 ± 0.0013	0.5229 ± 0.3761	0.4992 ± 0.0041
MSE Ridge	0.2552 ± 0.0059	0.2680 ± 0.0012	0.0864 ± 0.0604	0.4991 ± 0.0042
MSE Lasso	0.2956 ± 0.0071	0.4102 ± 0.0031	0.0340 ± 0.0058	0.4340 ± 0.0088
MSE best baseline	$\mathbf{0.2524 \pm 0.0104}$	$\mathbf{0.2497 \pm 0.0031}$	0.1069 ± 0.0116	0.4806 ± 0.0156
MSE LinCFA (**ours**)	0.2970 ± 0.0176	0.3048 ± 0.0045	$\mathbf{0.0227 \pm 0.0157}$	$\mathbf{0.4141 \pm 0.0125}$
RRMSE full	0.5684 ± 0.0186	0.6025 ± 0.0030	0.4775 ± 0.2907	0.8282 ± 0.0089
RRMSE Ridge	0.5700 ± 0.0184	0.6044 ± 0.0031	0.2725 ± 0.1515	0.8279 ± 0.0086
RRMSE Lasso	0.6488 ± 0.0264	1.0354 ± 0.0683	0.1821 ± 0.0274	0.7880 ± 0.0174
RRMSE best baseline	$\mathbf{0.5665 \pm 0.0331}$	$\mathbf{0.5835 \pm 0.0066}$	0.4273 ± 0.0075	0.8473 ± 0.0379
RRMSE LinCFA (**ours**)	0.6530 ± 0.0143	0.6590 ± 0.0102	$\mathbf{0.1564 \pm 0.0790}$	$\mathbf{0.7492 \pm 0.0224}$

The best score for each dataset and metric is reported in bold

becomes a heuristic quantity. In this case, the empirical validation of the method on real-world datasets, which have no guarantee of linearity, shows a promising applicability of the proposed method outside linear contexts.

In future work it may be interesting to relax the linearity assumption in the theoretical analysis, considering the target as a general function of the input features and applying a general machine learning method to the data. Another possible way to enrich the results obtained in this paper is to consider structured data, where prior knowledge of their relationship can be useful to identify the most suitable features for the aggregation (e.g., on climatological data, features that are registered by two adjacent sensors are more likely to be aggregated).

Appendix A: Two-dimensional analysis: additional proofs and results

This section shows proofs and additional technical results that are not reported in Sect. 4 to keep the exposition clear.

A.1 Variance

This subsection contains some additional proofs related to the bivariate analysis of variance presented in the main paper.

Proof of Equation (3) Given the training set of features $\mathbf{X}$ and target $\mathbf{y}$, in a linear regression model, the estimated weights are computed as $\hat{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$. Therefore:

$$\begin{aligned} var_{\mathcal{T}}(\hat{w}|\mathbf{X}) &= var_{\mathcal{T}}((\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}|\mathbf{X}) \\ &= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1} var_{\mathcal{T}}(\mathbf{y}|\mathbf{X}). \end{aligned}$$

Since $(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{X} = \mathbf{I}$ and $var_{\mathcal{T}}(\mathbf{y}|\mathbf{X}) = \sigma^2$ by hypothesis, the result follows. □

Proof of Lemma 1 To prove this results it is enough to start from Eq. (3) and substitute the values of $\mathbf{X}$.

For the one-dimensional setting:

$$\begin{aligned} var_{\mathcal{T}}(\hat{w}|\mathbf{X}) &= (\mathbf{X}^T\mathbf{X})^{-1}\sigma^2 \\ &= \left(\begin{bmatrix} \bar{x}^1 & \dots & \bar{x}^n \end{bmatrix} \begin{bmatrix} \bar{x}^1 \\ \dots \\ \bar{x}^n \end{bmatrix} \right)^{-1} \sigma^2 = \frac{\sigma^2}{\sum_{i=1}^n (\bar{x}^i)^2}. \end{aligned}$$

Recalling that the expected value of the random variables x_1 and x_2 is zero by hypothesis, then $\sum_{i=1}^n (\bar{x}^i)^2 = (n-1)\hat{\sigma}^2_{\bar{x}}$.

For the two dimensional setting:

$$\begin{aligned} var_{\mathcal{T}}(\hat{w}|\mathbf{X}) &= (\mathbf{X}^T\mathbf{X})^{-1}\sigma^2 \\ &= \left(\begin{bmatrix} x_1^1 & \dots & x_1^n \\ x_2^1 & \dots & x_2^n \end{bmatrix} \begin{bmatrix} x_1^1 & x_2^1 \\ \dots & \dots \\ x_1^n & x_2^n \end{bmatrix} \right)^{-1} \sigma^2 \\ &= \left(\begin{bmatrix} (x_1^1)^2 + \dots + (x_1^n)^2 & x_1^1 x_2^1 + \dots + x_1^n x_2^n \\ x_1^1 x_2^1 + \dots + x_1^n x_2^n & (x_2^1)^2 + \dots + (x_2^n)^2 \end{bmatrix} \right)^{-1} \sigma^2 \\ &= \frac{\sigma^2}{(\sum_{i=1}^n (x_1^i)^2 \sum_{i=1}^n (x_2^i)^2) - (\sum_{i=1}^n (x_1^i x_2^i))^2} \\ &\quad \times \begin{bmatrix} (x_2^1)^2 + \dots + (x_2^n)^2 & -(x_1^1 x_2^1 + \dots + x_1^n x_2^n) \\ -(x_1^1 x_2^1 + \dots + x_1^n x_2^n) & (x_1^1)^2 + \dots + (x_1^n)^2 \end{bmatrix}. \end{aligned}$$

The result follows recalling again that the expected value of the random variables x_1, x_2 is zero, therefore $\sum_{i=1}^n (x_1^i)^2 = (n-1)\hat{\sigma}^2_{x_1}$, $\sum_{i=1}^n (x_2^i)^2 = (n-1)\hat{\sigma}^2_{x_2}$ and $\sum_{i=1}^n (x_1^i x_2^i) = (n-1)\hat{cov}(x_1, x_2)$. □

Proof of Theorem 1 Let us consider a training dataset $\mathcal{T}$ and a univariate test sample (x, y). Then the variance is:

$$\mathbb{E}_{x,\mathcal{T}}[(h_{\mathcal{T}}(x) - \bar{h}(x))^2] = \mathbb{E}_x \mathbb{E}_{\mathcal{T}}[(h_{\mathcal{T}}(x) - \bar{h}(x))^2].$$

Therefore, for the one-dimensional regression:

$$\begin{aligned}
&\mathbb{E}_x \mathbb{E}_{\mathcal{T}}[(h_{\mathcal{T}}(x) - \bar{h}(x))^2] = \mathbb{E}_x \mathbb{E}_{\mathcal{T}}[(\hat{w}x - \mathbb{E}_{\mathcal{T}}[\hat{w}x])^2] \\
&\quad = \mathbb{E}_x \mathbb{E}_{\mathcal{T}}[(x(\hat{w} - \mathbb{E}_{\mathcal{T}}[\hat{w}])^2] = \mathbb{E}_x[x^2] \mathbb{E}_{\mathcal{T}}[(\hat{w} - \mathbb{E}_{\mathcal{T}}[\hat{w}])^2] \\
&\quad = var_x(x) var_{\mathcal{T}}(\hat{w}) = \sigma_x^2 var_{\mathcal{T}}(\hat{w}).
\end{aligned}$$

Conditioning on the features training set $\mathbf{X}$:

$$\mathbb{E}_x \mathbb{E}_{\mathcal{T}}[(h_{\mathcal{T}}(x) - \bar{h}(x))^2 | \mathbf{X}] = \sigma_x^2 var_{\mathcal{T}}(\hat{w}|\mathbf{X}) = \frac{\sigma_x^2 \sigma^2}{(n-1)\hat{\sigma}_x^2}.$$

Regarding the two dimensional regression:

$$\begin{aligned}
&\mathbb{E}_x \mathbb{E}_{\mathcal{T}}[(h_{\mathcal{T}}(x) - \bar{h}(x))^2] \\
&\quad = \mathbb{E}_x \mathbb{E}_{\mathcal{T}}[(\hat{w}_1 x_1 + \hat{w}_2 x_2 - \mathbb{E}_{\mathcal{T}}[\hat{w}_1 x_1 + \hat{w}_2 x_2])^2] \\
&\quad = \mathbb{E}_x \mathbb{E}_{\mathcal{T}}[(x_1(\hat{w}_1 - \mathbb{E}_{\mathcal{T}}[\hat{w}_1]) + x_2(\hat{w}_2 - \mathbb{E}_{\mathcal{T}}[\hat{w}_2]))^2] \\
&\quad = \mathbb{E}_x \mathbb{E}_{\mathcal{T}}[(x_1(\hat{w}_1 - \mathbb{E}_{\mathcal{T}}[\hat{w}_1]))^2] + \mathbb{E}_x \mathbb{E}_{\mathcal{T}}[(x_2(\hat{w}_2 - \mathbb{E}_{\mathcal{T}}[\hat{w}_2]))^2] \\
&\qquad + 2\mathbb{E}_x \mathbb{E}_{\mathcal{T}}[x_1 x_2(\hat{w}_1 - \mathbb{E}_{\mathcal{T}}[\hat{w}_1])(\hat{w}_2 - \mathbb{E}_{\mathcal{T}}[\hat{w}_2])] \\
&\quad = var_x(x_1) var_{\mathcal{T}}(\hat{w}_1) + var_x(x_2) var_{\mathcal{T}}(\hat{w}_2) \\
&\qquad + 2cov_x(x_1, x_2) cov_{\mathcal{T}}(\hat{w}_1, \hat{w}_2).
\end{aligned}$$

Conditioning on the features training set:

$$\begin{aligned}
&\mathbb{E}_x \mathbb{E}_{\mathcal{T}}[(h_{\mathcal{T}}(x) - \bar{h}(x))^2 | \mathbf{X}] \\
&\quad = var_x(x_1) var_{\mathcal{T}}(\hat{w}_1 | \mathbf{X}) + var_x(x_2) var_{\mathcal{T}}(\hat{w}_2 | \mathbf{X}) \\
&\qquad + 2cov_x(x_1, x_2) cov_{\mathcal{T}}(\hat{w}_1, \hat{w}_2 | \mathbf{X}) \\
&\quad = \frac{\sigma^2(\sigma_{x_1}^2 \hat{\sigma}_{x_2}^2 + \sigma_{x_2}^2 \hat{\sigma}_{x_1}^2 - 2cov(x_1, x_2) \hat{cov}(x_1, x_2))}{(n-1)(\hat{\sigma}_{x_1}^2 \hat{\sigma}_{x_2}^2 - \hat{cov}(x_1, x_2)^2)}.
\end{aligned}$$

□

Proof of Theorem 3 From Theorem 1, the difference of variances between the two-dimensional and the one-dimensional cases is:

$$\begin{aligned}
&\frac{\sigma^2(\sigma_{x_1}^2 \hat{\sigma}_{x_2}^2 + \sigma_{x_2}^2 \hat{\sigma}_{x_1}^2 - 2cov(x_1, x_2) \hat{cov}(x_1, x_2))}{(n-1)(\hat{\sigma}_{x_1}^2 \hat{\sigma}_{x_2}^2 - \hat{cov}(x_1, x_2)^2)} \\
&\quad - \sigma_{x_1+x_2}^2 \frac{\sigma^2}{(n-1)\hat{\sigma}_{x_1+x_2}^2},
\end{aligned}$$

that with the assumptions of Eq. (10) can be written as:

$$\frac{\sigma^2(2\sigma_x^2\hat{\sigma}_x^2 - 2cov(x_1,x_2)\hat{cov}(x_1,x_2))}{(n-1)(\hat{\sigma}_x^4 - \hat{cov}(x_1,x_2)^2)} - \sigma^2_{x_1+x_2}\frac{\sigma^2}{(n-1)\hat{\sigma}^2_{x_1+x_2}}.$$

Recalling that $\sigma^2_{x_1+x_2} = \sigma^2_{x_1} + \sigma^2_{x_2} + 2cov(x_1,x_2)$, and that the same applies for the sample variance, the expression above is equal to:

$$\frac{\sigma^2(2\sigma_x^2\hat{\sigma}_x^2 - 2cov(x_1,x_2)\hat{cov}(x_1,x_2))}{(n-1)(\hat{\sigma}_x^4 - \hat{cov}(x_1,x_2)^2)} - (2\sigma_x^2 + 2cov(x_1,x_2))\frac{\sigma^2}{(n-1)(2\hat{\sigma}_x^2 + 2\hat{cov}(x_1,x_2))}.$$

Applying the common denominator the result follows:

$$\begin{aligned}
&\frac{\sigma^2}{(n-1)(\hat{\sigma}_x^4 - \hat{cov}(x_1,x_2)^2)} \\
&\quad \times [(2\sigma_x^2\hat{\sigma}_x^2 - 2cov(x_1,x_2)\hat{cov}(x_1,x_2)) \\
&\quad - (\sigma_x^2 + cov(x_1,x_2))(\hat{\sigma}_x^2 - \hat{cov}(x_1,x_2))] \\
&= \frac{\sigma^2(\sigma_x^2 - cov(x_1,x_2))(\hat{\sigma}_x^2 + \hat{cov}(x_1,x_2))}{(n-1)\hat{\sigma}_x^4(1-\hat{\rho}^2_{x_1,x_2})} \\
&= \frac{\sigma^2}{(n-1)} \cdot \frac{\sigma_x^2(1-\rho_{x_1,x_2})}{\hat{\sigma}_x^2(1-\hat{\rho}_{x_1,x_2})}.
\end{aligned}$$

□

A.2 Bias

This subsection contains some technical results and proofs used to compute the difference of biases between the two considered model in the bivariate linear setting of the main paper.

A.2.1 Expected value of the estimators

The expected value with respect to the training set $\mathcal{T}$ of the vector $\hat{w}$ of the regression coefficients estimates is necessary for the computations of the bias of the models. Given the training features $\mathbf{X}$, its known expression in a general problem $y = f(\mathbf{X}) + \epsilon$ is given by (Johnson and Wichern 2007):

$$\mathbb{E}_{\mathcal{T}}[\hat{w}|\mathbf{X}] = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T f(\mathbf{X}). \tag{A1}$$

Proof Given the training set of features $\mathbf{X}$ and target $\mathbf{y}$, in a linear regression model, the estimated weights are computed as $\hat{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$. Therefore:

$$\begin{aligned}\mathbb{E}_{\mathcal{T}}[\hat{w}|\mathbf{X}] &= \mathbb{E}_{\mathcal{T}}[(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}|\mathbf{X}] \\ &= \mathbb{E}_{\mathcal{T}}[(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T(f(\mathbf{X}) + \epsilon)|\mathbf{X}] \\ &= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\,\mathbb{E}_{\mathcal{T}}[f(\mathbf{X})|\mathbf{X}] + (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\,\mathbb{E}_{\mathcal{T}}[\epsilon|\mathbf{X}] \\ &= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T f(\mathbf{X}),\end{aligned}$$

where the last equality holds since the expected value of the noise term ϵ is null by hypothesis. □

The following lemma shows the expected value of the weights for the two models that we are considering.

Lemma 3 *Let the real model be linear with respect to the features x_1 and x_2 ($y = w_1x_1 + w_2x_2 + \epsilon$). Then, in the one-dimensional case $\hat{y} = \hat{w}\bar{x}$, we have*:

$$\mathbb{E}_{\mathcal{T}}[\hat{w}|\mathbf{X}] = \frac{2(w_1\hat{\sigma}^2_{x_1} + w_2\hat{\sigma}^2_{x_2} + (w_1 + w_2)c\hat{o}v(x_1, x_2))}{\hat{\sigma}^2_{x_1} + \hat{\sigma}^2_{x_2} + 2c\hat{o}v(x_1, x_2)}. \tag{A2}$$

In the two-dimensional case $\hat{y} = \hat{w}_1x_1 + \hat{w}_2x_2$ the estimators are unbiased:

$$\mathbb{E}_{\mathcal{T}}[\hat{w}|\mathbf{X}] = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}. \tag{A3}$$

Proof To prove this result it is enough to apply Equation (A1) in the two settings.

In the one dimensional case, with $x = \frac{x_1+x_2}{2}$, it becomes:

$$\begin{aligned}\mathbb{E}_{\mathcal{T}}[\hat{w}|\mathbf{X}] &= \frac{\sum_{i=1}^n x^i f(x)}{(n-1)\hat{\sigma}^2_x} \\ &= \frac{2(\sum_{i=1}^n x_1^i f(x_1^i, x_2^i) + \sum_{i=1}^n x_2^i f(x_1^i, x_2^i))}{(n-1)\hat{\sigma}^2_{x_1+x_2}}.\end{aligned}$$

Assuming the real model to be linear ($y = w_1x_1 + w_2x_2 + \epsilon$):

$$\begin{aligned}\mathbb{E}_{\mathcal{T}}[\hat{w}|\mathbf{X}] &= \frac{2(\sum_{i=1}^n x_1^i f(x_1^i, x_2^i) + \sum_{i=1}^n x_2^i f(x_1^i, x_2^i))}{(n-1)\hat{\sigma}^2_{x_1+x_2}} \\ &= \frac{2(\sum_{i=1}^n x_1^i (w_1x_1^i + w_2x_2^i) + \sum_{i=1}^n x_2^i (w_1x_1^i + w_2x_2^i))}{(n-1)\hat{\sigma}^2_{x_1+x_2}} \\ &= \frac{2(w_1\sum_{i=1}^n (x_1^i)^2 + (w_1 + w_2)\sum_{i=1}^n x_1^i x_2^i + w_2\sum_{i=1}^n (x_2^i)^2)}{(n-1)\hat{\sigma}^2_{x_1+x_2}}.\end{aligned}$$

Remembering that the expected values of x_1, x_2 are equal to zero it is possible to substitute the summations with sample variances and covariances, obtaining the result.

In the two dimensional setting, from the general result and substituting $f(\mathbf{X}) = \mathbf{X}w$:

$$\mathbb{E}_{\mathcal{T}}[\hat{w}|\mathbf{X}] = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T f(\mathbf{X}) = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{X}w = w.$$

□

A.2.2 Bias of the model

In Eq. (2), we defined the (squared) bias as follows:

$$\mathbb{E}_x[(\bar{h}(x) - \bar{y})^2] = \mathbb{E}_x[(\mathbb{E}_{\mathcal{T}}[h(x)] - \mathbb{E}_{y|x}[y])^2]. \tag{A4}$$

The following result shows the bias of the two specific models considered in this section.

Theorem 9 *Let the real model be linear with respect to the two features x_1, x_2 ($y = w_1x_1 + w_2x_2 + \epsilon$). Then, in the one dimensional case $y = \hat{w}\bar{x}$, we have*:

$$\begin{aligned}
&\mathbb{E}_x[(\bar{h}(x) - \bar{y})^2] \\
&= \frac{\sigma^2_{x_1+x_2}}{(\hat{\sigma}^2_{x_1+x_2})^2}(w_1\hat{\sigma}^2_{x_1} + w_2\hat{\sigma}^2_{x_2} + (w_1+w_2)\hat{cov}(x_1,x_2))^2 \\
&+ w_1^2\sigma^2_{x_1} + w_2^2\sigma^2_{x_2} + 2w_1w_2cov(x_1,x_2) \\
&- \frac{2}{\hat{\sigma}^2_{x_1+x_2}}(w_1\sigma^2_{x_1} + w_2\sigma^2_{x_2} + (w_1+w_2)cov(x_1,x_2)) \\
&\times (w_1\hat{\sigma}^2_{x_1} + w_2\hat{\sigma}^2_{x_2} + (w_1+w_2)\hat{cov}(x_1,x_2)).
\end{aligned} \tag{A5}$$

On the other hand, in the two dimensional case $y = \hat{w}_1x_1 + \hat{w}_2x_2$ the model is unbiased:

$$\mathbb{E}_x[(\bar{h}(x) - \bar{y})^2] = 0. \tag{A6}$$

Proof The proof combines the results of Lemma 3 with the definition of bias given in Eq. (A4).

Let us consider a training dataset $\mathcal{T}$ and a test sample x, y. Given the definition of (squared) bias:

$$\mathbb{E}_x[(\bar{h}(x) - \bar{y})^2] = \mathbb{E}_x[(\mathbb{E}_{\mathcal{T}}[h(x)] - \mathbb{E}_{y|x}[y])^2],$$

in the one dimensional case, considering $x = \frac{x_1+x_2}{2}$:

$$\begin{aligned}
&\mathbb{E}_x[(\mathbb{E}_{\mathcal{T}}[\hat{w}x] - (w_1x_1 + w_2x_2))^2] \\
&\quad = \mathbb{E}_x[(x\,\mathbb{E}_{\mathcal{T}}[\hat{w}] - (w_1x_1 + w_2x_2))^2] \\
&\quad = \mathbb{E}_x[x^2\,\mathbb{E}_{\mathcal{T}}[\hat{w}]^2] + \mathbb{E}_x[(w_1x_1 + w_2x_2)^2] \\
&\qquad - 2\,\mathbb{E}_x[\mathbb{E}_{\mathcal{T}}[\hat{w}]x(w_1x_1 + w_2x_2)].
\end{aligned}$$

Conditioning on the features training set and exploiting the independence between train and test set:

$$\begin{aligned}
&\mathbb{E}_x[(\bar{h}(x) - \bar{y})^2|\mathbf{X}] \\
&\quad = \sigma_x^2\,\mathbb{E}_{\mathcal{T}}[\hat{w}|\mathbf{X}]^2 + \mathbb{E}_x[(w_1x_1 + w_2x_2)^2] \\
&\qquad - 2\,\mathbb{E}_{\mathcal{T}}[\hat{w}|\mathbf{X}]\,\mathbb{E}_x[x(w_1x_1 + w_2x_2)].
\end{aligned}$$

That, substituting $x = \frac{x_1+x_2}{2}$, is equal to:

$$\begin{aligned}
&\frac{1}{4}(\sigma^2_{x_1} + \sigma^2_{x_2} + 2cov(x_1, x_2))\,\mathbb{E}_{\mathcal{T}}[\hat{w}|\mathbf{X}]^2 \\
&\quad + (w_1^2\sigma^2_{x_1} + w_2^2\sigma^2_{x_2} + 2w_1w_2cov(x_1, x_2)) \\
&\quad - \mathbb{E}_{\mathcal{T}}[\hat{w}|\mathbf{X}](w_1\sigma^2_{x_1} + w_2\sigma^2_{x_2} + (w_1 + w_2)cov(x_1, x_2)).
\end{aligned}$$

Substituting in the last equation the expression found in Lemma 3 for $\mathbb{E}_{\mathcal{T}}[\hat{w}|\mathbf{X}]$ in the one-dimensional setting, the result follows.

In the two dimensional regression, the bias is:

$$\mathbb{E}_x[(\bar{h}(x) - \bar{y})^2] = \mathbb{E}_x[(\mathbb{E}_{\mathcal{T}}[\hat{w_1}x_1 + \hat{w_2}x_2] - (w_1x_1 + w_2x_2))^2].$$

Conditioning on the features training set, exploiting the independence between train and test set and recalling $\mathbb{E}_{\mathcal{T}}[\hat{w}|\mathbf{X}] = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$:

$$\begin{aligned}
&\mathbb{E}_x[(x_1\,\mathbb{E}_{\mathcal{T}}[\hat{w_1}] + x_2\,\mathbb{E}_{\mathcal{T}}[\hat{w_2}] - (w_1x_1 + w_2x_2))^2] \\
&\quad = \mathbb{E}_x[(x_1w_1 + x_2w_2 - w_1x_1 - w_2x_2)^2] = 0.
\end{aligned}$$

□

A.2.3 Comparisons

The asymptotic and finite-samples results for the comparisons of biases can be found in the main paper, in this subsection the related proofs are shown.

Proof of Theorem 4 Considering the bias of the two models computed in Theorem 9 and exploiting consistency of the estimators, the difference between the one and the two dimensional model is equal to:

$$\begin{aligned}
&\frac{\sigma^2_{x_1+x_2}}{(\sigma^2_{x_1+x_2})^2}(w_1\sigma^2_{x_1} + w_2\sigma^2_{x_2} + (w_1+w_2)cov(x_1,x_2))^2 \\
&\quad + w_1^2\sigma^2_{x_1} + w_2^2\sigma^2_{x_2} + 2w_1w_2cov(x_1,x_2) \\
&\quad - \frac{2}{\sigma^2_{x_1+x_2}}(w_1\sigma^2_{x_1} + w_2\sigma^2_{x_2} + (w_1+w_2)cov(x_1,x_2)) \\
&\quad \times (w_1\sigma^2_{x_1} + w_2\sigma^2_{x_2} + (w_1+w_2)cov(x_1,x_2)) \\
&= \frac{(\sigma^2_{x_1}\sigma^2_{x_2} - cov(x_1,x_2)^2)(w_1-w_2)^2}{\sigma^2_{x_1+x_2}}.
\end{aligned}$$

The result follows by definition of covariance. □

Proof of Theorem 5 From Theorem 9, the difference between the one-dimensional and the two-dimensional bias is equal to the one-dimensional bias, since the two-dimensional model is unbiased. Therefore it is equal to:

$$\begin{aligned}
&\frac{\sigma^2_{x_1+x_2}(w_1\hat{\sigma}^2_{x_1} + w_2\hat{\sigma}^2_{x_2} + (w_1+w_2)\hat{cov}(x_1,x_2))^2}{(\hat{\sigma}^2_{x_1+x_2})^2} \\
&\quad + w_1^2\sigma^2_{x_1} + w_2^2\sigma^2_{x_2} + 2w_1w_2cov(x_1,x_2) \\
&\quad - \frac{2(w_1\sigma^2_{x_1} + w_2\sigma^2_{x_2} + (w_1+w_2)cov(x_1,x_2))}{\hat{\sigma}^2_{x_1+x_2}} \\
&\quad \times (w_1\hat{\sigma}^2_{x_1} + w_2\hat{\sigma}^2_{x_2} + (w_1+w_2)\hat{cov}(x_1,x_2)).
\end{aligned}$$

Substituting the assumptions from Eq. (10) we get:

$$\begin{aligned}
&\frac{2(\sigma^2_x + cov(x_1,x_2))((w_1+w_2)(\hat{cov}(x_1,x_2) + \hat{\sigma}^2_x))^2}{(\hat{\sigma}^2_{x_1+x_2})^2} \\
&\quad + \frac{(\hat{\sigma}^2_{x_1+x_2})^2(w_1^2\sigma^2_x + w_2^2\sigma^2_x + 2w_1w_2cov(x_1,x_2))}{(\hat{\sigma}^2_{x_1+x_2})^2} + \\
&\quad - \frac{2(\hat{\sigma}^2_{x_1+x_2})(w_1\sigma^2_x + w_2\sigma^2_x + (w_1+w_2)cov(x_1,x_2))}{(\hat{\sigma}^2_{x_1+x_2})^2} \\
&\quad \times (w_1\hat{\sigma}^2_x + w_2\hat{\sigma}^2_x + (w_1+w_2)\hat{cov}(x_1,x_2)),
\end{aligned}$$

from which, after basic algebraic computations, the result follows. □

Appendix B Two-dimensional analysis: additional setting

In the main paper, the finite-sample analysis in the two-dimensional case is performed with the assumption that the two features x_1, x_2 have respectively the same variance and sample variance. In this section are reported the results after the relaxation of this hypothesis.

In particular, the only assumption for this finite-sample analysis is unitary variance:

$$\sigma_{x_1} = \sigma_{x_2} = 1, \tag{B7}$$

implying by definition of covariance that $cov(x_1, x_2) = \rho_{x_1,x_2}$.

This is not an impacting restriction since it is always possible to scale a random variable to have unitary variance dividing it by its standard deviation.

The following theorem shows the difference of variance between the two-dimensional and the one-dimensional linear regression models.

Theorem 10 *In the finite-case with unitary variances, the difference of variances of the linear model with two features compared to the one with a single feature* (*which is their mean*), *is equal to*:

$$\begin{aligned}\frac{\sigma^2}{(n-1)} \times \Big[& \frac{(\hat{\sigma}^2_{x_1} - \hat{\sigma}^2_{x_2})^2}{\hat{\sigma}^2_{x_1}\hat{\sigma}^2_{x_2}(1-\hat{\rho}^2_{x_1,x_2})\hat{\sigma}^2_{x_1+x_2}} \\ & + \frac{2(1-\rho_{x_1,x_2})(\hat{\sigma}^2_{x_1} + c\hat{o}v(x_1,x_2))(\hat{\sigma}^2_{x_2} + c\hat{o}v(x_1,x_2))}{\hat{\sigma}^2_{x_1}\hat{\sigma}^2_{x_2}(1-\hat{\rho}^2_{x_1,x_2})\hat{\sigma}^2_{x_1+x_2}} \Big].\end{aligned} \tag{B8}$$

Proof Starting from the difference of variances between the two-dimensional and the one-dimensional model (Theorem 1):

$$\begin{aligned}& \frac{\sigma^2}{(n-1)(\hat{\sigma}^2_{x_1}\hat{\sigma}^2_{x_2} - c\hat{o}v(x_1,x_2)^2)} \\ & \quad \times (\sigma^2_{x_1}\hat{\sigma}^2_{x_2} + \sigma^2_{x_2}\hat{\sigma}^2_{x_1} - 2cov(x_1,x_2)c\hat{o}v(x_1,x_2)) \\ & \quad - \sigma^2_{x_1+x_2}\frac{\sigma^2}{(n-1)\hat{\sigma}^2_{x_1+x_2}},\end{aligned}$$

exploiting the unitary variance assumption becomes:

$$\begin{aligned}& \frac{\sigma^2}{(n-1)(\hat{\sigma}^2_{x_1}\hat{\sigma}^2_{x_2} - c\hat{o}v(x_1,x_2)^2)} \\ & \quad \times (\hat{\sigma}^2_{x_2} + \hat{\sigma}^2_{x_1} - 2cov(x_1,x_2)c\hat{o}v(x_1,x_2)) \\ & \quad - \frac{\sigma^2(2+2cov(x_1,x_2))}{(n-1)(\hat{\sigma}^2_{x_1} + \hat{\sigma}^2_{x_2} + 2c\hat{o}v(x_1,x_2))}.\end{aligned}$$

Applying the common denominator:

$$\begin{aligned}
&\frac{1}{(n-1)(\hat{\sigma}^2_{x_1}\hat{\sigma}^2_{x_2} - c\hat{o}v(x_1,x_2)^2)(\hat{\sigma}^2_{x_1} + \hat{\sigma}^2_{x_2} + 2c\hat{o}v(x_1,x_2))} \\
&\times \Big[\sigma^2(\hat{\sigma}^4_{x_1} + \hat{\sigma}^4_{x_2} + 2\hat{\sigma}^2_{x_1} c\hat{o}v(x_1,x_2) \\
&+ 2\hat{\sigma}^2_{x_2} c\hat{o}v(x_1,x_2) - 2\hat{\sigma}^2_{x_1} c\hat{o}v(x_1,x_2)cov(x_1,x_2)) \\
&+ \sigma^2(-2\hat{\sigma}^2_{x_2} c\hat{o}v(x_1,x_2)cov(x_1,x_2) \\
&- 2cov(x_1,x_2)c\hat{o}v(x_1,x_2)^2) \\
&+ \sigma^2(2c\hat{o}v(x_1,x_2)^2 - 2\hat{\sigma}^2_{x_1}\hat{\sigma}^2_{x_2} cov(x_1,x_2))\Big].
\end{aligned}$$

Finally, adding and subtracting on the numerator the term $2\hat{\sigma}^2_{x_1}\hat{\sigma}^2_{x_2}$ and grouping the terms, it is equal to:

$$\begin{aligned}
&\frac{\sigma^2}{(n-1)\hat{\sigma}^2_{x_1}\hat{\sigma}^2_{x_2}(1-\hat{\rho}^2_{x_1,x_2})\hat{\sigma}^2_{x_1+x_2}} \\
&\times \Big[(\hat{\sigma}^2_{x_1} - \hat{\sigma}^2_{x_2})^2 + 2(1-\rho_{x_1,x_2}) \\
&\times (\hat{\sigma}^2_{x_1} + c\hat{o}v(x_1,x_2))(\hat{\sigma}^2_{x_2} + c\hat{o}v(x_1,x_2))\Big].
\end{aligned}$$

□

Remark 14 When the number of samples n tends to infinity the result becomes the same found in the asymptotic analysis. Moreover, when the sample variances of the two features are equal, the result becomes the same of the finite case analysis with equal sample and real variances.

Lemma 4 *The quantity found as difference of variances between the two-dimensional and one-dimensional case in this general setting is always non-negative.*

Proof Recalling the result of Equation (B8), the first factor $\frac{\sigma^2}{(n-1)}$ and the denominator of the second one $\hat{\sigma}^2_{x_1}\hat{\sigma}^2_{x_2}(1-\hat{\rho}^2_{x_1,x_2})\hat{\sigma}^2_{x_1+x_2}$ are always non-negative, so the difference of features is positive if and only if the second numerator is positive:

$$\begin{aligned}
&(\hat{\sigma}^2_{x_1} - \hat{\sigma}^2_{x_2})^2 + 2(1-\rho_{x_1,x_2}) \\
&\quad\times (\hat{\sigma}^2_{x_1} + c\hat{o}v(x_1,x_2))(\hat{\sigma}^2_{x_2} + c\hat{o}v(x_1,x_2)) \geq 0.
\end{aligned}$$

Focusing on the term $2(1-\rho_{x_1,x_2})(\hat{\sigma}^2_{x_1}+\hat{cov}(x_1,x_2))(\hat{\sigma}^2_{x_2}+\hat{cov}(x_1,x_2))$, the function $(\hat{\sigma}^2_{x_1}+\hat{cov}(x_1,x_2))(\hat{\sigma}^2_{x_2}+\hat{cov}(x_1,x_2))$ takes minimum value when $\hat{cov}(x_1,x_2)=-\frac{\hat{\sigma}^2_{x_1}+\hat{\sigma}^2_{x_2}}{2}$, therefore the minimum value of this term is:

$$2(1-\rho_{x_1,x_2})\left(-\frac{1}{4}\right)(\hat{\sigma}^2_{x_1}-\hat{\sigma}^2_{x_2})^2.$$

Substituting it back in the original inequality:

$$\begin{aligned}&(\hat{\sigma}^2_{x_1}-\hat{\sigma}^2_{x_2})^2-\frac{1}{2}(1-\rho_{x_1,x_2})(\hat{\sigma}^2_{x_1}-\hat{\sigma}^2_{x_2})^2\\&=\frac{1}{2}(1+\rho_{x_1,x_2})(\hat{\sigma}^2_{x_1}-\hat{\sigma}^2_{x_2})^2,\end{aligned}$$

that is a quantity always non-negative and proves the lemma. □

The following theorem shows the difference of (squared) bias between the one-dimensional and the two-dimensional models.

Theorem 11 *In the finite-case, assuming unitary variances* $\sigma_{x_1}=\sigma_{x_2}=1$, *the increase of bias due to the aggregation of the two features with their average is equal to*:

$$\begin{aligned}&\frac{1}{(\hat{\sigma}^2_{x_1+x_2})^2}\\&\quad\times(2(1-\rho_{x_1,x_2})(\hat{\sigma}^2_{x_1}+\hat{\sigma}^2_{x_2}+\hat{cov}(x_1,x_2))\hat{cov}(x_1,x_2)\\&\quad+\hat{\sigma}^4_{x_1}+\hat{\sigma}^4_{x_2}-2\rho_{x_1,x_2}\hat{\sigma}^2_{x_1}\hat{\sigma}^2_{x_2})(w_1-w_2)^2.\end{aligned}\tag{B9}$$

Proof Recalling the expression of the difference of bias between the one-dimensional and the two-dimensional linear regression models (Theorem 9):

$$\begin{aligned}&\frac{\sigma^2_{x_1+x_2}}{(\hat{\sigma}^2_{x_1+x_2})^2}(w_1\hat{\sigma}^2_{x_1}+w_2\hat{\sigma}^2_{x_2}+(w_1+w_2)\hat{cov}(x_1,x_2))^2\\&\quad+w_1^2\sigma^2_{x_1}+w_2^2\sigma^2_{x_2}+2w_1w_2cov(x_1,x_2)\\&\quad-\frac{2}{\hat{\sigma}^2_{x_1+x_2}}\\&\quad\times(w_1\sigma^2_{x_1}+w_2\sigma^2_{x_2}+(w_1+w_2)cov(x_1,x_2))\\&\quad\times(w_1\hat{\sigma}^2_{x_1}+w_2\hat{\sigma}^2_{x_2}+(w_1+w_2)\hat{cov}(x_1,x_2)),\end{aligned}$$

exploiting the unitary variance assumption can be written as:

$$\begin{aligned}
&\frac{2(1+\rho_{x_1,x_2})}{(\hat{\sigma}^2_{x_1+x_2})^2}(w_1\hat{\sigma}^2_{x_1}+w_2\hat{\sigma}^2_{x_2}+(w_1+w_2)\hat{cov}(x_1,x_2))^2\\
&\quad+\frac{(w_1^2+w_2^2+2w_1w_2\rho_{x_1,x_2})(\hat{\sigma}^2_{x_1}+\hat{\sigma}^2_{x_2}+2\hat{cov}(x_1,x_2))^2}{(\hat{\sigma}^2_{x_1+x_2})^2}\\
&\quad-\frac{2}{(\hat{\sigma}^2_{x_1+x_2})^2}(w_1+w_2)(1+\rho_{x_1,x_2})\\
&\quad\times(w_1\hat{\sigma}^2_{x_1}+w_2\hat{\sigma}^2_{x_2}+(w_1+w_2)\hat{cov}(x_1,x_2))\\
&\quad\times(\hat{\sigma}^2_{x_1}+\hat{\sigma}^2_{x_2}+2\hat{cov}(x_1,x_2)).
\end{aligned}$$

After basic algebraic computations the result follows. □

Remark 15 When the number of samples n tends to infinity the result becomes the same found in the asymptotic analysis. Moreover, when the sample variances of the two features are equal, the result becomes the same of the finite case analysis with equal sample and real variances.

Theorem 12 *Necessary and sufficient condition for positivity of the difference between the reduction of variance and the increase of bias when aggregating two features with their average in the unitary-variance finite-sample setting is*:

$$\begin{aligned}
&\sigma^2[(\hat{\sigma}^2_{x_1}-\hat{\sigma}^2_{x_2})^2\\
&\quad+2(1-\rho_{x_1,x_2})(\hat{\sigma}^2_{x_1}+\hat{cov}(x_1,x_2))(\hat{\sigma}^2_{x_2}+\hat{cov}(x_1,x_2))]\\
&\quad\times(\hat{\sigma}^2_{x_1}+\hat{\sigma}^2_{x_2}+2\hat{cov}(x_1,x_2))\\
&\quad-[2(1-\rho_{x_1,x_2})(\hat{\sigma}^2_{x_1}+\hat{\sigma}^2_{x_2}+\hat{cov}(x_1,x_2))\hat{cov}(x_1,x_2)\\
&\quad+\hat{\sigma}^4_{x_1}+\hat{\sigma}^4_{x_2}-2\rho\hat{\sigma}^2_{x_1}\hat{\sigma}^2_{x_2}]\\
&\quad\times(w_1-w_2)^2(n-1)(\hat{\sigma}^2_{x_1}\hat{\sigma}^2_{x_2}-\hat{cov}(x_1,x_2)^2)\geq 0.
\end{aligned}\tag{B10}$$

Proof The result is obtained subtracting the results of the two previous theorems, after algebraic computations. □

Appendix C Two-dimensional analysis: theoretical and practical quantities

This section elaborates the inequalities found in the main paper in Theorem 6, 7 considering only theoretical quantities or, on the other hand, quantities that can all be computed from data. In this way, in the bivariate case, we have both a theoretical conclusion of the analysis and an empirical one that can be used in practice.

For the asymptotic analysis it is straightforward to obtain a theoretical and an empirical expression, indeed at the limit the estimators converge in probability to the theoretical quantities.

Theorem 13 *In the asymptotic setting of Theorem 6, considering only theoretical quantities, the following inequalities hold*:

$$\begin{cases} \rho^2_{x_1,x_2} \geq 1 - \frac{\sigma^2 \sigma^2_{x_1+x_2}}{(n-1)\sigma^2_{x_1}\sigma^2_{x_2}(w_1-w_2)^2} \\ \rho_{x_1,x_2} \geq 1 - \frac{2\sigma^2}{(n-1)(w_1-w_2)^2} \ (\textit{if } \sigma_{x_1} = \sigma_{x_2} = 1). \end{cases} \tag{C11}$$

On the other hand, considering only quantities that can be derived from data:

$$\begin{cases} \hat{\rho}^2_{x_1,x_2} \geq 1 - \frac{s^2 \hat{\sigma}^2_{x_1+x_2}}{(n-1)\hat{\sigma}^2_{x_1}\hat{\sigma}^2_{x_2}(\hat{w}_1-\hat{w}_2)^2} \\ \hat{\rho}_{x_1,x_2} \geq 1 - \frac{2s^2}{(n-1)(\hat{w}_1-\hat{w}_2)^2} \ (\textit{if } \hat{\sigma}_{x_1} = \hat{\sigma}_{x_2} = 1), \end{cases} \tag{C12}$$

where $s^2 = \frac{\hat{\epsilon}^T \hat{\epsilon}}{n-3}$ *is the unbiased estimator of the variance* σ^2 *of the residual* ϵ *of the linear regression* (*an estimate* $\hat{\epsilon}$ *of the residual can be computed subtracting the predicted value to the real value of the target*).

Proof Equation (C11) is the same result of Theorem 6.

To derive Eq. (C12) it is sufficient to substitute the theoretical quantities with their consistent estimators. □

For the finite-samples analysis it is necessary to introduce confidence intervals to substitute theoretical with empirical quantities and viceversa.

Theorem 14 *In the finite-case setting of Theorem 7, considering only empirical quantities, the following inequality holds with probability at least* $1 - \delta$:

$$\begin{aligned} \hat{\rho}_{x_1,x_2} \geq 1 &- \frac{2(n-3)s^2}{(n-1)\chi^2_{n-3}\left(\frac{\delta}{2}\right)\hat{\sigma}^2_x} \\ &\times \frac{1}{(|\hat{w}_1 - \hat{w}_2| + \sqrt{3F_{3,n-3}\left(\frac{\delta}{2}\right)}(\sqrt{\hat{var}(\hat{w}_1)} + \sqrt{\hat{var}(\hat{w}_2)}))^2}, \end{aligned} \tag{C13}$$

where $\chi^2_{n-3}(\cdot)$ *represents a Chi-squared distribution with* $n-3$ *degrees of freedom and* $F_{3,n-3}(\cdot)$ *a Fisher distribution with* $3, n-3$ *degrees of freedom.*

Proof The unilateral confidence interval for the variance σ^2 of the residual ϵ of the linear regression model $y = w_1 x_1 + w_2 x_2 + \epsilon$, assuming $\epsilon \sim \mathcal{N}(0, \sigma^2)$ is, with probability $1 - \alpha$ (Johnson and Wichern 2007):

$$\frac{(n-r-1)s^2}{\chi^2_{n-r-1}(\alpha)} \leq \sigma^2.$$

The simultaneous confidence interval for the weights w_1, w_2 of the linear regression model $y = w_1 x_1 + w_2 x_2 + \epsilon$ is, with probability $1-\gamma$:

$$\begin{cases} w_1 \in [\hat{w_1} \pm \sqrt{\hat{var}(\hat{w}_1)}\sqrt{(r+1)F_{r+1,n-r-1}(\gamma)}] \\ w_2 \in [\hat{w_2} \pm \sqrt{\hat{var}(\hat{w}_2)}\sqrt{(r+1)F_{r+1,n-r-1}(\gamma)}]. \end{cases}$$

Considering the confidence intervals and the inequality of Theorem 7, with probability $1-\delta$:

$$\begin{aligned} \hat{\rho}_{x_1,x_2} &\geq 1 - \frac{2(n-3)s^2}{(n-1)\chi^2_{n-3}\left(\frac{\delta}{2}\right)\hat{\sigma}_x^2} \\ &\quad \times \frac{1}{(|\hat{w}_1 - \hat{w}_2| + \sqrt{3F_{3,n-3}\left(\frac{\delta}{2}\right)}(\sqrt{\hat{var}(\hat{w}_1)} + \sqrt{\hat{var}(\hat{w}_2)}))^2} \\ &\geq 1 - \frac{2\sigma^2}{(n-1)\hat{\sigma}_x^2(w_1 - w_2)^2}. \end{aligned}$$

This means that the inequality holds and concludes the proof. □

Remark 16 At the limit the quantity $\frac{\chi^2}{n-3}$ tends to 1, so the result of Theorem 14 is coherent with the asymptotic result.

In order to obtain the result with only theoretical quantities in the finite-sample case, it is necessary to introduce two bounds on the difference between covariance and sample covariance.

Proposition 15 *The following inequalities hold.*

- With probability $1-\delta$:

$$\hat{cov}(x_1, x_2) - cov(x_1, x_2) \leq 3\sqrt{\frac{log\left(\frac{4}{\delta}\right)}{n-1}}. \tag{C14}$$

- with probability $1-\delta$:

$$cov(x_1, x_2) - \hat{cov}(x_1, x_2) \leq 4\sqrt{\frac{log\left(\frac{4}{\delta}\right)}{n-1}}; \tag{C15}$$

Proof The proof exploits Hoeffding's inequality by applying it to the random variable $Z = x_1 x_2$. The proof will derive the results of the proposition for two general random variables X, Y and n data x_i, y_i sampled from their distribution. We denote with $\bar{X}, \bar{Y}$ the means of the considered samples.

From Hoeffding's inequality (Hoeffding 1963) applied to the variable $Z = XY$, with probability $1 - \delta$, we get:

$$|\,\mathbb{E}\,[XY] - \frac{1}{n}\sum_{i=1}^{n} x_i y_i| \leq \sqrt{\frac{log\left(\frac{2}{\delta}\right)}{2n}},$$

which implies:

$$|\,\mathbb{E}\,[XY] - \frac{1}{n-1}\sum_{i=1}^{n} x_i y_i| \leq \sqrt{\frac{log\left(\frac{2}{\delta}\right)}{n-1}}.$$

Then with probability $1 - 2\delta$:

$$\begin{aligned}
&\hat{cov}(X,Y) - cov(X,Y) \\
&= \frac{1}{n-1}\sum_{i=1}^{n} x_i y_i - \frac{n}{n-1}\bar{X}\bar{Y} - \mathbb{E}\,[XY] + \mathbb{E}\,[X]\,\mathbb{E}\,[Y] \\
&\leq \frac{1}{n-1}\sum_{i=1}^{n} x_i y_i - \bar{X}\bar{Y} - \mathbb{E}\,[XY] + \mathbb{E}\,[X]\,\mathbb{E}\,[Y] \pm \bar{X}\,\mathbb{E}\,[Y] \\
&\leq \sqrt{\frac{log\left(\frac{2}{\delta}\right)}{n-1}} + \bar{Y}\sqrt{\frac{log\left(\frac{2}{\delta}\right)}{2n}} + \mathbb{E}\,[Y]\sqrt{\frac{log\left(\frac{2}{\delta}\right)}{2n}} \leq 3\sqrt{\frac{log\left(\frac{2}{\delta}\right)}{n-1}},
\end{aligned}$$

where the second inequality applies Hoeffding's inequality three times. Equation (C14) is therefore proved.

On the other hand, with probability $1 - 2\delta$:

Table 6 95% confidence intervals for bivariate synthetic experiments with large difference of weights ($w_1 = 0.2, w_2 = 0.8$)

Quantity	Confidence interval for different standard deviations of the noise		
	$\sigma = 0.5$ ($\sigma^2 = 0.25$)	$\sigma = 1$ ($\sigma^2 = 1$)	$\sigma = 10$ ($\sigma^2 = 100$)
Theoretical $\bar{\rho}$	0.997217	0.988867	−0.113338
Empirical $\bar{\rho}$ (median)	0.997218	0.988807	0.819620
$\hat{\rho}_{x_1,x_2}$	0.919045 ± 1.78e−4	0.918558 ± 1.66e−4	0.918901 ± 1.7e−4
$\hat{\sigma}^2$	0.250211 ± 5.31e−4	0.999966 ± 2.29e−4	100.354737 ± 0.230573
$\hat{w}_1$	0.198552 ± 2.094e−3	0.200106 ± 4.124e−3	0.198441 ± 0.041063
$\hat{w}_2$	0.800265 ± 2.084e−3	0.800015 ± 4.151e−3	0.783078 ± 0.040011
$\hat{w}$	0.998817 ± 8.33e−4	1.000121 ± 1.678e−3	0.981519 ± 0.015914
R^2 full	0.781094 ± 5.2e−5	0.487304 ± 1.23e−4	0.010205 ± 2.64e−4
R^2 aggr	0.764944 ± 3e−5	0.485527 ± 7.8e−5	0.010630 ± 1.69e−4
MSE full	0.275187 ± 6.5e−5	1.000209 ± 2.40e−4	103.020861 ± 0.027468
MSE aggr	0.295489 ± 3.7e−5	1.003677 ± 1.52e−4	102.976615 ± 0.017587
Var full	0.001507 ± 2.23e−4	0.006333 ± 9.02e−4	0.590507 ± 0.082236
Var aggr	0.001005 ± 1.50e−4	0.004108 ± 5.66e−4	0.383723 ± 0.056062
Bias full	0.273680 ± 3.389e−3	0.993876 ± 1.281e−3	102.430354 ± 13.192218
Bias aggr	0.294484 ± 3.773e−3	0.999569 ± 1.270e−3	102.592893 ± 13.165301
Aggregations (theo)	**0**	**0**	**500**
Aggregations (emp)	**0**	**24**	**332**

Table 7 95% confidence intervals for bivariate synthetic experiments with small difference of weights ($w_1 = 0.47, w_2 = 0.52$)

Quantity	Confidence interval for different standard deviations of the noise		
	$\sigma = 0.5$ ($\sigma^2 = 0.25$)	$\sigma = 1$ ($\sigma^2 = 1$)	$\sigma = 10$ ($\sigma^2 = 100$)
Theoretical $\bar{\rho}$	0.599198	−0.603206	−25.675559
Empirical $\bar{\rho}$ (median)	0.866424	0.813952	0.802706
$\hat{\rho}_{x_1,x_2}$	0.919192 ± 1.78e−4	0.918774 ± 1.81e−4	0.919326 ± 1.72e−4
$\hat{\sigma}^2$	0.250829 ± 5.65e−4	1.003768 ± 2.276e−4	99.955617 ± 0.225969
$\hat{w}_1$	0.467944 ± 2.064e−3	0.466161 ± 3.907e−3	0.370056 ± 0.039486
$\hat{w}_2$	0.522283 ± 2.079e−3	0.526410 ± 3.959e−3	0.634829 ± 0.039561
$\hat{w}$	0.990227 ± 8.15e−4	0.992571 ± 1.646e−3	1.004884 ± 1.634e−3
R^2 full	0.793788 ± 6.4e−5	0.505334 ± 9.1e−5	0.012507 ± 2.31e−4
R^2 aggr	0.794190 ± 6.0e−5	0.506237 ± 7.5e−5	0.014699 ± 2.0e−5
MSE full	0.262609 ± 8.2e−5	0.948725 ± 1.75e−4	94.572608 ± 2.209e−3
MSE aggr	0.262098 ± 7.7e−5	0.946993 ± 1.45e−4	94.362608 ± 1.914e−3
Var full	0.001524 ± 2.16e−4	0.005719 ± 8.22e−4	0.571782 ± 0.080326
Var aggr	0.000998 ± 1.44e−4	0.003997 ± 5.76e−4	0.371541 ± 0.054317
Bias full	0.261085 ± 0.034614	0.943006 ± 0.127423	94.000826 ± 11.836568
Bias aggr	0.261105 ± 0.034928	0.943996 ± 0.127362	93.991098 ± 11.826050
Aggregations (theo)	**500**	**500**	**500**
Aggregations (emp)	**314**	**339**	**346**

$$
\begin{aligned}
&cov(X,Y) - c\hat{o}v(X,Y) \\
&= -\frac{1}{n-1}\sum_{i=1}^{n} x_i y_i + \frac{n}{n-1}\bar{X}\bar{Y} + \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y] \\
&= -\frac{1}{n-1}\sum_{i=1}^{n} x_i y_i + \bar{X}\bar{Y} + \frac{1}{n-1}\bar{X}\bar{Y} \\
&\quad + \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y] \pm \bar{X}\mathbb{E}[Y] \\
&\leq \sqrt{\frac{log\left(\frac{2}{\delta}\right)}{n-1}} + \bar{X}\sqrt{\frac{log\left(\frac{2}{\delta}\right)}{2n}} + \mathbb{E}[Y]\sqrt{\frac{log\left(\frac{2}{\delta}\right)}{2n}} + \frac{1}{n-1}\bar{X}\bar{Y} \\
&\leq 4\sqrt{\frac{log\left(\frac{2}{\delta}\right)}{n-1}},
\end{aligned}
$$

where the first inequality is again due to the application of Hoeffding's inequality three times. From this result, Eq. C15 follows. □

It is now possible to derive the expression of Eq. (18) with only theoretical quantities.

Theorem 16 *In the finite-case setting of Theorem 7, considering only theoretical quantities, the following inequality holds with probability at least* $1-\delta$:

$$
\begin{aligned}
\rho_{x_1,x_2} \geq 1 &- \frac{2\sigma^2}{(n-1)\sigma_x^2(w_1-w_2)^2} \\
&+ \frac{1}{\sigma_x^2}\left(\frac{2log\left(\frac{2}{\delta}\right)}{n-1} + 2\sigma_x\sqrt{\frac{2log\left(\frac{2}{\delta}\right)}{n-1}} + 4\sqrt{\frac{log\left(\frac{8}{\delta}\right)}{n-1}}\right).
\end{aligned}
\tag{C16}
$$

Proof To prove the theorem it is enough to apply the upper bound for the sample variance from (Maurer and Pontil 2009) and the lower bound for the sample covariance from the inequalities of Proposition 15.

Regarding the sample variance, with probability $1-\alpha$ it holds (Maurer and Pontil 2009):

$$
\hat{\sigma}_x^2 \leq \left(\sigma_x + \sqrt{\frac{2log\left(\frac{1}{\alpha}\right)}{n-1}}\right)^2.
$$

For the sample covariance, Proposition 15 shows that with probability $1-\gamma$:

$$
c\hat{o}v(x_1,x_2) \geq cov(x_1,x_2) - 4\sqrt{\frac{log\left(\frac{4}{\gamma}\right)}{n-1}}.
$$

Table 8 Detailed synthetic experiment in the three dimensional setting

Quantity	95% Confidence interval
Theoretical $\bar{\rho}$ (lower,upper)	0.826063, 0.932936
Empirical $\bar{\rho}$ (median)	0.831487, 0.933358
$\hat{\rho}_{x_1,x_2}$	$0.880300 \pm 1.07e{-4}$
$\hat{\sigma}^2$	$0.249982 \pm 2.29e{-4}$
$\hat{w}_1$	$0.399401 \pm 9.12e{-4}$
$\hat{w}_2$	$0.600999 \pm 6.72e{-4}$
$\hat{w}$	$1.179325 \pm 3.41e{-4}$
R^2 full	$0.825028 \pm 6e{-6}$
R^2 aggr	$0.825319 \pm 5e{-6}$
MSE full	$0.285611 \pm 9e{-6}$
MSE aggr	$0.285137 \pm 8e{-6}$
Var full	0.001526 ± 0.001557
Var aggr	0.000976 ± 0.003587
Bias full	0.284086 ± 0.046098
Bias aggr	0.284161 ± 0.045194
Aggregations (theo)	**500**
Aggregations (emp)	**335**

Starting from the inequality of Theorem 7:

$$\hat{\rho}_{x_1,x_2} \geq 1 - \frac{2\sigma^2}{(n-1)\hat{\sigma}_x^2(w_1 - w_2)^2},$$

it is equal to:

$$\hat{\sigma}_x^2 - c\hat{o}v(x_1, x_2) \leq \frac{2\sigma^2}{(n-1)(w_1 - w_2)^2}.$$

Therefore, with probability $1 - \delta$:

$$\begin{aligned}
&\hat{\sigma}_x^2 - c\hat{o}v(x_1, x_2) \\
&\quad \leq \left(\sigma_x + \sqrt{\frac{2log\left(\frac{2}{\delta}\right)}{n-1}}\right)^2 - \left(cov(x_1, x_2) - 4\sqrt{\frac{log\left(\frac{8}{\delta}\right)}{n-1}}\right) \\
&\quad \leq \frac{2\sigma^2}{(n-1)(w_1 - w_2)^2}.
\end{aligned}$$

After basic algebraic computations, the result follows. □

Remark 17 The empirical result of Theorem 14 depends on the distribution of the residual, assuming it to be Gaussian. On the other hand, the theoretical expression of Theorem 16 does not need any assumption on the distribution.

Appendix D three-dimensional algorithm

This section contains detailed results and proofs related to Sect. 5 of the main paper.

D.1 Variance

The following theorem shows the asymptotic difference of variance between the two considered linear regression models.

Theorem 17 *Let* $\mathbf{X} = [\mathbf{x_1}\ \mathbf{x_2}\ \mathbf{x_3}]$, $\hat{\mathbf{X}} = [\mathbf{x_1}\ \mathbf{x_2}]$ *and* $\bar{\mathbf{X}} = [\bar{\mathbf{x}}]$, *with* $\bar{x} = \frac{x_1+x_2}{2}$. *Then, for the one-dimensional linear regression* $\hat{y} = \hat{w}\frac{x_1+x_2}{2}$, *we have*:

$$var_{\mathcal{T}}(\hat{w}|\mathbf{X}) = (\bar{\mathbf{X}}^T\bar{\mathbf{X}})^{-1}\sigma^2, \tag{D17}$$

and for the two-dimensional linear regression $\hat{y} = \hat{w}_1 x_1 + \hat{w}_2 x_2$, *we have*:

$$var_{\mathcal{T}}(\hat{w}|\mathbf{X}) = (\hat{\mathbf{X}}^T\hat{\mathbf{X}})^{-1}\sigma^2. \tag{D18}$$

Proof The results follow from the general expression of variance of the estimators from Equation (3) and substituting respectively $\bar{\mathbf{X}}$ and $\hat{\mathbf{X}}$ for the two considered models. □

Remark 18 Since the linear regression models are the same of the bivariate case, starting from the result of Theorem 17, the variance of the estimators in the two cases remains the same of Lemma 1 and the asymptotic difference of variances remains the one of Theorem 2 ($\Delta_{var}^{n\to\infty} = \frac{\sigma^2}{(n-1)}$).

D.2 Bias

This subsection introduces the asymptotic difference of bias of the two considered linear regression models in the three dimensional setting.

As in the bivariate setting, the first step is to calculate the bias for each of the two considered models. In the asymptotic case, assuming unitary variances of the features $\sigma_{x_1} = \sigma_{x_2} = \sigma_{x_3} = 1$, for the one-dimensional regression $\hat{y} = \hat{w}\frac{x_1+x_2}{2}$ and for the two-dimensional regression $\hat{y} = \hat{w_1}x_1 + \hat{w_2}x_2$, the (squared) bias $\mathbb{E}_x[(\bar{h}(x) - \bar{y})^2]$ can be expressed with two functions, that will be denoted respectively with $\mathcal{F}(\cdot)$ and $\mathcal{G}(\cdot)$. These functions depend on the three features, their coefficients and their correlations.

For the extension to the D-dimensional case, it will be necessary to keep the feature x_3 having general variance $\sigma^2_{x_3}$. With little changes in the algebraic computations of the proof, the bias of the two models can be easily extended (see Appendix D for the details).

From the results obtained, it is possible to compute the increase of bias due to the aggregation of the two variables x_1, x_2 with their average $\bar{x} = \frac{x_1+x_2}{2}$.

Theorem 18 *In the asymptotic setting, let the relationship between the features and the target be linear with Gaussian noise. Assuming unitary variances of the features* $\sigma_{x_1} = \sigma_{x_2} = \sigma_{x_3} = 1$, *the increase of bias due to the aggregation of the features* x_1 *and* x_2 *with their average is given by*:

$$\begin{aligned}\Delta^{n\to\infty}_{Bias} = &\frac{1}{2}(1-\rho_{x_1,x_2})(w_1 - w_2)^2 \\ &+ (w_1 w_3 - w_2 w_3)(\rho_{x_1,x_3} - \rho_{x_2,x_3}) \\ &+ w_3^2 \frac{(\rho_{x_1,x_3} - \rho_{x_2,x_3})^2}{2(1-\rho_{x_1,x_2})}.\end{aligned} \tag{D19}$$

Proof The result follows from the difference of the biases of the two models, after algebraic computations. □

Remark 19 Letting the feature x_3 having general variance $\sigma^2_{x_3}$, with little changes in the algebraic computations of the proof, the difference of biases is given by:

Table 9 Extended result of experiments on *Life Expectancy* dataset. The total number of samples n has been divided into train (66% of data) and test (33% of data) sets

	Life expectancy			
	# samples $n = 1649$		# features $D = 18$	
Linear regression	Reduced dim	R^2	*MSE*	*RRMSE*
Full	18	0.8309 ± 0.0031	0.1836 ± 0.0033	0.4712 ± 0.0084
PCA	12.8 ± 0.3	0.8284 ± 0.0014	0.1863 ± 0.0004	0.4837 ± 0.0087
Supervised PCA	11.6 ± 4.4	0.8121 ± 0.0141	0.2041 ± 0.0153	0.5128 ± 0.0278
Kernel PCA	13.0 ± 0.6	0.8215 ± 0.0045	0.1938 ± 0.0049	0.4805 ± 0.0081
LLE	17.0 ± 0.5	0.6429 ± 0.0265	0.3876 ± 0.0287	0.8427 ± 0.0587
LPP	16.4 ± 0.7	0.8145 ± 0.0074	0.2013 ± 0.0081	0.4986 ± 0.0116
Isomap	15.4 ± 1.6	0.7856 ± 0.0059	0.2328 ± 0.0064	0.5621 ± 0.0208
RReliefF	16.6 ± 0.4	0.8313 ± 0.0034	0.1832 ± 0.0037	0.5035 ± 0.0236
LinCFA	13.8 ± 1.7	$\mathbf{0.8317 \pm 0.0027}$	$\mathbf{0.1828 \pm 0.0029}$	$\mathbf{0.4697 \pm 0.0074}$
SVM for regression	Reduced dim	R^2	*MSE*	*RSE*
Full	18	0.8980 ± 0.0046	0.1108 ± 0.0050	0.3551 ± 0.0097
PCA	12.8 ± 0.3	0.8911 ± 0.0048	0.1182 ± 0.0052	0.3657 ± 0.0088
Supervised PCA	11.6 ± 4.4	0.8772 ± 0.0093	0.1332 ± 0.0102	0.3859 ± 0.0152
Kernel PCA	13.0 ± 0.6	0.9046 ± 0.0067	0.1036 ± 0.0072	0.3415 ± 0.0131
LLE	17.0 ± 0.5	0.8039 ± 0.0186	0.2129 ± 0.0201	0.5304 ± 0.0339
LPP	16.4 ± 0.7	0.8622 ± 0.0048	0.1495 ± 0.0053	0.4285 ± 0.0071
Isomap	15.4 ± 1.6	0.8568 ± 0.0079	0.1555 ± 0.0086	0.4348 ± 0.0169
RReliefF	16.6 ± 0.4	0.8935 ± 0.0081	0.1156 ± 0.0087	0.3631 ± 0.0149
LinCFA	13.8 ± 1.7	$\mathbf{0.9097 \pm 0.0059}$	$\mathbf{0.0981 \pm 0.0064}$	$\mathbf{0.3273 \pm 0.0117}$
XGBoost	Reduced dim	R^2	*MSE*	*RSE*
Full	18	0.9238 ± 0.0039	0.0828 ± 0.0042	0.2905 ± 0.0067
PCA	12.8 ± 0.3	0.8754 ± 0.0039	0.1353 ± 0.0042	0.3807 ± 0.0076
Supervised PCA	11.6 ± 4.4	0.8732 ± 0.0132	0.1387 ± 0.0143	0.3810 ± 0.0181
Kernel PCA	13.0 ± 0.6	0.8746 ± 0.0438	0.1361 ± 0.0078	0.3821 ± 0.0326
LLE	17.0 ± 0.5	0.7915 ± 0.0201	0.2264 ± 0.0217	0.5281 ± 0.0351
LPP	16.4 ± 0.7	0.8813 ± 0.0063	0.1288 ± 0.0069	0.3724 ± 0.0100
Isomap	15.4 ± 1.6	0.8542 ± 0.0088	0.1583 ± 0.0095	0.4240 ± 0.0128
RReliefF	16.6 ± 0.4	0.9267 ± 0.0031	0.0796 ± 0.0034	0.2852 ± 0.0059
LinCFA	13.8 ± 1.7	$\mathbf{0.9304 \pm 0.0049}$	$\mathbf{0.0756 \pm 0.0053}$	$\mathbf{0.2753 \pm 0.0108}$
Neural Network	Reduced dim	R^2	*MSE*	*RSE*
Full	18	0.9156 ± 0.0013	0.0917 ± 0.0014	0.3028 ± 0.0039
PCA	12.8 ± 0.3	0.9023 ± 0.0049	0.1061 ± 0.0053	0.3286 ± 0.0079
Supervised PCA	11.6 ± 4.4	0.8832 ± 0.0153	0.1268 ± 0.0167	0.3639 ± 0.0291
Kernel PCA	13.0 ± 0.6	0.9182 ± 0.0032	0.0887 ± 0.0035	0.3026 ± 0.0079
LLE	17.0 ± 0.5	0.6931 ± 0.0248	0.3332 ± 0.0269	0.7568 ± 0.0608
LPP	16.4 ± 0.7	0.8150 ± 0.0142	0.2017 ± 0.0154	0.5421 ± 0.0351
Isomap	15.4 ± 1.6	0.8544 ± 0.0096	0.1581 ± 0.0104	0.4035 ± 0.0191
RReliefF	16.6 ± 0.4	0.9123 ± 0.0094	0.0953 ± 0.0102	0.3117 ± 0.0171
LinCFA	13.8 ± 1.7	$\mathbf{0.9192 \pm 0.0032}$	$\mathbf{0.0877 \pm 0.0035}$	$\mathbf{0.2973 \pm 0.0059}$

Table 9 (continued)

	Life expectancy			
	# samples $n = 1649$		# features $D = 18$	
Ridge Regression	18	0.8302 ± 0.0030	0.1843 ± 0.0032	0.4749 ± 0.0085
Lasso Regression	18	0.7810 ± 0.0045	0.2377 ± 0.0049	0.4948 ± 0.0042

$$\begin{aligned}\Delta_{Bias}^{n\to\infty} = &\frac{1}{2}(1-\rho_{x_1,x_2})(w_1-w_2)^2 \\ &+ \sigma_{x_3}(w_1w_3 - w_2w_3)(\rho_{x_1,x_3} - \rho_{x_2,x_3}) \\ &+ w_3^2\sigma_{x_3}^2\,\frac{(\rho_{x_1,x_3} - \rho_{x_2,x_3})^2}{2(1-\rho_{x_1,x_2})}.\end{aligned} \tag{D20}$$

Bias of the two models, expressions and derivations In the asymptotic setting, letting the relationship between the features and the target be linear with Gaussian noise and assuming unitary variances of the features $\sigma_{x_1} = \sigma_{x_2} = \sigma_{x_3} = 1$, for the one-dimensional regression $\hat{y} = \hat{w}\frac{x_1+x_2}{2}$:

$$\begin{aligned}&\mathbb{E}_x[(\bar{h}(x) - \bar{y})^2] \\ &= \mathcal{F}(x_1, x_2, x_3, w_1, w_2, w_3, \rho_{x_1,x_2}, \rho_{x_1,x_3}, \rho_{x_2,x_3}) \\ &= -\frac{((w_1+w_2)(1+\rho_{x_1,x_2}) + w_3(\rho_{x_1,x_3} + \rho_{x_2,x_3}))^2}{2(1+\rho_{x_1,x_2})} \\ &\quad + \mathbb{E}_x[(w_1x_1 + w_2x_2 + w_3x_3)^2].\end{aligned}$$

For the two-dimensional regression $\hat{y} = \hat{w_1}x_1 + \hat{w_2}x_2$:

Table 10 Extended result of experiments on *Finance* dataset. The total number of samples n has been divided into train (66% of data) and test (33% of data) sets

	Finance			
	# samples $n = 1299$		# features $D = 75$	
Linear regression	Reduced dim	R^2	*MSE*	*RSE*
Full	75	-4.6094 ± 4.2851	8.5071 ± 6.4986	0.7050 ± 0.2453
PCA	26.4 ± 0.7	0.8467 ± 0.0051	0.2324 ± 0.0077	0.4164 ± 0.0089
Supervised PCA	29.2 ± 14.2	0.8670 ± 0.0123	0.2017 ± 0.0187	0.3874 ± 0.0320
Kernel PCA	48.8 ± 0.9	0.8953 ± 0.0020	0.1587 ± 0.0031	$\mathbf{0.3289 \pm 0.0049}$
LLE	49.1 ± 1.4	0.8221 ± 0.0198	0.2698 ± 0.0301	0.4601 ± 0.0481
LPP	30.1 ± 10.2	0.7942 ± 0.0243	0.3122 ± 0.0369	0.4607 ± 0.0304
Isomap	43.2 ± 5.5	0.7097 ± 0.0393	0.4403 ± 0.0596	0.6445 ± 0.0558
RReliefF	45.6 ± 1.8	$\mathbf{0.8972 \pm 0.0029}$	$\mathbf{0.1558 \pm 0.0045}$	0.3328 ± 0.0105
LinCFA	14.6 ± 0.9	0.8838 ± 0.0018	0.1762 ± 0.0028	0.3609 ± 0.0083
SVM for regression	Reduced dim	R^2	*MSE*	*RSE*
Full	75	0.7989 ± 0.0282	0.3049 ± 0.0428	0.5704 ± 0.0635
PCA	26.4 ± 0.7	$\mathbf{0.8010 \pm 0.0215}$	$\mathbf{0.3018 \pm 0.0326}$	$\mathbf{0.5317 \pm 0.0496}$
Supervised PCA	29.2 ± 14.2	0.7594 ± 0.0426	0.3649 ± 0.0645	0.6183 ± 0.0774
Kernel PCA	48.8 ± 0.9	0.7997 ± 0.0272	0.3037 ± 0.0413	0.5684 ± 0.0625
LLE	49.1 ± 1.4	0.6732 ± 0.0506	0.4956 ± 0.0767	0.8259 ± 0.1057
LPP	30.1 ± 10.2	0.7180 ± 0.0509	0.4276 ± 0.0772	0.7093 ± 0.0971
Isomap	43.2 ± 5.5	0.7527 ± 0.0339	0.3750 ± 0.0515	0.6756 ± 0.0858
RReliefF	45.6 ± 1.8	0.7667 ± 0.0314	0.3587 ± 0.0476	0.5965 ± 0.0627
LinCFA	14.6 ± 0.9	0.7070 ± 0.0537	0.4445 ± 0.0815	0.7193 ± 0.1197
XGBoost	Reduced dim	R^2	*MSE*	*RSE*
Full	75	$\mathbf{0.9010 \pm 0.0097}$	$\mathbf{0.1501 \pm 0.0147}$	$\mathbf{0.3255 \pm 0.0308}$
PCA	26.4 ± 0.7	0.8384 ± 0.0496	0.2451 ± 0.0223	0.4501 ± 0.0443
Supervised PCA	29.2 ± 14.2	0.8123 ± 0.0423	0.2847 ± 0.0642	0.4172 ± 0.0337
Kernel PCA	48.8 ± 0.9	0.8169 ± 0.0513	0.2776 ± 0.0743	0.5133 ± 0.0765
LLE	49.1 ± 1.4	0.7836 ± 0.0405	0.3281 ± 0.0614	0.4863 ± 0.0401
LPP	30.1 ± 10.2	0.8142 ± 0.0478	0.2817 ± 0.0725	0.4460 ± 0.0521
Isomap	43.2 ± 5.5	0.8027 ± 0.0126	0.2992 ± 0.0191	0.5069 ± 0.0206
RReliefF	45.6 ± 1.8	0.8975 ± 0.0104	0.1603 ± 0.0158	0.3287 ± 0.0323
LinCFA	14.6 ± 0.9	0.8830 ± 0.0088	0.1774 ± 0.0134	0.3524 ± 0.0229
Neural Network	Reduced dim	R^2	*MSE*	*RSE*
Full	75	0.9025 ± 0.0124	0.1478 ± 0.0188	0.2986 ± 0.0179
PCA	26.4 ± 0.7	0.8851 ± 0.0064	0.1742 ± 0.0097	0.3267 ± 0.0097
Supervised PCA	29.2 ± 14.2	0.9039 ± 0.0087	$\mathbf{0.1411 \pm 0.0132}$	0.2979 ± 0.0141
Kernel PCA	48.8 ± 0.9	0.9037 ± 0.0124	0.1459 ± 0.0187	0.3108 ± 0.0239
LLE	49.1 ± 1.4	0.8419 ± 0.0176	0.2398 ± 0.0268	0.4191 ± 0.0301
LPP	30.1 ± 10.2	0.0287 ± 0.7499	1.4731 ± 1.1374	0.7210 ± 0.1373
Isomap	43.2 ± 5.5	0.7136 ± 0.1107	0.4343 ± 0.1679	0.4836 ± 0.0521
RReliefF	45.6 ± 1.8	0.9026 ± 0.0105	0.1425 ± 0.0159	$\mathbf{0.2928 \pm 0.0146}$
LinCFA	14.6 ± 0.9	$\mathbf{0.9064 \pm 0.0068}$	0.1419 ± 0.0104	0.2974 ± 0.0076

Table 10 (continued)

	Finance			
	# samples $n = 1299$		# features $D = 75$	
Ridge Regression	75	0.8939 ± 0.0067	0.1690 ± 0.0102	0.3383 ± 0.0093
Lasso Regression	75	0.8671 ± 0.0051	0.2016 ± 0.0077	0.3926 ± 0.0057

$$
\begin{aligned}
&\mathbb{E}_x[(\bar{h}(x) - \bar{y})^2] \\
&\quad = \mathcal{G}(x_1, x_2, x_3, w_1, w_2, w_3, \rho_{x_1,x_2}, \rho_{x_1,x_3}, \rho_{x_2,x_3}) \\
&\quad = (w_1 + aw_3)^2 + (w_2 + bw_3)^2 \\
&\qquad + 2\rho_{x_1,x_2}(w_1 + aw_3)(w_2 + bw_3) \\
&\qquad - 2(w_1 + aw_3)(w_1 + w_2\rho_{x_1,x_2} + w_3\rho_{x_1,x_3}) \\
&\qquad - 2(w_2 + bw_3)(w_1\rho_{x_1,x_2} + w_2 + w_3\rho_{x_2,x_3}) \\
&\qquad + \mathbb{E}_x[(w_1x_1 + w_2x_2 + w_3x_3)^2], \\
&\qquad \text{with} \begin{cases} a = \frac{\rho_{x_1,x_3} - \rho_{x_1,x_2}\rho_{x_2,x_3}}{1 - \rho^2_{x_1,x_2}} \\ b = \frac{\rho_{x_2,x_3} - \rho_{x_1,x_2}\rho_{x_1,x_3}}{1 - \rho^2_{x_1,x_2}}. \end{cases}
\end{aligned}
$$

Proof In the one dimensional setting, letting $\bar{x} = \frac{x_1 + x_2}{2}$:

$$
\begin{aligned}
&\mathbb{E}_x[(\bar{h}(x) - \bar{y})^2] \\
&\quad = \mathbb{E}_x[(\mathbb{E}_{\mathcal{T}}[\hat{w}\bar{x}] - (w_1x_1 + w_2x_2 + w_3x_3))^2] \\
&\quad = \sigma^2_{\bar{x}}\,\mathbb{E}_{\mathcal{T}}[\hat{w}]^2 + \mathbb{E}_x[(w_1x_1 + w_2x_2 + w_3x_3)^2] \\
&\qquad - 2\,\mathbb{E}_{\mathcal{T}}[\hat{w}]^2\,\mathbb{E}_x[\bar{x}(w_1x_1 + w_2x_2 + w_3x_3)].
\end{aligned}
$$

Where the last equivalence is due to the fact that train and test data are independent. Then, conditioning on the training features set $\mathbf{X}$ and substituting the value of $\mathbb{E}_{\mathcal{T}}[\hat{w}|\mathbf{X}]$ from Equation (A1), if follows:

Table 11 Extended result of experiments on the first climate dataset. The total number of samples n has been divided into train (66% of data) and test (33% of data) sets

	Climatological I			
	# samples $n = 1038$		# features $D = 136$	
Linear regression	Reduced dim	R^2	*MSE*	*RSE*
Full	136	0.2934 ± 0.0859	0.2891 ± 0.0351	0.7697 ± 0.0090
PCA	12.6 ± 0.4	0.4537 ± 0.0518	0.2228 ± 0.0212	0.5561 ± 0.0327
Supervised PCA	28.8 ± 12.2	0.5821 ± 0.0183	0.1653 ± 0.0075	0.4893 ± 0.0167
Kernel PCA	47.6 ± 1.8	$\mathbf{0.6317 \pm 0.0201}$	$\mathbf{0.1503 \pm 0.0082}$	$\mathbf{0.4229 \pm 0.0113}$
LLE	49.6 ± 0.7	-0.3749 ± 0.1745	0.5609 ± 0.0604	1.3843 ± 0.1421
LPP	36.8 ± 8.1	0.0868 ± 0.0817	0.3725 ± 0.0334	0.6357 ± 0.0706
Isomap	29.6 ± 13.8	0.0986 ± 0.0912	0.2861 ± 0.1296	1.6421 ± 1.5716
RReliefF	37.0 ± 4.3	0.5789 ± 0.0384	0.1718 ± 0.0157	0.4552 ± 0.0334
LinCFA	35.2 ± 3.9	0.5727 ± 0.0435	0.1743 ± 0.0178	0.4514 ± 0.0352
SVM for regression	Reduced dim	R^2	*MSE*	*RSE*
Full	136	0.3881 ± 0.0146	0.2496 ± 0.0060	0.6880 ± 0.0121
PCA	12.6 ± 0.4	0.3805 ± 0.0317	0.2527 ± 0.0129	0.6968 ± 0.0236
Supervised PCA	28.8 ± 12.2	0.5177 ± 0.0284	0.1967 ± 0.0116	0.5725 ± 0.0162
Kernel PCA	47.6 ± 1.8	0.3936 ± 0.0154	0.2473 ± 0.0063	0.6817 ± 0.0121
LLE	49.6 ± 0.7	-0.2471 ± 0.1349	0.5088 ± 0.0550	1.1860 ± 0.0857
LPP	36.8 ± 8.1	-1.2048 ± 0.3119	0.8995 ± 0.1272	5.0265 ± 1.2326
Isomap	29.6 ± 13.8	-0.2140 ± 0.4079	0.4953 ± 0.1664	1.5001 ± 0.9075
RReliefF	37.0 ± 4.3	0.3539 ± 0.0563	0.2636 ± 0.0229	0.7138 ± 0.0629
LinCFA	35.2 ± 3.9	$\mathbf{0.5559 \pm 0.0171}$	$\mathbf{0.1812 \pm 0.0070}$	$\mathbf{0.5255 \pm 0.0171}$
XGBoost	Reduced dim	R^2	*MSE*	*RSE*
Full	136	$\mathbf{0.5297 \pm 0.0233}$	$\mathbf{0.1919 \pm 0.0095}$	0.5416 ± 0.0270
PCA	12.6 ± 0.4	0.3282 ± 0.0780	0.2740 ± 0.0318	0.6909 ± 0.0603
Supervised PCA	28.8 ± 12.2	0.3008 ± 0.0887	0.2783 ± 0.0362	0.7421 ± 0.0115
Kernel PCA	47.6 ± 1.8	0.2437 ± 0.1411	0.3085 ± 0.0694	0.7671 ± 0.1885
LLE	49.6 ± 0.7	-0.5282 ± 0.1808	0.6234 ± 0.0815	1.3419 ± 0.0858
LPP	36.8 ± 8.1	-1.6206 ± 0.4807	1.0691 ± 0.1961	1.8496 ± 0.1857
Isomap	29.6 ± 13.8	-0.5788 ± 0.8403	0.6441 ± 0.3428	1.1578 ± 0.2209
RReliefF	37.0 ± 4.3	0.4851 ± 0.0386	0.2101 ± 0.0158	0.5744 ± 0.0271
LinCFA	35.2 ± 3.9	0.5242 ± 0.0153	0.1941 ± 0.0063	$\mathbf{0.5165 \pm 0.0225}$
Neural Network	Reduced dim	R^2	*MSE*	*RSE*
Full	136	0.3998 ± 0.0746	0.2489 ± 0.0427	0.5958 ± 0.0925
PCA	12.6 ± 0.4	0.4013 ± 0.0344	0.2442 ± 0.0141	0.5863 ± 0.0286
Supervised PCA	28.8 ± 12.2	$\mathbf{0.5542 \pm 0.0612}$	$\mathbf{0.1814 \pm 0.0249}$	$\mathbf{0.4809 \pm 0.0249}$
Kernel PCA	47.6 ± 1.8	0.3731 ± 0.1075	0.2557 ± 0.0438	0.5906 ± 0.0531
LLE	49.6 ± 0.7	-0.3154 ± 0.1234	0.5366 ± 0.0503	1.3091 ± 0.0909
LPP	36.8 ± 8.1	0.0184 ± 0.1095	0.4004 ± 0446	0.7530 ± 0.1206
Isomap	29.6 ± 13.8	-0.4238 ± 0.3765	0.5808 ± 0.1536	1.4640 ± 1.0830
RReliefF	37.0 ± 4.3	0.5144 ± 0.0416	0.1981 ± 0.0169	0.5062 ± 0.0327
LinCFA	35.2 ± 3.9	0.5165 ± 0.0513	0.1973 ± 0.0209	0.5208 ± 0.0453

Table 11 (continued)

	Climatological I			
	# samples $n = 1038$		# features $D = 136$	
Ridge Regression	136	0.5559 ± 0.0347	0.1812 ± 0.0142	0.4695 ± 0.0267
Lasso Regression	136	0.5043 ± 0.0233	0.2022 ± 0.0095	0.5032 ± 0.0507

$$
\begin{aligned}
&\mathbb{E}_x[(\bar{h}(x) - \bar{y})^2 | \mathbf{X}] \\
&= \frac{\sigma^2_{x_1+x_2}}{(\hat{\sigma}^2_{x_1+x_2})^2} \\
&\times (w_1 \hat{\sigma}^2_{x_1} + w_2 \hat{\sigma}^2_{x_2} + (w_1 + w_2) c\hat{o}v(x_1, x_2) \\
&+ w_3(c\hat{o}v(x_1, x_3) + c\hat{o}v(x_2, x_3)))^2 \\
&+ \mathbb{E}_x[(w_1 x_1 + w_2 x_2 + w_3 x_3)^2] \\
&- \frac{2}{\hat{\sigma}^2_{x_1+x_2}} (w_1 \hat{\sigma}^2_{x_1} + w_2 \hat{\sigma}^2_{x_2} + (w_1 + w_2) c\hat{o}v(x_1, x_2) \\
&+ w_3(c\hat{o}v(x_1, x_3) + c\hat{o}v(x_2, x_3))) \\
&\times (w_1 \sigma^2_{x_1} + w_2 \sigma^2_{x_2} + (w_1 + w_2) cov(x_1, x_2) \\
&+ w_3(cov(x_1, x_3) + cov(x_2, x_3))).
\end{aligned}
$$

Considering the asymptotic case, substituting the sample estimators with the real statistical measures and the variances with 1 the result follows.

In the two dimensional setting:

$$
\begin{aligned}
&\mathbb{E}_x[(\bar{h}(x) - \bar{y})^2] \\
&= \mathbb{E}_x[(\mathbb{E}_{\mathcal{T}}[\hat{w_1} x_1 + \hat{w_2} x_2] - (w_1 x_1 + w_2 x_2 + w_3 x_3))^2] \\
&= \sigma^2_{x_1} \mathbb{E}_{\mathcal{T}}[\hat{w_1}]^2 + \sigma^2_{x_2} \mathbb{E}_{\mathcal{T}}[\hat{w_2}]^2 + 2cov(x_1, x_2) \mathbb{E}_{\mathcal{T}}[\hat{w_1}] \mathbb{E}_{\mathcal{T}}[\hat{w_2}] \\
&\quad + \mathbb{E}_x[(w_1 x_1 + w_2 x_2 + w_3 x_3)^2] \\
&\quad - 2(\mathbb{E}_{\mathcal{T}}[\hat{w_1}] \mathbb{E}_x[x_1(w_1 x_1 + w_2 x_2 + w_3 x_3)] \\
&\quad + \mathbb{E}_{\mathcal{T}}[\hat{w_2}] \mathbb{E}_x[x_1(w_1 x_1 + w_2 x_2 + w_3 x_3)]),
\end{aligned}
$$

exploiting again the independence between train and test data.

Then, conditioning on the training set $\mathbf{X}$, substituting the value of $\mathbb{E}_{\mathcal{T}}[\hat{w}|\mathbf{X}]^2$ from Equation (A1) and calling:

$$
\begin{cases}
a = \frac{\hat{\sigma}^2_{x_2} c\hat{o}v(x_1,x_3) - c\hat{o}v(x_1,x_2) c\hat{o}v(x_2,x_3)}{\hat{\sigma}^2_{x_1} \hat{\sigma}^2_{x_2} - c\hat{o}v(x_1,x_2)^2} \\
b = \frac{\hat{\sigma}^2_{x_1} c\hat{o}v(x_2,x_3) - c\hat{o}v(x_1,x_2) c\hat{o}v(x_1,x_3)}{\hat{\sigma}^2_{x_1} \hat{\sigma}^2_{x_2} - c\hat{o}v(x_1,x_2)^2}
\end{cases},
$$

it follows:

Table 12 Extended result of experiments on the second climate dataset. The total number of samples n has been divided into train (66% of data) and test (33% of data) sets

	Climatological II			
	# samples $n = 981$		# features $D = 1991$	
Linear regression	Reduced dim	R^2	*MSE*	*RSE*
Full	1991	0.7529 ± 0.0230	0.2341 ± 0.0209	0.5865 ± 0.0308
PCA	19.0 ± 1.1	0.7639 ± 0.0177	0.2149 ± 0.0162	0.5699 ± 0.0233
Supervised PCA	38.8 ± 5.1	0.8551 ± 0.0151	0.1318 ± 0.0138	0.4946 ± 0.0635
Kernel PCA	35.6 ± 7.9	0.7991 ± 0.0061	0.1830 ± 0.0056	0.5119 ± 0.0072
LLE	15.4 ± 13.8	0.1150 ± 0.0138	0.8058 ± 0.0126	2.3321 ± 0.2983
LPP	42.4 ± 1.5	-3.8622 ± 0.4975	3.5167 ± 0.4531	0.9992 ± 0.0091
Isomap	3.0 ± 0.9	0.1395 ± 0.0134	0.7835 ± 0.0123	2.1913 ± 0.0972
RReliefF	21.4 ± 8.5	0.7648 ± 0.0696	0.2141 ± 0.05441	0.5491 ± 0.0975
LinCFA	37.0 ± 3.6	$\mathbf{0.9203 \pm 0.0073}$	$\mathbf{0.0725 \pm 0.0067}$	$\mathbf{0.2999 \pm 0.0107}$
SVM for regression	Reduced dim	R^2	*MSE*	*RSE*
Full	1991	0.5572 ± 0.0173	0.4032 ± 0.0157	1.0968 ± 0.0376
PCA	19.0 ± 1.1	0.5595 ± 0.0147	0.4010 ± 0.0135	0.9916 ± 0.0149
Supervised PCA	38.8 ± 5.1	0.3852 ± 0.0687	0.5598 ± 0.0625	1.4188 ± 0.1016
Kernel PCA	35.6 ± 7.9	0.5476 ± 0.0142	0.4118 ± 0.0129	1.0791 ± 0.0361
LLE	15.4 ± 13.8	0.0966 ± 0.0425	0.8224 ± 0.0423	1.8402 ± 0.1477
LPP	42.4 ± 1.5	-0.7677 ± 0.1242	1.6095 ± 0.1131	1.5616 ± 0.0815
Isomap	3.0 ± 0.9	0.1064 ± 0.0251	0.8137 ± 0.0228	1.8189 ± 0.1478
RReliefF	21.4 ± 8.5	0.6921 ± 0.08851	0.2803 ± 0.0716	0.6573 ± 0.04842
LinCFA	37.0 ± 3.6	$\mathbf{0.8300 \pm 0.0429}$	$\mathbf{0.1548 \pm 0.0391}$	$\mathbf{0.4886 \pm 0.0704}$
XGBoost	Reduced dim	R^2	*MSE*	*RSE*
Full	1991	0.8680 ± 0.0084	0.1202 ± 0.0077	0.4110 ± 0.0211
PCA	19.0 ± 1.1	0.5157 ± 0.0201	0.4409 ± 0.0183	0.9251 ± 0.0627
Supervised PCA	38.8 ± 5.1	0.3326 ± 0.0713	0.6076 ± 0.0649	1.2514 ± 0.0994
Kernel PCA	35.6 ± 7.9	0.4579 ± 0.0191	0.4935 ± 0.0174	1.089 ± 0.0945
LLE	15.4 ± 13.8	-0.1681 ± 0.1139	1.0635 ± 0.1037	1.5113 ± 0.1809
LPP	42.4 ± 1.5	-0.3715 ± 0.0861	1.2488 ± 0.0784	1.8731 ± 0.1109
Isomap	3.0 ± 0.9	-0.2240 ± 0.1322	1.1145 ± 0.1204	1.4910 ± 0.0679
RReliefF	21.4 ± 8.5	0.7139 ± 0.0774	0.2605 ± 0.0615	0.5847 ± 0.0727
LinCFA	37.0 ± 3.6	$\mathbf{0.8830 \pm 0.0160}$	$\mathbf{0.1065 \pm 0.0145}$	$\mathbf{0.3651 \pm 0.0262}$
Neural Network	Reduced dim	R^2	*MSE*	*RSE*
Full	1991	0.6909 ± 0.0392	0.2814 ± 0.0357	0.5777 ± 0.0383
PCA	19.0 ± 1.1	0.4234 ± 0.0852	0.5249 ± 0.0776	0.7729 ± 0.0417
Supervised PCA	38.8 ± 5.1	0.6940 ± 0.0464	0.2785 ± 0.0423	0.5836 ± 0.0581
Kernel PCA	35.6 ± 7.9	0.6401 ± 0.0642	0.3277 ± 0.0612	0.7106 ± 0.0351
LLE	15.4 ± 13.8	0.1251 ± 0.0374	0.8291 ± 0.0412	1.9947 ± 0.0931
LPP	42.4 ± 1.5	-1.4285 ± 0.1967	1.1300 ± 0.1791	1.0001 ± 0.0294
Isomap	3.0 ± 0.9	0.0038 ± 0.0641	0.9071 ± 0.0583	1.7032 ± 0.1426
RReliefF	21.4 ± 8.5	0.7237 ± 0.1767	0.2515 ± 0.1608	0.5655 ± 0.2367
LinCFA	37.0 ± 3.6	$\mathbf{0.8971 \pm 0.0127}$	$\mathbf{0.0937 \pm 0.0115}$	$\mathbf{0.3390 \pm 0.0234}$

Table 12 (continued)

	Climatological II			
	# samples $n = 981$		# features $D = 1991$	
Ridge regression	1991	0.7885 ± 0.0177	0.1926 ± 0.0162	0.4556 ± 0.0311
Lasso regression	1991	0.9032 ± 0.0046	0.0882 ± 0.0042	0.3095 ± 0.1291

$$
\begin{aligned}
&\mathbb{E}_x[(\bar{h}(x) - \bar{y})^2 | \mathbf{X}] \\
&\quad = \sigma_{x_1}^2 (w_1 + w_3 a)^2 + \sigma_{x_2}^2 (w_2 + w_3 b)^2 \\
&\qquad + 2cov(x_1, x_2)(w_1 + w_3 a)(w_2 + w_3 b) \\
&\qquad + \mathbb{E}_x[(w_1 x_1 + w_2 x_2 + w_3 x_3)^2] \\
&\qquad - 2((w_1 + w_3 a)(w_1 \sigma_{x_1}^2 + w_2 cov(x_1, x_2) + w_3 cov(x_1, x_3)) \\
&\qquad + (w_2 + w_3 b)(w_1 cov(x_1, x_2) + w_2 \sigma_{x_2}^2 + w_3 cov(x_2, x_3))).
\end{aligned}
$$

For the asymptotic case, substituting the sample estimators with the real statistical measures and the variances with 1 the result follows. □

Extension of bias of the models to general variance of third variable Considering a general variance $\sigma_{x_3}^2$ for the third variable x_3, the bias of the models computed in the previous paragraph become (respectively for the one and two dimensional estimates):

$$
\begin{aligned}
&\mathbb{E}_x[(\bar{h}(x) - \bar{y})^2] \\
&\quad = -\frac{((w_1 + w_2)(1 + \rho_{x_1,x_2}) + w_3 \sigma_{x_3}(\rho_{x_1,x_3} + \rho_{x_2,x_3}))^2}{2(1 + \rho_{x_1,x_2})} \\
&\qquad + \mathbb{E}_x[(w_1 x_1 + w_2 x_2 + w_3 x_3)^2], \\
&\mathbb{E}_x[(\bar{h}(x) - \bar{y})^2] = (w_1 + a w_3)^2 + (w_2 + b w_3)^2 \\
&\qquad + 2\rho_{x_1,x_2}(w_1 + a w_3)(w_2 + b w_3) \\
&\qquad - 2(w_1 + a w_3)(w_1 + w_2 \rho_{x_1,x_2} + w_3 \rho_{x_1,x_3} \sigma_{x_3}) \\
&\qquad - 2(w_2 + b w_3)(w_1 \rho_{x_1,x_2} + w_2 + w_3 \rho_{x_2,x_3} \sigma_{x_3}) \\
&\qquad + \mathbb{E}_x[(w_1 x_1 + w_2 x_2 + w_3 x_3)^2], \\
&\qquad \text{with} \begin{cases} a = \sigma_{x_3} \frac{\rho_{x_1,x_3} - \rho_{x_1,x_2}\rho_{x_2,x_3}}{1 - \rho_{x_1,x_2}^2} \\ b = \sigma_{x_3} \frac{\rho_{x_2,x_3} - \rho_{x_1,x_2}\rho_{x_1,x_3}}{1 - \rho_{x_1,x_2}^2}. \end{cases}
\end{aligned}
$$

Table 13 Extended result of experiments on the Boston housing dataset. The total number of samples n has been divided into train (66% of data) and test (33% of data) sets

	Boston Housing			
	# samples $n = 506$		# features $D = 13$	
Linear regression	Reduced dim	R^2	*MSE*	*RSE*
Full	13	0.7027 ± 0.0070	0.2552 ± 0.0060	0.5684 ± 0.0186
PCA	9.2 ± 0.4	0.6636 ± 0.0146	0.2887 ± 0.0126	0.6367 ± 0.0173
Supervised PCA	7.0 ± 3.2	0.6881 ± 0.0178	0.2678 ± 0.0153	0.6173 ± 0.0291
Kernel PCA	11.4 ± 0.8	0.6904 ± 0.0145	0.2657 ± 0.0138	0.6006 ± 0.0256
LLE	12.2 ± 0.4	0.5074 ± 0.1091	0.4229 ± 0.0921	0.9270 ± 0.0932
LPP	11.4 ± 1.1	$\mathbf{0.7059 \pm 0.0121}$	$\mathbf{0.2524 \pm 0.0104}$	$\mathbf{0.5665 \pm 0.0331}$
Isomap	9.2 ± 2.2	0.6483 ± 0.0349	0.3019 ± 0.0301	0.6814 ± 0.0299
RReliefF	11.6 ± 0.4	0.6905 ± 0.0137	0.2657 ± 0.0118	0.5815 ± 0.0196
LinCFA	7.2 ± 0.9	0.6541 ± 0.0206	0.2970 ± 0.0176	0.6530 ± 0.0143
SVM for regression	Reduced dim	R^2	*MSE*	*RSE*
Full	13	0.7748 ± 0.0219	0.1933 ± 0.0188	0.5686 ± 0.0333
PCA	9.2 ± 0.4	0.7502 ± 0.0117	0.2101 ± 0.0102	0.5838 ± 0.0270
Supervised PCA	7.0 ± 3.2	0.7535 ± 0.0424	0.2116 ± 0.0492	0.5863 ± 0.0541
Kernel PCA	11.4 ± 0.8	0.7923 ± 0.0154	0.1711 ± 0.0132	0.5445 ± 0.0206
LLE	12.2 ± 0.4	0.5522 ± 0.0435	0.3845 ± 0.0373	0.9249 ± 0.0671
LPP	11.4 ± 1.1	0.7299 ± 0.0587	0.2318 ± 0.0611	0.6392 ± 0.0310
Isomap	9.2 ± 2.2	0.7076 ± 0.0232	0.2510 ± 0.0199	0.6708 ± 0.0416
RReliefF	11.6 ± 0.4	0.7669 ± 0.0392	0.2001 ± 0.0337	0.5825 ± 0.0685
LinCFA	7.2 ± 0.9	$\mathbf{0.8004 \pm 0.0145}$	$\mathbf{0.1710 \pm 0.0124}$	$\mathbf{0.5115 \pm 0.0214}$
XGBoost	Reduced dim	R^2	*MSE*	*RSE*
Full	13	0.7817 ± 0.0145	0.1875 ± 0.0125	0.5024 ± 0.0356
PCA	9.2 ± 0.4	0.6732 ± 0.0216	0.2819 ± 0.0186	0.6393 ± 0.0428
Supervised PCA	7.0 ± 3.2	0.7279 ± 0.0221	0.2335 ± 0.0290	0.6202 ± 0.0386
Kernel PCA	11.4 ± 0.8	$\mathbf{0.8056 \pm 0.0143}$	$\mathbf{0.1668 \pm 0.0123}$	$\mathbf{0.4784 \pm 0.0192}$
LLE	12.2 ± 0.4	0.4953 ± 0.1273	0.4333 ± 0.0949	0.8546 ± 0.1354
LPP	11.4 ± 1.1	0.7117 ± 0.0288	0.2613 ± 0.0687	0.6487 ± 0.0982
Isomap	9.2 ± 2.2	0.6797 ± 0.0307	0.2750 ± 0.0264	0.6715 ± 0.0173
RReliefF	11.6 ± 0.4	0.7709 ± 0.0361	0.1967 ± 0.0310	0.5252 ± 0.0309
LinCFA	7.2 ± 0.9	0.7591 ± 0.0311	0.2068 ± 0.0267	0.5185 ± 0.0220
Neural Network	Reduced dim	R^2	*MSE*	*RSE*
Full	13	0.8238 ± 0.0070	0.1513 ± 0.0060	0.4350 ± 0.0179
PCA	9.2 ± 0.4	0.8007 ± 0.0134	0.1653 ± 0.0115	0.4589 ± 0.0189
Supervised PCA	7.0 ± 3.2	0.7788 ± 0.0383	0.1898 ± 0.0257	0.4971 ± 0.0471
Kernel PCA	11.4 ± 0.8	0.8106 ± 0.0299	0.1625 ± 0.0271	0.4471 ± 0.0402
LLE	12.2 ± 0.4	0.5026 ± 0.0389	0.4270 ± 0.0334	0.9880 ± 0.0594
LPP	11.4 ± 1.1	0.6175 ± 0.0432	0.3285 ± 0.0371	0.9866 ± 0.1514
Isomap	9.2 ± 2.2	0.7235 ± 0.0192	0.2374 ± 0.0165	0.5949 ± 0.0281
RReliefF	11.6 ± 0.4	0.8187 ± 0.0376	0.1556 ± 0.0323	0.4310 ± 0.0338
LinCFA	7.2 ± 0.9	$\mathbf{0.8336 \pm 0.0107}$	$\mathbf{0.1486 \pm 0.0092}$	$\mathbf{0.4270 \pm 0.0203}$

Table 13 (continued)

	Boston Housing			
	# samples $n = 506$		# features $D = 13$	
Ridge Regression	13	0.7027 ± 0.0069	0.2552 ± 0.0059	0.5700 ± 0.0184
Lasso Regression	13	0.6557 ± 0.0082	0.2956 ± 0.0071	0.6488 ± 0.0264

Appendix E Experiments

This section provides more details and results on the experiments performed in the two-dimensional, three-dimensional and D-dimensional settings.

E.1 Bivariate synthetic data

As introduced in Sect. 6 of the main paper, the experiments performed in the bivariate setting are synthetic. In particular, for each of the six experiments, the data are computed as follows. The samples of the first independent variable x_1 are extracted from a uniform distribution in the interval [0, 1]. The second feature x_2 is a linear combination between the feature x_1 and a random sample extracted from a uniform distribution in the interval [0, 1] (specifically $x_2 = 0.8x_1 + 0.2u$, $u \sim \mathcal{U}([0, 1])$). Finally, the target variable y is a linear combination between the two features x_1, x_2 with weights w_1, w_2 and the addition of a gaussian noise with variance σ^2.

Tables 6,7 provide more details about the bivariate results introduced in Table 1 in the main paper.

In Table 6 the extended results associated with large difference between weights $w_1 = 0.2, w_2 = 0.8$ and three different values of standard deviation of the noise $\sigma \in \{0.5, 1, 10\}$ are reported, repeating $s = 500$ times each experiment, considering $n = 500$ data for training and $n = 500$ data for testing. The quantity $\bar{\rho}$ represents the minimum value of correlation for which it is convenient to aggregate the two features and it is computed exploiting the asymptotic result of Equation (17). From its theoretical values is clear that, for a reasonable amount of variance of the noise, since the weights of the linear model are significantly different, it is better to keep the features separated. This is confirmed by the confidence intervals of the R^2 and the MSE, which are both better in the bivariate case (full) rather than the univariate case (aggr). On the other hand, when the variance of the noise is large, the process becomes much more noisy and it is convenient to aggregate the two features. Considering the empirical $\bar{\rho}$ rather than the theoretical one, which is unknown with real datasets, the only situation where in the majority of the cases it is useful to aggregate is with large variance. It is important also to notice that the confidence intervals are much larger in the noisy setting, meaning that there is much more uncertainty and therefore it is useful to aggregate the two features in order to reduce it.

In Table 7 the same results are reported, considering a linear relationship with small difference between weights $w_1 = 0.47, w_2 = 0.52$. In this setting, since the weights are closer, also with small amount of variance it is convenient to aggregate

Table 14 Extended result of experiments on the *superconductivity* dataset. The total number of samples n has been divided into train (66% of data) and test (33% of data) sets

	Superconductivity			
	# samples $n = 21263$		# features $D = 81$	
Linear regression	Reduced dim	R^2	*MSE*	*RSE*
Full	81	0.7290 ± 0.0013	0.2674 ± 0.0013	0.6025 ± 0.0030
PCA	17.0 ± 0.4	0.5930 ± 0.0073	0.4016 ± 0.0069	0.8137 ± 0.006
Supervised PCA	39.8 ± 11.8	0.6752 ± 0.0370	0.3205 ± 0.0366	0.6863 ± 0.0493
Kernel PCA	49.8 ± 0.4	$\mathbf{0.7470 \pm 0.0031}$	$\mathbf{0.2497 \pm 0.0031}$	$\mathbf{0.5835 \pm 0.0066}$
LLE	38 ± 12.6	0.1422 ± 0.0935	0.8465 ± 0.0926	2.4620 ± 1.2888
LPP	49.8 ± 0.3	0.7101 ± 0.0035	0.2862 ± 0.0024	0.6329 ± 0.0139
Isomap	49.2 ± 1.4	0.6315 ± 0.0302	0.3636 ± 0.0302	0.7633 ± 0.0377
RReliefF	49.8 ± 0.4	0.6927 ± 0.0156	0.3033 ± 0.0154	0.6574 ± 0.0253
LinCFA	49.8 ± 2.9	0.6912 ± 0.0046	0.3048 ± 0.0045	0.6590 ± 0.0102
SVM for regression	Reduced dim	R^2	*MSE*	*RSE*
Full	81	0.8295 ± 0.0012	0.1683 ± 0.0012	0.4305 ± 0.0015
PCA	17.0 ± 0.4	0.8055 ± 0.0010	0.1919 ± 0.0010	0.4662 ± 0.0023
Supervised PCA	39.8 ± 11.8	0.8158 ± 0.0184	0.1817 ± 0.0183	0.4495 ± 0.0105
Kernel PCA	49.8 ± 0.4	0.8253 ± 0.181	0.1724 ± 0.0172	0.4355 ± 0.0168
LLE	38 ± 12.6	0.7990 ± 0.0341	0.1982 ± 0.0337	0.4838 ± 0.0438
LPP	49.8 ± 0.3	$\mathbf{0.8554 \pm 0.0191}$	$\mathbf{0.1427 \pm 0.0119}$	$\mathbf{0.3954 \pm 0.0124}$
Isomap	49.2 ± 1.4	0.7832 ± 0.0053	0.2140 ± 0.0052	0.4959 ± 0.0062
RReliefF	49.8 ± 0.4	0.8217 ± 0.0044	0.1759 ± 0.0043	0.4405 ± 0.0077
LinCFA	49.8 ± 2.9	0.8194 ± 0.0022	0.1782 ± 0.0022	0.4460 ± 0.0041
XGBoost	Reduced dim	R^2	*MSE*	*RSE*
Full	81	$\mathbf{0.9012 \pm 0.0019}$	$\mathbf{0.0975 \pm 0.0019}$	$\mathbf{0.3266 \pm 0.0033}$
PCA	17.0 ± 0.4	0.8832 ± 0.0021	0.1153 ± 0.0021	0.3596 ± 0.0036
Supervised PCA	39.8 ± 11.8	0.8866 ± 0.0317	0.1119 ± 0.0217	0.3523 ± 0.0326
Kernel PCA	49.8 ± 0.4	0.8853 ± 0.0122	0.1132 ± 0.0212	0.3560 ± 0.0142
LLE	38 ± 12.6	0.8388 ± 0.0291	0.1590 ± 0.0128	0.4315 ± 0.0474
LPP	49.8 ± 0.3	0.8853 ± 0.0119	0.1132 ± 0.0179	0.3550 ± 0.0273
Isomap	49.2 ± 1.4	0.8704 ± 0.0116	0.1279 ± 0.0171	0.3778 ± 0.0123
RReliefF	49.8 ± 0.4	0.8898 ± 0.0024	0.1189 ± 0.0034	0.3389 ± 0.0042
LinCFA	49.8 ± 2.9	0.8982 ± 0.0018	0.1004 ± 0.0018	0.3319 ± 0.0028
Neural Network	Reduced dim	R^2	*MSE*	*RSE*
Full	81	0.8556 ± 0.0023	$\mathbf{0.1425 \pm 0.0023}$	$\mathbf{0.3994 \pm 0.0099}$
PCA	17.0 ± 0.4	0.8261 ± 0.0036	0.1716 ± 0.0036	0.4398 ± 0.0077
Supervised PCA	39.8 ± 11.8	0.8492 ± 0.0270	0.1488 ± 0.0269	0.4125 ± 0.0262
Kernel PCA	49.8 ± 0.4	$\mathbf{0.8559 \pm 0.0123}$	0.1430 ± 0.0125	0.4089 ± 0.0197
LLE	38 ± 12.6	0.7870 ± 0.0130	0.2101 ± 0.0304	0.5258 ± 0.0557
LPP	49.8 ± 0.3	0.7932 ± 0.0313	0.2041 ± 0.0712	0.5142 ± 0.0591
Isomap	49.2 ± 1.4	0.7945 ± 0.0069	0.2028 ± 0.0068	0.4931 ± 0.0105
RReliefF	49.8 ± 0.4	0.8428 ± 0.0112	0.1552 ± 0.0110	0.4180 ± 0.0296
LinCFA	49.8 ± 2.9	0.8462 ± 0.0036	0.1518 ± 0.0035	0.4118 ± 0.0178

Table 14 (continued)

	Superconductivity			
	# samples $n = 21263$		# features $D = 81$	
Ridge Regression	81	0.7284 ± 0.0012	0.2680 ± 0.0012	0.6044 ± 0.0031
Lasso Regression	81	0.5844 ± 0.0032	0.4102 ± 0.0031	1.0354 ± 0.0683

the two features if they are sufficiently correlated. Also with the empirical evaluation of the threshold, the two features would be aggregated for the majority of the repetitions of the experiment, leading to a non-worsening of the *MSE* and R^2 coefficient for the aggregated case, as shown by the confidence intervals.

E.1 Three-dimensional synthetic data

This subsection explains more details about the synthetic experiments performed in the three-dimensional setting and introduced in Sect. 6 of the main paper. They show the usefulness of the extension to the three-dimensional linear regression model. In particular the samples of the first independent variable x_1 are extracted from a uniform distribution in the interval [0, 1]. The second feature x_2 is a linear combination between the feature x_1 and a random sample extracted from a uniform distribution in the interval [0, 1] (specifically $x_2 = 0.65x_1 + 0.35u$, $u \sim \mathcal{U}([0, 1])$). The third feature x_3 is a linear combination between the features x_1, x_2 and a random sample extracted from a uniform distribution in the interval [0, 1] ($x_3 = 0.5x_1 + 0.5x_2 + 0.5u$, $u \sim \mathcal{U}([0, 1])$). Finally, the target variable y is a linear combination between the three features x_1, x_2, x_3 with weights $w_1 = 0.4$, $w_2 = 0.6$ that are closer than the third weight $w_3 = 0.2$ and the addition of a gaussian noise with variance $\sigma^2 = 0.25$.

The experiment has been repeated $s = 500$ times with $n = 500$ samples both for the train and the test set. As reported in Table 8, which is an extension of Table 2 reported in the main paper, the theoretical values of correlation thresholds computed from the asymptotic result of Eq. (20) and the empirical ones computed substituting the unbiased estimators of the quantities show that it is convenient to aggregate the two features x_1, x_2. This is confirmed both by the *MSE* and the R^2 coefficient, which are statistically not worse in the aggregated case than in the three dimensional one.

E.3 D-dimensional synthetic data

This subsection provides more details on the application, introduced in Sect. 6 of the main paper, of the algorithm *LinCFA* on a D-dimensional synthetic dataset. In particular, $D = 100$ features are considered. The samples of the first independent variable x_1

Table 15 Extended result of experiments on the *Cifar-10* dataset. The total number of samples n has been divided into train (66% of data) and test (33% of data) sets

	Cifar-10			
	# samples $n = 6000$		# features $D = 3071$	
Linear regression	Reduced dim	R^2	*MSE*	*RSE*
Full	3071	0.1374 ± 0.6204	0.5229 ± 0.3761	0.4775 ± 0.2907
PCA	76	0.8236 ± 0.0192	0.1069 ± 0.0116	0.4273 ± 0.0075
Supervised PCA	76	0.7738 ± 0.0695	0.1371 ± 0.0421	0.4640 ± 0.0727
Kernel PCA	76	0.8237 ± 0.0193	0.1068 ± 0.0117	0.4272 ± 0.0076
LLE	76	0.3830 ± 0.0481	0.3740 ± 0.0291	1.2374 ± 0.1758
LPP	76	-0.7661 ± 0.0936	1.6769 ± 0.2811	1.4799 ± 0.3078
Isomap	76	0.5126 ± 0.0366	0.2954 ± 0.0222	0.8736 ± 0.0188
RReliefF	76	0.3902 ± 0.1408	0.3696 ± 0.0854	1.0677 ± 0.2641
LinCFA	75.6 ± 6.5	$\mathbf{0.9626 \pm 0.0260}$	$\mathbf{0.0227 \pm 0.0157}$	$\mathbf{0.1564 \pm 0.0790}$
SVM for regression	Reduced dim	R^2	*MSE*	*RSE*
Full	3071	0.8643 ± 0.0091	0.0823 ± 0.0051	0.4340 ± 0.0195
PCA	76	0.7996 ± 0.0036	0.1214 ± 0.0022	0.5148 ± 0.0118
Supervised PCA	76	0.7886 ± 0.0488	0.1281 ± 0.0296	0.5219 ± 0.0764
Kernel PCA	76	0.7998 ± 0.0057	0.1213 ± 0.0035	0.5146 ± 0.0118
LLE	76	0.4450 ± 0.0307	0.3364 ± 0.0365	1.0229 ± 0.1251
LPP	76	0.0591 ± 0.0076	0.6098 ± 0.0046	1.4441 ± 0.0676
Isomap	76	0.5424 ± 0.0089	0.2773 ± 0.0054	0.8489 ± 0.0126
RReliefF	76	0.3742 ± 0.1513	0.3793 ± 0.0917	1.0712 ± 0.2110
LinCFA	75.6 ± 6.5	$\mathbf{0.9831 \pm 0.0024}$	$\mathbf{0.0103 \pm 0.0015}$	$\mathbf{0.1317 \pm 0.0098}$
XGBoost	Reduced dim	R^2	*MSE*	*RSE*
Full	3071	0.9791 ± 0.0005	0.0156 ± 0.0003	0.1563 ± 0.0016
PCA	76	0.5946 ± 0.0125	0.2457 ± 0.0076	0.7256 ± 0.0218
Supervised PCA	76	0.6722 ± 0.0429	0.1986 ± 0.0261	0.7008 ± 0.0507
Kernel PCA	76	0.5809 ± 0.1607	0.2540 ± 0.0974	0.7355 ± 0.0467
LLE	76	0.2479 ± 0.1896	0.4559 ± 0.1149	1.0588 ± 0.0339
LPP	76	-1.0676 ± 0.0736	1.3078 ± 0.0971	1.0679 ± 0.0463
Isomap	76	0.4579 ± 0.0130	0.3286 ± 0.0079	0.9852 ± 0.0150
RReliefF	76	0.3306 ± 0.1681	0.4057 ± 0.1019	1.1142 ± 0.2357
LinCFA	75.6 ± 6.5	$\mathbf{0.9794 \pm 0.0060}$	$\mathbf{0.0131 \pm 0.0036}$	$\mathbf{0.1466 \pm 0.0208}$
Neural Network	Reduced dim	R^2	*MSE*	*RSE*
Full	3071	0.5449 ± 0.3039	0.2759 ± 0.1842	0.5709 ± 0.1284
PCA	76	0.1642 ± 0.1186	0.5066 ± 0.0720	0.7414 ± 0.0265
Supervised PCA	76	0.7140 ± 0.1019	0.1733 ± 0.0618	0.4934 ± 0.0353
Kernel PCA	76	0.7842 ± 0.0619	0.1390 ± 0.0545	0.4901 ± 0.0407
LLE	76	0.4341 ± 0.0114	0.3430 ± 0.0069	1.1290 ± 0.0541
LPP	76	0.1556 ± 0.0036	0.5007 ± 0.0492	1.4341 ± 0.0359
Isomap	76	-2.5277 ± 0.7311	2.1386 ± 0.4432	0.9901 ± 0.0356
RReliefF	76	0.1831 ± 0.1675	0.4952 ± 0.1016	1.0741 ± 0.1914
LinCFA	75.6 ± 6.5	$\mathbf{0.9576 \pm 0.0333}$	$\mathbf{0.0257 \pm 0.0202}$	$\mathbf{0.1772 \pm 0.0632}$

Table 15 (continued)

	Cifar-10			
	# samples $n = 6000$		# features $D = 3071$	
Ridge Regression	3071	0.8575 ± 0.0996	0.0864 ± 0.0604	0.2725 ± 0.1515
Lasso Regression	3071	0.9439 ± 0.0095	0.0340 ± 0.0058	0.1821 ± 0.0274

are extracted from a uniform distribution in the interval [0, 1]. Then, each feature x_i, is a linear combination between one of the previous features x_j, $j < i$ and a random sample extracted from a uniform distribution in the interval [0, 1] (specifically $x_i = 0.7x_j + 0.3u,\ u \sim \mathcal{U}([0, 1])$). Finally, the target variable y is a linear combination between the D features $x_1, .., x_{100}$, with coefficients randomly sampled from a uniform distribution in the interval [0, 1], and a gaussian noise with standard deviation $\sigma = 10$.

The algorithm is applied on the features both evaluating the threshold computed with the exact coefficients and with their unbiased estimates. The experiment has been repeated $s = 500$ times on a dataset of $n = 500$ samples both in train and test set, considering both the exact parameters (unkown in practice) and their estimators.

E.4 D-Dimensional real datasets

Data This subsection describes with more details the datasets introduced in Sect. 6 of the main paper to apply the proposed algorithm LinCFA on real data. Additional experiments are also discussed and their results are reported extensively.

The first dataset considered in the main paper focuses on the prediction of Life Expectancy from several factors that can be categorized into immunization related factors, mortality factors, economical factors and social factors. The dataset is available on Kaggle[8] and it is also provided in the repository of this work. It is made of $D = 18$ continuous input variables and a scalar output.

The second dataset reported in the main paper is a financial dataset made of $D = 75$ continuous features and a scalar output. The model predicts the cash ratio depending on other metrics from which it is possible to derive many fundamental financial indicators. The dataset is taken from Kaggle[9] and it is provided in the repository of this work.

Finally, the algorithm is tested on two climatological dataset composed by $D = 136$ and $D = 1991$ continuous climatological features and a scalar target which represents the state of vegetation of a basin of Po river. This datasets have been composed by the authors merging different sources for the vegetation index, temperature and precipitation over different basins (see (Didan 2015; Cornes et al. 2018; Zellner and Castelli 2022)), and they are available in the repository of this work. The main purpose of this regession tasks is to predict the state of the vegetation of a region through meteorological features, which may add insights on the relationship between temperature and precipitation and the state of the vegetation in a highly regulated basin with complex hydrological interactions as the Po River basin.

[8] https://www.kaggle.com/datasets/kumarajarshi/life-expectancy-who

[9] https://www.kaggle.com/datasets/dgawlik/nyse

Table 16 Extended result of experiments on the *Gene Expression* dataset. The total number of samples n has been divided into train (66% of data) and test (33% of data) sets

	Gene Expression			
	# samples $n = 801$		# features $D = 19133$	
Linear regression	Reduced dim	R^2	*MSE*	*RSE*
Full	19133	0.5166 ± 0.0041	0.4992 ± 0.0041	0.8282 ± 0.0089
PCA	220.2 ± 3.5	0.5278 ± 0.0113	0.4877 ± 0.0117	0.8559 ± 0.0089
Supervised PCA	35.2 ± 9.0	0.5289 ± 0.0151	0.4806 ± 0.0156	0.8473 ± 0.0379
Kernel PCA	34.4 ± 12.1	0.4944 ± 0.0104	0.5221 ± 0.0107	1.0009 ± 0.0463
LLE	48.0 ± 3.5	0.2844 ± 0.0114	0.7390 ± 0.0118	1.8372 ± 0.1271
LPP	19.4 ± 7.3	0.3442 ± 0.0161	0.6768 ± 0.0112	1.3779 ± 0.1087
Isomap	19.6 ± 2.9	0.3039 ± 0.0131	0.7189 ± 0.0135	1.5403 ± 0.1370
RReliefF	29.2 ± 9.9	0.2021 ± 0.0773	0.8242 ± 0.0798	1.8724 ± 0.5534
LinCFA	19.6 ± 1.7	$\mathbf{0.5990 \pm 0.0121}$	$\mathbf{0.4141 \pm 0.0125}$	$\mathbf{0.7492 \pm 0.0224}$
SVM for regression	Reduced dim	R^2	*MSE*	*RSE*
Full	19133	0.4825 ± 0.0067	0.5345 ± 0.0069	1.2478 ± 0.0210
PCA	220.2 ± 3.5	0.5277 ± 0.0158	0.4877 ± 0.0164	0.9176 ± 0.0214
Supervised PCA	35.2 ± 9.0	0.4436 ± 0.0190	0.5746 ± 0.0196	1.1301 ± 0.0554
Kernel PCA	34.4 ± 12.1	0.4862 ± 0.0173	0.5306 ± 0.0178	1.0551 ± 0.0353
LLE	48.0 ± 3.5	0.2560 ± 0.0497	0.7684 ± 0.0514	1.6913 ± 0.1328
LPP	19.4 ± 7.3	0.3951 ± 0.0167	0.6492 ± 0.0139	1.2482 ± 0.0224
Isomap	19.6 ± 2.9	0.3048 ± 0.0143	0.7180 ± 0.0147	1.3935 ± 0.0755
RReliefF	29.2 ± 9.9	0.1926 ± 0.1065	0.8338 ± 0.1100	2.0174 ± 0.5152
LinCFA	19.6 ± 1.7	$\mathbf{0.5783 \pm 0.0241}$	$\mathbf{0.4355 \pm 0.0249}$	$\mathbf{0.8450 \pm 0.0274}$
XGBoost	Reduced dim	R^2	*MSE*	*RSE*
Full	19133	0.4966 ± 0.0463	0.5199 ± 0.0478	0.9742 ± 0.0537
PCA	220.2 ± 3.5	0.3212 ± 0.0201	0.7010 ± 0.0206	1.3687 ± 0.0470
Supervised PCA	35.2 ± 9.0	0.3138 ± 0.0414	0.7086 ± 0.0428	1.1762 ± 0.0527
Kernel PCA	34.4 ± 12.1	0.3807 ± 0.0344	0.6395 ± 0.0355	1.2007 ± 0.0475
LLE	48.0 ± 3.5	0.1565 ± 0.0432	0.8712 ± 0.0447	1.2854 ± 0.0317
LPP	19.4 ± 7.3	0.2836 ± 0.0136	0.7175 ± 0.0462	1.6594 ± 0.0449
Isomap	19.6 ± 2.9	0.2342 ± 0.0279	0.7909 ± 0.0288	1.4187 ± 0.0695
RReliefF	29.2 ± 9.9	0.0297 ± 0.1373	1.0022 ± 0.1418	1.6248 ± 0.2209
LinCFA	19.6 ± 1.7	$\mathbf{0.5301 \pm 0.0222}$	$\mathbf{0.4854 \pm 0.0229}$	$\mathbf{0.8275 \pm 0.0301}$
Neural network	Reduced dim	R^2	*MSE*	*RSE*
Full	19133	-5.6864 ± 3.2901	6.9063 ± 3.3982	1.0145 ± 0.0545
PCA	220.2 ± 3.5	-7.3654 ± 0.6679	8.6404 ± 0.6899	0.9864 ± 0.0309
Supervised PCA	35.2 ± 9.0	0.3068 ± 0.0275	0.7159 ± 0.0284	1.0232 ± 0.0814
Kernel PCA	34.4 ± 12.1	0.4869 ± 0.0171	0.5299 ± 0.0177	0.9955 ± 0.0283
LLE	48.0 ± 3.5	0.2949 ± 0.0237	0.7282 ± 0.0245	1.6877 ± 0.1169
LPP	19.4 ± 7.3	0.4594 ± 0.0148	0.5403 ± 0.0285	1.0003 ± 0.0249
Isomap	19.6 ± 2.9	-1.6316 ± 0.4833	1.7886 ± 0.4992	1.0194 ± 0.0154
RReliefF	29.2 ± 9.9	-0.1161 ± 0.1605	1.1528 ± 0.1658	1.2996 ± 0.1127
LinCFA	19.6 ± 1.7	$\mathbf{0.5828 \pm 0.0226}$	$\mathbf{0.4308 \pm 0.0233}$	$\mathbf{0.7607 \pm 0.0209}$

Table 16 (continued)

	Gene Expression			
	# samples $n = 801$		# features $D = 19133$	
Ridge regression	19133	0.5167 ± 0.0041	0.4991 ± 0.0042	0.8279 ± 0.0086
Lasso regression	19133	0.5839 ± 0.0095	0.4340 ± 0.0088	0.7880 ± 0.0174

Table 17 Experiments on real datasets considering the same number of reduced features and linear regression. The total number of samples n has been divided into train (66% of data) and test (33% of data) sets

Quantity	Life exp	Financial	Climatological I	Climatological II
# samples n	1649	1299	1038	981
Full dim (# features D)	18	75	136	1991
Reduced dim PCA	14	12	38	37
Reduced dim supervised PCA	14	12	38	37
Reduced dim LinCFA	14	12	38	37
R^2 full	0.834	−1.441	0.298	−1.402
R^2 PCA	0.830	0.731	0.557	0.795
R^2 supervised PCA	0.809	0.844	0.528	0.866
R^2 LinCFA	0.835	0.885	0.604	0.922
MSE full	0.180	3.702	0.286	1.277
MSE PCA	0.195	0.407	0.181	0.187
MSE supervised PCA	0.207	0.236	0.192	0.121
MSE LinCFA	0.179	0.162	0.145	0.070
RSE full	0.468	0.829	0.748	0.999
RSE PCA	0.479	0.748	0.488	0.519
RSE supervised PCA	0.489	0.412	0.528	0.539
RSE LinCFA	0.465	0.359	0.454	0.299

In this section are also reported the results of the application of the proposed algorithm and the identified baselines on three additional classical regression datasets, which have been selected to further verify the validity of the proposed approach. Although these dataset does not show a particularly meaningful linear dependency between the features and the target, they have been selected to empirically test the validity of the method outside linear contexts, where its validity has theoretical guarantees. In particular, we considered a simple classical dataset with 13 features and 506 samples, the Boston Housing dataset (Harrison and Rubinfeld 1978). This dataset is a classical statistical and ML benchmark collected by the U.S Census Service. Its aim is to inspect the relationship between the price of houses in Boston suburbs and some features related to crime, environment, politics, and social aspects.

As a second additional benchmark dataset, we selected the Superconductivity dataset (Hamidieh 2018) with 81 features and 21263 samples, from the UCI repository.

Table 18 Experiments on additional real datasets considering the same number of reduced features and linear regression. The total number of samples n has been divided into train (66% of data) and test (33% of data) sets

Quantity	Boston housing	Superconductivity
# samples n	506	21263
Full dim (# features D)	13	81
Reduced dim PCA	6	50
Reduced dim supervised PCA	6	50
Reduced dim LinCFA	6	50
R^2 full	0.7261	0.7303
R^2 PCA	0.6863	0.6992
R^2 supervised PCA	0.6192	0.7018
R^2 LinCFA	0.6638	0.6868
MSE full	0.2351	0.2661
MSE PCA	0.2692	0.2967
MSE supervised PCA	0.3268	0.2942
MSE LinCFA	0.2886	0.3090
RSE full	0.5632	0.6006
RSE PCA	0.6556	0.6496
RSE supervised PCA	0.7283	0.6444
RSE LinCFA	0.6858	0.6663

Table 19 Computational time (in seconds) for each dimensionality reduction method on the *Climate II* dataset, confidence intervals produced with five repetitions

Algorithm	Time (Validation Time)
PCA	0.45 ± 0.10 (73.62 ± 2.71)
Supervised PCA	2.51 ± 0.24 (86.35 ± 7.67)
Kernel PCA	0.35 ± 0.07 (16.06 ± 0.71)
LLE	0.25 ± 0.01 (13.11 ± 0.29)
LPP	2.44 ± 0.21 (105.06 ± 2.07)
Isomap	0.27 ± 0.03 (41.10 ± 4.72)
RReliefF	15.86 ± 0.17 (100.69 ± 5.95)
LinCFA	45.83 ± 2.13

Indeed, this repository mantains more than 600 hundreds of datasets to make them available to the ML community, widely used from the eighties in statistics and ML. This specific dataset allows to test the *LinCFA* Algorithm on a larger set of features with many samples where linear models perform poorly w.r.t. non-linear approaches. The dataset is aimed to identify the relationship between the critical temperature of different superconductors (one for each sample) and the features available, which describe the main characteristics of the superconductors (e.g., atomic mass and radius).

Finally, as a third additional dataset, we considered the Cifar-10 dataset (Krizhevsky et al. 2009), that is a famous classification dataset composed by 32×32 images of 10 different classes with 60000 samples. In the experiments we considered 6000 samples at random, to have a number of samples comparable with the

number of features, which enforces a larger risk of overfitting and the need of dimensionality reduction methods. Moreover, since the *LinCFA* algorithm is designed for regression tasks, we transformed the problem into a regression by considering each pixel of each of the three color layers as a feature and removing a pixel, considered as target. In this way, also a more recent ML benchmark dataset with a significantly larger number of features retrieved from images has been added in the comparisons.

In order to further explore the behavior of the LinCFA on a dataset with a large number of features (19133) and a relatively small number of samples (801) from bioinformatics, a *gene expression* dataset from the UCI ML repository has been considered (Fiorini 2016)[10]. In particular, one gene expression has been considered as target variable and the other gene expressions available in the dataset have been considered as features, filtering the constant columns. The dataset is part of the RNA-Seq (HiSeq) PANCAN data set, where the author of the dataset has performed a random extraction of gene expressions of patients having different types of tumor.

Results In this section the extensive results related to the eight datasets under analysis are reported. In particular, we firstly apply the LinCFA Algorithm on the training data to perform the aggregation of features, repeating the experiment five times, bootstrapping the training set with different seeds. The same is done considering the dimensionality reduction methods considered as baselines: PCA, Supervised PCA, Kernel PCA, LLE, LPP, Isomap, RReliefF. For all the methods, from $D = 1$ to $D = 50$ reduced features are considered, and the best performing number of components is selected. The only exception is Cifar-10 dataset, where the proposed algorithm selects $D = 73$ reduced features, and the same number is forced to all the methods, since trying all the different values starting from $D = 1$ would be much computational expensive. Then, supervised learning regression methods have been applied to the reduced features. In particular, given the linearity guarantees of the proposed method, linear regression has firstly been applied and its results are reported in the main paper. Additionally, SVM, XGBoost and Neural Network methods have been also applied, to further inspect the behavior of the method. These approaches have been applied to each dataset, reduced with the proposed method or with the baselines, with default hyperparameters. Additionally, the performances of the full dataset has been reported, considering the same models and additionally taking into account Ridge and Lasso regression, which are regularized variants of the standard linear regression and have better performances on the full dataset. The metrics selected to evaluate the test performances of the methods are the R^2 score, the *MSE* and the Relative Root Mean Squared Error (*RRMSE*). In this way, the performances are considered taking into account the quantity on which the theoretical analysis is based on (the *MSE*), a classical metric that evaluates the performance in regression settings (R^2) and a relative metric that filters the magnitudes of the predictions (*RRMSE*).

The confidence intervals of test scores obtained with linear regression on the datasets considered and with the best performing baseline are reported in Sect. 6 of the main paper. The complete results on all the datasets are reported in one table for each dataset in Tables 9, 10, 11, 12, 13, 14, 15, 16.

Additionally, Tables 17, 18 show the results of the application of PCA and Supervised PCA on a number of reduced features equal to the one identified by LinCFA,

[10] https://archive.ics.uci.edu/dataset/401/gene+expression+cancer+rna+seq

in comparison with LinCFA itself and the not-reduced dataset, considering linear regression. This further analysis is reported to deepdive the performances when these two baselines have the same number of features identified by the proposed method. The majority of the results show that the LinCFA Algorithm leads to competitive performances w.r.t. the baselines, both considering linear and non-linear models and dimensionality reduction approaches. In particular, for the Life Expectancy, Finance, Climate I and Climate II datasets, non-linear methods do not clearly outperform linear regression and the LinCFA Algorithm leads to the best score or close to the best result. The Boston Housing dataset, Superconductivity and Cifar-10 show a better performance with non-linear approaches, with LinCFA that performs similarly to the application of the best performing non-linear metods directly on the full dataset.

To conclude, in Table 19 we provide an example of time complexity of the proposed algorithms in practice, in comparison with the other dimensionality reduction approaches considered. In particular, we considered as an example the *Climate II* dataset, and we ran all the experiments on a BullSequana XH2000 supercomputer using the 3rd generation of AMD EPYC CPUs, allocating 1 node and 32 GB of RAM for the experiment. Considering the time between the beginning and the end of the dimensionality reduction phase, we provide confidence intervals with five repetitions.

The LinCFA algorithm exploits the theoretical threshold to perform the aggregation, therefore there are no hyperparameters to tune. On the contrary, the dimensionality reduction and feature selection approaches considered as baselines need to tune at least the number of desired reduced features. For this reason, we reported both the computational time related to one specific application of the baseline algorithms, once the number of reduced components has been identified, together with the computational time needed to perform the validation (written in parentheses in the table), that we considered to select between 1 and 50 reduced components.

Although this empirical evaluation depends on the choice of considering up to 50 possible reduced features for the baselines and on the different implementations, we can conclude that in general the proposed algorithm has a fixed computational time due to the absence of hyperparameters to tune, at a cost on a larger computational time w.r.t. single computations of the baselines, whose computational time increases depending on the number of parameters that the user is willing to tune and on the extension of the grid search considered.

Acknowledgements This work has been supported by the CLINT research project funded by the H2020 Programme of the European Union under Grant Agreement No 101003876. This paper is supported by PNRR-PE-AI FAIR project funded by the NextGeneration EU program.

Funding Open access funding provided by Politecnico di Milano within the CRUI-CARE Agreement.

Declarations

Conflict of interest The authors have no competing interests to declare that are relevant to the content of this article.

References

Bair E, Hastie T, Paul D et al (2006) Prediction by supervised principal components. J Am Stat Assoc 101(473):119–137. https://doi.org/10.1198/016214505000000628

Barshan E, Ghodsi A, Azimifar Z et al (2011) Supervised principal component analysis: visualization, classification and regression on subspaces and submanifolds. Pattern Recognit 44:1357–1371. https://doi.org/10.1016/j.patcog.2010.12.015

Belkin M, Niyogi P (2001) Laplacian eigenmaps and spectral techniques for embedding and clustering. Adv Neural Inf Process Syst 14

Bishop CM, Nasrabadi NM (2006) Pattern recognition and machine learning, vol 4. Springer, NY

Chandrashekar G, Sahin F (2014) A survey on feature selection methods. Comput Electr Eng 40(1):16–28

Chao G, Luo Y, Ding W (2019) Recent advances in supervised dimension reduction: a survey. Mach Learn Knowl Extr 1(1):341–358. https://doi.org/10.3390/make1010020

Chen T, Guestrin C (2016) XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, https://doi.org/10.1145/2939672.2939785

Coppersmith D, Winograd S (1990) Matrix multiplication via arithmetic progressions. J Symb Comput 9(3):251–280. https://doi.org/10.1016/S0747-7171(08)80013-2

Cornes RC, van der Schrier G, van den Besselaar EJM et al (2018) An ensemble version of the E-OBS temperature and precipitation data sets. J Geophys Res-Atmos 123(17):9391–9409. https://doi.org/10.1029/2017JD028200

Cunningham JP, Ghahramani Z (2015) Linear dimensionality reduction: survey, insights, and generalizations. J Mach Learn Res 16(1):2859–2900

Didan K (2015) Myd13q1 modis/aqua vegetation indices 16-day l3 global 250m sin grid v006, NASA eosdis lp daac. Retrieved from doi https://doi.org/10.5067/MODIS/MYD13Q1.006

Drucker H, Burges CJ, Kaufman L, et al (1996) Support vector regression machines. Adv Neural Inf Process syst 9

Espadoto M, Martins RM, Kerren A et al (2021) Toward a quantitative survey of dimension reduction techniques. IEEE Trans Vis Comput Graph 27:2153–2173. https://doi.org/10.1109/TVCG.2019.2944182

Fiorini S (2016) Gene expression cancer RNA-Seq. UCI Mach Learn Repos https://doi.org/10.24432/C5R88H

Fisher RA (1936) The use of multiple measurements in taxonomic problems. Ann Eugen 7(2):179–188. https://doi.org/10.1111/j.1469-1809.1936.tb02137.x

Golub GH, Reinsch C (1970) Singular value decomposition and least squares solutions. Numer Math 14(5):403–420. https://doi.org/10.1007/BF02163027

Hamidieh K (2018) Superconductivty data. UCI Mach Learn Repos https://doi.org/10.24432/C53P47

Harrison D Jr, Rubinfeld DL (1978) Hedonic housing prices and the demand for clean air. J Environ Econ Manag 5(1):81–102

Hastie T, Tibshirani R, Friedman JH et al (2009) The elements of statistical learning: data mining, inference, and prediction, vol 2. Springer, NY

He X, Niyogi P (2003) Locality preserving projections. Adv Neural Inf Process Syst 16

Hinton GE, Salakhutdinov RR (2006) Reducing the dimensionality of data with neural networks. Science 313(5786):504–507. https://doi.org/10.1126/science.1127647

Hoeffding W (1963) Probability inequalities for sums of bounded random variables. J Am Stat Assoc 58(301):13–30. https://doi.org/10.1080/01621459.1963.10500830
Hoerl AE, Kennard RW (1970) Ridge regression: biased estimation for nonorthogonal problems. Technometrics 12(1):55–67
Hotelling H (1933) Analysis of a complex of statistical variables into principal components. J Educ Psychol 24:498–520. https://doi.org/10.1037/h0071325
Hyvärinen A (1999) Fast and robust fixed-point algorithms for independent component analysis. IEEE T Neural Netw 10(3):626–634. https://doi.org/10.1109/72.761722
Jacod J, Protter P (2004) Probability essentials. Springer Science & Business Media, SN
Jenssen R (2009) Kernel entropy component analysis. IEEE Transact Pattern Anal Mach Intell 32(5):847–860. https://doi.org/10.1109/TPAMI.2009.100
Jing L, Zhang C, Ng MK (2012) SNMFCA: supervised NMF-based image classification and annotation. IEEE T Image Process 21(11):4508–4521. https://doi.org/10.1109/TIP.2012.2206040
Johnson R, Wichern D (2007) Applied multivariate statistical analysis. Pearson Prentice Hall, Hoboken
Kononenko I, Šimec E, Robnik-Šikonja M (1997) Overcoming the myopia of inductive learning algorithms with relief. Appl Intell 7:39–55
Kovalerchuk B, Ahmad MA, Teredesai A (2021) Survey of explainable machine learning with visual and granular methods beyond quasi-explanations. Interpret Artif Intell A Perspect Granul Comput 217–267
Krizhevsky A, Hinton G, et al (2009) Learning multiple layers of features from tiny images
Lafon S, Lee AB (2006) Diffusion maps and coarse-graining: a unified framework for dimensionality reduction, graph partitioning, and data set parameterization. IEEE Transact Pattern Anal Mach Intell 28(9):1393–1403. https://doi.org/10.1109/TPAMI.2006.184
Lahav O, Mastronarde N, van der Schaar M (2018) What is interpretable? using machine learning to design interpretable decision-support systems. arXiv preprint arXiv:1811.10799
Lawrence J (1993) Introduction to neural networks. California Scientific Software, California
Li J, Cheng K, Wang S et al (2017) Feature selection: a data perspective. ACM Comput Surv (CSUR) 50(6):1–45
Lu Y, Lai Z, Xu Y et al (2016) Nonnegative discriminant matrix factorization. IEEE Transact Circuits Syst Video Technol 27(7):1392–1405. https://doi.org/10.1109/TCSVT.2016.2539779
Maurer A, Pontil M (2009) Empirical Bernstein bounds and sample-variance penalization. In: The 22nd conference on learning theory
Pearson K (1901) Liii. on lines and planes of closest fit to systems of points in space. Lond Edinb Dublin Philos Mag J Sci 2(11):559–572. https://doi.org/10.1080/14786440109462720
Pedregosa F, Varoquaux G, Gramfort A et al (2011) Scikit-learn: machine learning in python. J Mach Learn Res 12:2825–2830
Raducanu B, Dornaika F (2012) A supervised non-linear dimensionality reduction approach for manifold learning. Pattern Recognit 45(6):2432–2444. https://doi.org/10.1016/j.patcog.2011.12.006
Ribeiro B, Vieira A, Carvalho das Neves J (2008) Supervised isomap with dissimilarity measures in embedding learning. In: Lect Notes Comput Sc, https://doi.org/10.1007/978-3-540-85920-8_48
Robnik-Sikonja M, Kononenko I (1997) An adaptation of relief for attribute estimation in regression. In: International conference on machine learning https://api.semanticscholar.org/CorpusID:2579394
Roweis ST, Saul LK (2000) Nonlinear dimensionality reduction by locally linear embedding. Science 290(5500):2323–2326. https://doi.org/10.1126/science.290.5500.2323
Sammon JW (1969) A nonlinear mapping for data structure analysis. IEEE T Comput 100(5):401–409. https://doi.org/10.1109/T-C.1969.222678
Shawe-Taylor J, Cristianini N et al (2004) Kernel methods for pattern analysis. Cambridge University Press, Cambridge. https://doi.org/10.1017/CBO9780511809682
Sorzano COS, Vargas J, Montano AP (2014) A survey of dimensionality reduction techniques. arXiv preprint arXiv:1403.2877
Teh Y, Roweis S (2002) Automatic alignment of local representations. In: Advances in neural information processing systems, pp 841–848
Tenenbaum JB, Silva Vd, Langford JC (2000) A global geometric framework for nonlinear dimensionality reduction. Science 290(5500):2319–2323. https://doi.org/10.1126/science.290.5500.2319
Thurstone LL (1931) Multiple factor analysis. Psychol Rev 38(5):406. https://doi.org/10.1037/h0069792
Tibshirani R (1996) Regression shrinkage and selection via the lasso. J Royal Stat Soc Ser B (Methodological) 58(1):267–288

Ulfarsson MO, Solo V (2011) Vector l_0 sparse variable PCA. IEEE T Signal Proces 59(5):1949–1958. https://doi.org/10.1109/TSP.2011.2112653

Van Der Maaten L, Postma E, Van den Herik J et al (2009) Dimensionality reduction: a comparative. J Mach Learn Res 10:66–71

Weinberger KQ, Sha F, Saul LK (2004) Learning a kernel matrix for nonlinear dimensionality reduction. In: Proceedings of the twenty-first international conference on Machine learning, p 106, https://doi.org/10.1145/1015330.1015345

Yu S, Yu K, Tresp V, et al (2006) Supervised probabilistic principal component analysis. In: Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining, pp 464–473, https://doi.org/10.1145/1150402.1150454

Zaki MJ, Meira WJ (2014) Data mining and analysis: fundamental concepts and algorithms. Cambridge University Press, Cambridge

Zellner P, Castelli M (2022) Vegetation health index - 231 m 8 days (version 1.0) [data set]. Eurac Res https://doi.org/10.48784/161b3496-534a-11ec-b78a-02000a08f41d

Zhang SQ (2009) Enhanced supervised locally linear embedding. Pattern Recogn Lett 30:1208–1218. https://doi.org/10.1016/j.patrec.2009.05.011

Zhang Y, Zhang Z, Qin J et al (2018) Semi-supervised local multi-manifold ISOMAP by linear embedding for feature extraction. Pattern Recogn. https://doi.org/10.1016/j.patcog.2017.09.043

Zhang Z, Zha H (2004) Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. SIAM J Sci Comput 26(1):313–338. https://doi.org/10.1137/S1064827502419154

Authors and Affiliations

Paolo Bonetti[1] · Alberto Maria Metelli[1] · Marcello Restelli[1]

✉ Paolo Bonetti
paolo.bonetti@polimi.it

Alberto Maria Metelli
albertomaria.metelli@polimi.it

Marcello Restelli
marcello.restelli@polimi.it

[1] DEIB, Politecnico di Milano, Milan 20133, Italy