The Performance of a Kernel-Based Variable Dimension Reduction Method

Thanh Do Van[1] and Hai Nguyen Minh[2](✉)

[1] Nguyen Tat Thanh University, Ho Chi Minh, Vietnam
dvthanh@cmc-u.edu.vn
[2] Industrial University of Ho Chi Minh City, Ho Chi Minh, Vietnam
nguyenminhhaidhcn@iuh.edu.vn

Abstract. Building forecast models, especially nowcast models, on large data sets of time series variables is a topic of great interest. The most popular method used to build such models is the dynamic factor model, in which factors are extracted from input data sets using the principal component analysis (PCA) or sparse PCA (SPCA) methods. Many studies have shown that the forecast accuracy of models built under such an approach is higher than that of other benchmark models. But the PCA and SPCA methods are only effective when input data sets approximate a hyperplane, while real-world data sets are not always so.

The purpose of this article is to briefly introduce a kernel-based variable dimension reduction method called the KTPCA, and a process of trial and error of this method called the KTPCA#. Experimenting on real-world data sets at the same sampling frequency as well as mixed sampling frequencies shows that the KTPCA# method is superior to the PCA, SPCA, Randomized SPCA (RSPCA), and robust SPCA (ROBSPCA) methods.

Keywords: Time series · Big data · Nowcast · Dimensional reduction · Kernel trick · Factor model

1 Introduction

Forecasting on data sets of a large number of time series variables is a long-standing challenge. There are many approaches to solving the "high dimension curse" in forecasting exercises. First, several regression methods have been proposed to narrow down the coefficient values of some predictors that have a small contribution to the variability of the target variable so that those values are close to zero or zero.

Then the predictors with a coefficient close to zero or zero will be removed from the forecast model of the target variable on input predictors. The RIDGE regression method [1, 2] is intended to narrow the coefficient value of predictors that are less related to the target variable to close to zero, while the LASSO [3, 4], Adaptive LASSO [5], and Elastic Net regression [6] regression methods narrow down the coefficient of less relevant predictors to be zero, and thus the interpretation of forecasts is easier.

C. V. Phan and T. D. Nguyen (Eds.): ICTCC 2022, LNICST 473, pp. 43–60, 2023.
https://doi.org/10.1007/978-3-031-28790-9_4

Thus, the above-mentioned regression methods are all variable selection (or feature selection) ones according to the embedded approach. Bayesian regression methods are also considered variable selection techniques in the same way. One of the commonly used Bayesian regression methods for variable selection is the Bayesian Averaging model [7]. Bayesian regression models are closely related to RIDGE Regression and sparse LASSO regression models [8].

However, when the set of predictors is very large, it is clear that reducing the variable dimension by using the above regression models is not feasible because we still have to perform regression (which is essentially solving optimal exercises) on all predictors to select a subset of suitable predictors. Different from the variable selection approach, the feature learning (or variable transformation) approach transforms the original predictors into new variables with a much smaller number but still retain crucial information in the original predictors.

At present, there have been many feature learning techniques for reducing the variable dimension under different approaches. In scalar time series forecasting exercises, the PCA and SPCA methods are most commonly used [8]. Different from many previous beliefs that the variable dimension reduction performance of the SPCA methods is superior to that of the PCA method, the study [9] has shown experimentally that the dimensional reduction performances of the PCA method and the SPCA methods, including the RSPCA and ROBSPCA methods are competitive.

Van Der Maaten and Postma [10] conducted experiments on real-world data sets and artificial data sets to compare the dimensionality reduction performance of the PCA method with the top 12 nonlinear dimensionality reduction methods such as the kernel principal component analysis (KPCA), Isomap, Maximum Variance Unfolding, Locally Linear Embedding, Laplacian Eigenmap, Hessian LLE, Multilayer Autoencoders, Diffusion Maps, Multidimensional Scaling, Local Tangent Space Analysis, Locally Linear Coordination, and Charting a Manifold. The authors have shown that although the 12 nonlinear dimensionality reduction methods mentioned above can all reduce dimensionality well on artificial data sets, with real-world data sets, none of them reduces the dimensionality better than the PCA method. The article [11] also compares the dimensionality reduction performance of the PCA method and 02 other nonlinear PCA methods, including the robust fuzzy PCA (RFPCA) and KPCA methods, and uses an artificial neural network (ANN) technique for classification. That article has shown that PCA + ANN gives better classification results than RFPCA + ANN and KPCA + ANN.

The PCA method is a typical unsupervised linear learning technique to transform data sets in a high dimensional space to a much lower dimensional space while preserving the maximal variance and covariance structures of the original data set [12]. This method is only effective when data points of the data set of input predictors are approximately a hyperplane. But real-world datasets are not always like that.

The KPCA method is a natural extension of the PCA method [13, 14] and a dimensionality reduction technique for any large data set that may or may not approximate a manifold, whereas the 11 remaining methods are only manifold learning techniques, i.e., they are only suitable for data sets where their data points are approximately a manifold. In another study, we have shown that as a natural extension of the PCA method, the KPCA method is the observation dimension reduction technique, not the variable

dimension reduction technique.This method can be the variable dimension reduction technique, but then it is not a natural extension of the PCA method.

On the other hand, the works [8, 15–19] showed that the effective modeling method on macroeconomic large data sets is the dynamic factor model and Kalman filter, in which the dynamic factor model including the factor bridge equation (BE model for short) and the factor mixed data sampling (MIDAS for short) models are the most applied [20, 21]. In the dynamic factor model, factors are extracted from data sets of input predictors using the PCA method. The presentations above suggest that it is necessary to propose a new variable dimension reduction method that is a natural extension of the PCA method. It can be used to reducet the variable dimension of large data sets where their data points do not approximate a hyperplane and its variable dimension reduction performance must be superior to that of the PCA method.

This article briefly introduces the KTPCA# method and experiments showing that the variable dimension reduction performance of the KTPCA# method is superior to that of the PCA, SPCA, RSPCA, and ROBSPCA methods in forecasting exercises on large data sets of predictors at the same sampling frequency as well as mixed sampling frequency.

The structure of this article is as follows: following this section, Sect. 2 introduces the dynamic factor model as a necessary preparatory for further sections. Section 3 presents briefly the KTPCA# method. Section 4 introduces experimental results comparing the variable dimension reduction performance of the KTPCA# method with the PCA, SPCA, RSPCA, and ROBSPCA methods in forecasting as well as nowcasting exercises. Section 5 finally presents some conclusions and directions for further research in near future.

2 Dynamic Factor Model

2.1 Dynamic Factor Model

Suppose $y = (y_1, \ldots, y_m) \in \mathbb{R}^{\mathbf{m}}$ and $\boldsymbol{X_t} = [x_{1,t}, x_{2,t}, \ldots, x_{N,t}]$, where $x_{\mathrm{i,t}} = \left(x_{i,1}, x_{i,2}, \ldots, x_{i,m}\right) \in \mathbb{R}^{\mathbf{m}}$ are a target variable and a set of predictors, respectively; m and N are the number of observations and predictors, respectively. The model to forecast the target variable according to the predictors has the form:

$$y_t = F(y_{t-k}, \boldsymbol{X_t}) + u_t \tag{1}$$

here u_t is white noise, y_{t-k} the target variable y_t lagged at the order k ($k \geq 1$); F() is a linear or nonlinear function. In practice application, F() is estimated from m observations of the target variable and predictors. When N is very large, we can use the dynamic factor model to build a forecast model. This model assumes that p dynamic factors not observed can capture information in the set of N predictors $\boldsymbol{X_t}$ and $p << N$. In general form, the model is defined as follows [22, 23]:

$$\boldsymbol{X_t} = \boldsymbol{\Lambda f_t} + \varepsilon_t \tag{2}$$

$$\boldsymbol{f_t} = \psi(L)\boldsymbol{f_{t-1}} + \eta_t \tag{3}$$

here L is the lag operator, $\boldsymbol{f_t}$ is the matrix of p hidden factors (not be observed) as columns; $\boldsymbol{\Lambda}$ is the N x p weight matrix of the p factors; ε_t is a vector of idiosyncratic errors, which may be weakly correlated [24]. Stock and Watson [25] showed that the principal component vectors of a data set can consistently estimate hidden factors under the assumptions of the dynamic factor model. If $\boldsymbol{W}$ is the N × p matrix of the principal component vectors as the columns of the covariance matrix $\boldsymbol{S_X} = \frac{1}{\boldsymbol{m}} \boldsymbol{X_t}^{\boldsymbol{T}} \boldsymbol{X_t}$ then the hidden factors at time t are estimated by:

$$\hat{\boldsymbol{f}}_{\boldsymbol{t}} = \boldsymbol{W}^{\boldsymbol{T}} . \boldsymbol{X_t} \tag{4}$$

Then, the out-of-sample forecast of h periods of the target variable y_t is determined by regressing the variable y_{t+h} on $\hat{\boldsymbol{f}}_{\boldsymbol{t}}, \hat{\boldsymbol{f}}_{\boldsymbol{t-1}}, \ldots, \hat{\boldsymbol{f}}_{\boldsymbol{t-q+1}}$. In other words:

$$\hat{y}_{t+h} = \hat{\boldsymbol{f}}_{\boldsymbol{t}}^{\boldsymbol{T}} \hat{\delta}_1 + \hat{\boldsymbol{f}}_{\boldsymbol{t-1}}^{\boldsymbol{T}} \hat{\delta}_2 + \ldots + \hat{\boldsymbol{f}}_{\boldsymbol{t-q-1}}^{\boldsymbol{T}} \hat{\delta}_q \tag{5}$$

where $\hat{\delta}_i \in \mathbb{R}^{\mathbf{p}}$ is the vector of the coefficients estimated by the least-squares method. When the predictors are at frequencies differing from the frequency of the target variable and the number of factors is small, to regress the variable y_{t+h} on $\hat{\boldsymbol{f}}_{\boldsymbol{t}}, \hat{\boldsymbol{f}}_{\boldsymbol{t-1}}, \ldots, \hat{\boldsymbol{f}}_{\boldsymbol{t-q+1}}$ one has to represent the dynamic factor model as a factor state-space model. Details can be found in [26]. Although forecasting the target variable y_t using the dynamic factor model is done according to the two-step procedure, this forecast is still a linear function of predictors in $\boldsymbol{X_t}$. By replacing (4) in (5), and putting $\hat{\theta}_t = \widehat{\boldsymbol{W}} . \hat{\delta}_t$, then Eq. (5) can be written as:

$$\hat{y}_{t+h} = X_t^T \hat{\theta}_1 + X_{t-1}^T \hat{\theta}_2 + \ldots + X_{t-q+1}^T \hat{\theta}_q \tag{6}$$

As such in case the number of factors extracted from $\boldsymbol{X_t}$ is not large, we can estimate the coefficients $\hat{\theta}_i$ in another way, that is by using the RIDGE, LASSO, and Elastic Net regression models. But if the number of factors extracted from $\boldsymbol{X_t}$ is still too large, one only chooses k principal component vectors to include in Eq. (6) so that these k vectors still carry a lot of critical information in these hidden factors.

The build of nowcast models on large mixed-frequency data sets is usually based on the factor BE and MIDAS models.

2.2 The Factor BE Model

It is a linear regression model that links variables at higher frequencies with variables at lower frequencies [21]. The BE model is proposed very naturally and is defined as follows

$$y_t = \sum_{k=1}^{P} b_k y_{t-k} + \sum_{i=1}^{N} \sum_{j=0}^{r_i} \beta_{i_j} x_{i,t-j} + \sum_{j=1}^{M} \sum_{h=0}^{p_j} \gamma_{jh} F_{j,t-h} + c + u_t \tag{7}$$

where y_t is a low-frequency target variable at date t; $x_{i,t}$ are predictors at the same low-frequency as y_t; $F_{j,t}$ are factors at the same frequency as y_t and are aggregated from $F^H_{j,t/S}$ factors at a higher frequency. $F^H_{j,t/S}$ are extracted using a feature learning method from

a large set of original predictors $z^H_{j,t/s}$ sampled at the higher frequency with S values for each low-frequency value. $F^H_{j,t/S}$ as well as $z^H_{j,t/S}$ are called high-frequency components in a mixed frequency model; u_t is residuals; r_i (i = 1,…, N), p_j (j = 1,…, M) and p, are optimal lags of the variables $x_{i,t}$, $F_{j,t}$ and y_t, respectively. The optimal lags can be determined using the Akaike information criterion (AIC) or Bayesian information criterion (BIC).

If the predictors $x_{i,t}$, $z^H_{j,t/S}$, and the target variable are at the same frequency, i.e., S = 1, then $F_{j,t}$=$F^H_{j,t/S}$, , and model (7) becomes an autoregressive distributed lag model.

Model (7) can be rewritten as:

$$\psi(L)y_t = \sum_{i=1}^{N} \beta_i(L)x_{i,t} + \sum_{j=1}^{M} \gamma_j(L)F_{j,t} + c + u_t \tag{8}$$

where L denotes usual lag operator, $\psi(L) = 1 - \sum_{k=1}^{P} b_k L^k$, $\beta_i(L) = \sum_{j=0}^{r_i} \beta_{i_j} L^j$, and $\gamma_j(L) = \sum_{h=0}^{p_j} \gamma_{j_h} L^h$.

2.3 Factor MIDAS Model

The general factor MIDAS model is defined as follows [17, 27]:

$$\psi(L)y_t = \sum_{i=1}^{N} \beta_i(L)x_{i,t} + f(\{F^H_{t/S}\}, \theta, \lambda) + u_t \tag{9}$$

where y_t is the target variable sampled at a low frequency, at date t; $x_{i,t}$ are the predictors at the same low frequency as y_t; $\{F^H_{t/S}\}$ is a set of the factors extracted from a large set of predictors sampled at a higher frequency with S values for each low-frequency value; $\psi(L) = 1 - \sum_{k=1}^{P} b_k L^k$; $\beta_i(L) = \sum_{j=0}^{r_i} \beta_{i_j} L^j$; f is a function describing the effect of the higher frequency data in the low-frequency regression; b = (b_k), $\beta_i = \left(\beta_{i_j}\right)$, θ, and λ are vectors of parameters to be estimated.

U-MIDAS Model

If we like only to include each of the higher frequency components as a predictor in the low-frequency regression, then the model (9) can be given by

$$\varphi(L)y_t = \sum_{i=1}^{N} \beta_i(L)x_{i,t} + \sum_{\tau=0}^{k-1} {F^H_{(t-\tau)/s}}^T \theta_\tau + u_t \tag{10}$$

where T stands transpose, ${F^H_{(t-\tau)/S}}^T$ is a factor at the τ high-frequency periods before t. Then, a distinct θ_τ is associated with each of the S high-frequency lag factors. The number of θ_τ coefficients may be large. If these coefficients are not constrained, then model (10) is called the unrestricted MIDAS model (U-MIDAS for short).

STEP Weighting MIDAS Model (STEP-MIDAS)

If we like only to add an equally weighted sum (or average) of high-frequency data as a predictor in the low-frequency regression, then the MIDAS model (10) can take the form:

$$\psi(L)y_t = \sum_{i=1}^{N} \beta_i(L)x_{i,t} + (\sum_{\tau=0}^{S-1} F^H_{(t-\tau)/S})^T \lambda + u_t \tag{11}$$

The parameter vector λ is associated with a new predictor, and at that time, model (11) is also essentially the factor BE model, and it is almost the same as the model defined by the formula (9). Model (11) is called the equally weighted aggregation model.

In the STEP-MIDAS model, the coefficients on high-frequency data are restricted using a STEP function. Specifically, this model is derived from the model (11) and has the form:

$$\psi(L)y_t = \sum_{i=1}^{N} \beta_i(L)x_{i,t} + \sum_{\tau=0}^{K-1} (F^H_{(t-\tau)/S})^T \varphi_\tau + u_t \tag{12}$$

where K is a chosen number of lagged high-frequency periods to use (K may be less than or greater than S); δ is a step length; $\varphi_m = \theta_i$, for $K = \text{int}(m/\delta)$.

Polynomial Almon Weighting MIDAS Model (PAW-MIDAS model): for each high frequency up to k, the regression coefficients of high-frequency components in the model (14) are modeled as a p-dimensional lag polynomial in the MIDAS parameter θ, and the model is as follows:**.**

$$\psi(L)y_t = \sum_{i=1}^{N} \beta_i(L)x_{i,t} + \sum_{\tau=0}^{k-1} (F^H_{(t-\tau)/S})^T \left(\sum_{j=0}^{p} \tau^j \theta_j \right) + u_t \tag{13}$$

where p is the Almon polynomial order, and the chosen number of lags k may be less than or greater than S. Then, the number of coefficients to be estimated depends on the polynomial order and not the number of high-frequency lags.

The model (13) is called the Polynomial Almon weighting MIDAS model.

The exponential almon weighting MIDAS model (or EAW-MIDAS model) is a type of non-polynomial distributed lag MIDAS model. This model uses exponential weights and a lag polynomial of degree 2. Specifically, the EAW-MIDAS model takes the form [27]:

$$\psi(L)y_t = \sum_{i=1}^{N} \beta_i(L)x_{i,t} + \sum_{\tau=0}^{k-1} (F^H_{(t-\tau)/S})^T \left(\frac{\exp(\tau\theta_1 + \tau^2\theta_2)}{\sum_{j=0}^{k} \exp(j\theta_1 + j^2\theta_2)} \right) + u_t \tag{14}$$

where k is a chosen number of lags. Then the exponential weighting function and the lag polynomial depend on the two MIDAS coefficients θ_1 and θ_2.

It can be said that models (11) and (10) can be considered extreme polar ones of MIDAS models. Model (10) offers the greatest flexibility but requires a large number of coefficients. The models from (12) to (14) are considered to fall between these two models. By offering different restrictions on the effects of high-frequency variables at various lags, one can create the middle MIDAS models between the equally weighted aggregation model (11) and the U-MIDAS model (10). Such restrictions on the effects of high-frequency variables can be realized through MIDAS weighting functions.

3 Kernel-Based Factor Extraction Method

3.1 The KTPCA Method

Suppose $\mathbf{X} = [x_1, x_2, \ldots, x_N]$, here $x_i = (x_{i1}, x_{i2}, \ldots, x_{im})^T \in \mathbb{R}^{\mathbf{m}}$**;** m, N are respectively the number of observations, the number of predictors (or the dimension number of the data set **X**).

Dimensional reduction is a mapping $\mathcal{R} : \mathbb{R}^N \to \mathbb{R}^p$.

$(x_{1j}, x_{2j}, \ldots, x_{Nj}) \mapsto (z_{1j}, z_{2j}, \ldots, z_{pj})$, so that p $\ll N$ and the dataset of $\{z_1, z_2, \ldots, z_p\} = \mathcal{R}(\{x_1, x_2, \ldots, x_N\})$ still retains the critical information of the dataset **X,** here $z_j = (z_{j1}, z_{j2}, \ldots, z_{jm})^T, j = 1, 2, \ldots, p$; $(x_{1j}, x_{2j}, \ldots, x_{Nj})$ is called a data point of **X**. Without loss of generality, it can be assumed that **X** is mean-centered, i.e. $\sum_{j=1}^{m} x_{ij} = 0$ for all i = 1,…, N. The KTPCA method can be briefly presented as follows:

Suppose $\mathbf{K} = [\kappa(x_i, x_j)]$ is the kernel matrix of **X** corresponding to the kernel function κ. If the function κ is symmetric and positive definite (semi-deterministic), so is the matrix **K** [10]. Then the eigenvalues of **K** are positive (or non-negative) and **K** is diagonalized by an orthogonal matrix of **K**'s eigenvectors. Each factor of **X** is determined by linearly projecting the set **X** onto an eigenvector of the matrix **K** and the set of factors corresponding to the **K'**s eigenvectors is determined by:

$$\mathbf{PC}_{mxN} = \mathbf{X}_{\mathbf{mxN}}.\widetilde{\mathbf{E}}_{\mathbf{NxN}} \tag{15}$$

here $\widetilde{\mathbf{E}}_{\mathrm{NxN}}$ is the NxN matrix of N eigenvectors of the matrix **K**. In this matrix the columns are sorted in descending order of their respective eigenvalues.

Because N is very large, in the practical application one only needs to choose k vectors corresponding to the first k columns in the matrix $\mathbf{PC}_{mxN}$ for replacing the set of input predictors in forecast models on the data set **X**.

The question is, how is the number of chosen factors used to replace input predictors determined? With the note that when the kernel function $\kappa(x_i, x_j)$ is the dot product of two vectors, we see that the KTPCA method becomes the PCA method, and in this case, according to [28], the number of first principal components can be determined by the Cross-validation, Screening, or Use the Cumulative Percentage of Variance (or eigenvalues or mean eigenvalues), the Coded Error Function, the Akaike Information Criterion, the Minimum Description Length Criterion or the Variance of Reconstruction Error,… If the cumulative percentage of variance is used, it should be in the range (70% to 90%). If that percentage is <70%, then there is usually not enough information from the original data set in the chosen factors, and if this percentage is >90%, there may be a situation that is too suitable to make forecasts using the model. The selection of the number of factors in the KTPCA method can be done by selecting the number of principal component factors as in the PCA method.

3.2 The KTPCA# Method

Another problem is how to choose the suitable kernel function $\kappa(x_i, x_j)$ when implementing the KTPCA variable dimension reduction method due to kernels satisfying the requirements are very rich and varied. In the practical application, polynomial kernels $\kappa(x_i, x_j) = (c_1 \langle x_i, x_j \rangle + c_2)^d$ and Gaussian kernels $\kappa(x_i, x_j) = \exp(\frac{-\|x_i - x_j\|^2}{2\rho^2})$ are the most commonly used [14, 29]. In the case of Gaussian kernels the parameter ρ^2 needs to be taken around the value:

$$\frac{c}{N} \sum_{i=1}^{N} min_{i \neq j} \|\mathbf{x}_i - \mathbf{x}_j\|^2 \tag{16}$$

where c is a user-defined tuning parameter [30].

Thus, to choose the most suitable kernel κ, it is necessary to perform a trial and error process based on the criterion that the error of the forecast model is the smallest. The standard mean error of forecast models (Root Mean Squared Errors – RMSE forshot) is used in this article to evaluate the forecast accuracy of forecast and nowcast models. The KTPCA# method can be described in a general way in Fig. 1 below.

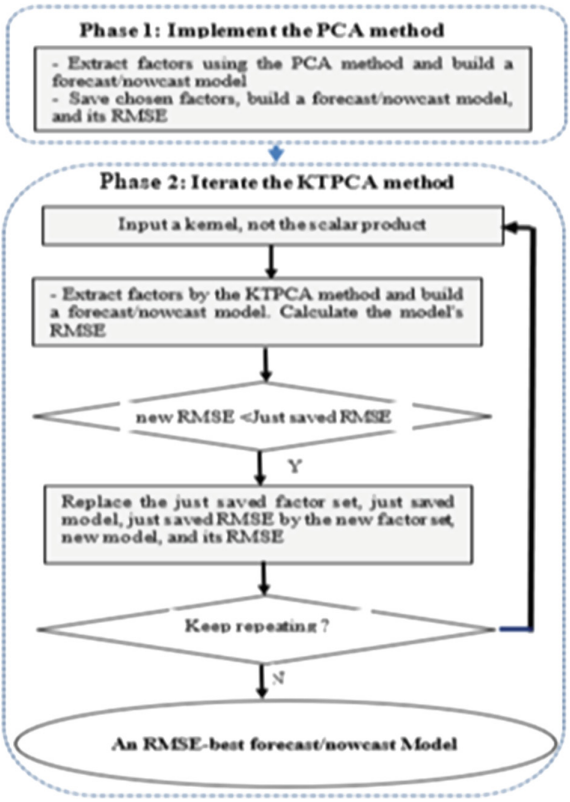

Fig. 1. Framework of the KTPCA # method

The KTPCA# method is an iterative process of kernels. That process depends on the user. The more iterative process with the more reasonable selection of kernels, the higher the probability of choosing the most suitable kernel with the forecast/nowcast model having the lowest forecast error. At the end of that iterative process, we get the result of variable dimension reduction and a forecast/nowcast model having the highest forecast accuracy among the models built according to the chosen kernels. Figure 1 also shows that the process of building a forecast/nowcast model on time series large data set using the KTPCA# method has integrated the variable dimension reduction phase and the phase using a forecast algorithm in one.

4 Dimensional Reduction Performance of KTPCA# Method

The variable dimension reduction performance of a method is measured by the RMSE of a forecast/nowcast model built based on a regression model on factors extracted from the data set of original predictors using this method. In this section, the variable dimension reduction performance of the KTPCA# method is compared with that of the PCA, SPCA, RSPCA, and ROBSPCA methods. Those are the methods commonly used and currently considered the most effective factor extraction methods for building forecast/nowcast models on scalar time series large data sets.

The extraction of factors by the SPCA, RSPCA, and ROBSPCA methods is performed using the 'Sparsepca' package [31], while this extraction by the KTPCA# method is performed by a self-developed tool based on the "Kernlab" package [32] and framework presented in Fig. 1 above.

Because the KTPCA# method can be used to reduce the variable dimension for data sets of predictors at the same sampling frequency as well as at mixed sampling frequencies, so the performance of this method is also compared and evaluated in the two types of these data sets.

4.1 For the Same Sampling Frequency Data Sets

Experimental Data Sets

Data sets used for the experiment include 04 real data sets of the Vietnam economy and 07 data sets in the UCI-Machine Learning Repository [33]. They are named EXP, VN30, CPI, VIP, Residential Building, S&P 500, DJI, and Nasdaq, Air Quality, Appliances energy, and Superconductivity. Table 1 below introduces some statistical characteristics of these data sets. In this table, the number of attributes is the number of predictors excluding the target variable. Missing data were processed by the weighted moving average method.

Table 1. The statistical characteristics of experimental data sets

Data sets	Type of data set	Type of Attribute	No. of Observs	No. of Attributes	Missing data	The target variable	Frequency
EXP	Time series	Real	60	63	No	Total exports	Monthly
VN30	Time series	Real	366	34	No	VN30 index	Daily
CPI	Time series	Real	72	102	No	CPI index	Monthly
VIP	Time Series	Real	60	265	No	Production value of industries	Monthly

(*continued*)

Table 1. (*continued*)

Data sets	Type of data set	Type of Attribute	No. of Observs	No. of Attributes	Missing data	The target variable	Frequency
Residential Building	Multivariate	Real	371	27[1]	No	Sales prices	
S&P500	Time series	Real	1760	52	Yes	S&P 500 index	Daily
DJI	Time series	Real	1760	81	Yes	Dow Jones Index	Daily
NASDAQ	Time series	Real	1760	81	Yes	Nasdaq Index	Daily
Air Quality	Time series	Real	9348	12	Yes	CO of Air	Hourly
Appliances Energy	Time series	Real	19704	23	No	The energy use of Appliances (wh)	Every 10 min
SuperConduct	Multivariate	Real	21263	81	No	Critical temperature	

[1]: Remove the column V1: zip codes

Experimental Method

To compare the variable dimension reduction performance of the KTPCA# method with those of PCA, SPCA, RSPCA, and ROBSPCA methods, for the experimental 11 data sets, excepting the special polynomial kernel $\boldsymbol{PL}_0(X_i, X_j) = \prec X_i, X_j>$ (then the KTPCA and PCA methods are the same), the article unanimously chooses only five other kernels to experiment with the KTPCA method, where two polynomial kernels and three Gaussian kernels. Specifically, they are as follows: for the EXP, VN30, CPI, Air quality, and Appliances Energy data sets, two chosen polynomial kernels are of the forms $\boldsymbol{PL}_1(X_i, X_j) = < X_i, X_j > +0.5)^2$ and $\boldsymbol{PL}_2(X_i, X_j) = < X_i, X_j > +0.5)^3$, while for the remaining data sets, two chosen polynomial kernels are $\boldsymbol{PL}_1(X_i,X_j) = 0.5 < X_i, X_j > +0.5)^2$ and $\boldsymbol{PL}_2(X_i,X_j) = 0.5 < X_i, X_j > +0.5)^3$. For Gaussian kernels with the parameter ρ^2, the three chosen kernels with this parameter value correspond to equal to, smaller than, and larger than the expected value, and they are denoted $\boldsymbol{GA}_3$, $\boldsymbol{GA}_4$, and$\boldsymbol{GA}_5$, respectively. The level of smaller or larger than the expected value depends on each specific data set and is based on the analysis of the number of factors extracted by the KTPCA method with the Gaussian kernel's parameter ρ^2 to obtain the expected value. The autoregressive distributed lag model is used to build forecast models on data sets of predictors at the same sampling frequency.

According to the work [34], when building forecast models on data sets of economic and financial variables at the monthly frequency using the autoregressive distributed lag model, the optimal lag of all variables in the models, in general, is 6, 12, or even 24.

Except for the EXP data set, here the maximum lag of variables in a forecast model is determined according to the experience [34] and is 6, for the remaining 10 data sets, the

maximum lag of all variables is precisely determined using a combination of the Akaike information criterion (AIC) and the seasonality of these time-series data sets. Thus, the maximum lag of the factors extracted using different variable dimension reduction methods for each data set is generally different.

All factors are tested for unit roots and transformed to stationary time series before performing model estimate, and in the estimated models, all variables are high statistical significance, at least below 10%. The conditions for the best, linear, and unbiased estimate (BLUE for short) are guaranteed.

Results

The average minimum distance between two data vectors of the 11 data sets used for the experiment is presented in Table 2. These values are the expected value of the parameter ρ^2 in the Gaussian kernel for corresponding data sets. Specifically, each expected value is an important guideline to choose suitable Gaussian kernels $\boldsymbol{\kappa}(x_i, x_j) = \exp\left(-\frac{\|X_i - X_j\|^2}{2\rho^2}\right)$ when performing the KTPCA method on a given corresponding data set.

Table 2. The average minimum distance between two data vectors of data sets

Data sets	*EXP*	*VN30*	*CPI*	*VIP*	*Res. Buil.*	*S&P500*
The average minimum distance between two data vectors of predictors ($=\rho^2$)	$e^{-0.5639}$	$e^{7.046}$	$e^{1.461}$	$e^{34.906}$	$e^{26.919}$	$e^{15.426}$
	DJI	NASDAQ	AirQuality	App. Energy	SuperCond.	
	$e^{15.171}$	$e^{12.971}$	$e^{18.977}$	$e^{13.595}$	$e^{22.353}$	

With the cumulative eigenvalue percentage threshold of 75% for all the aforementioned variable dimension reduction methods and the experimental data sets, results of variable dimension reduction, the RMSE of built forecast models on the factors extracted by the PCA, SPCA, RSPCA, ROBSPCA methods as well as the KTPCA# method with the $\boldsymbol{PL}_0$, $\boldsymbol{PL}_1$, $\boldsymbol{PL}_2$, $\boldsymbol{GA}_3$, $\boldsymbol{GA}_4$, and$\boldsymbol{GA}_5$ kernels are shown in Table 3, where the values of the parameter ρ^2 in the kernels $\boldsymbol{GA}_4$, and$\boldsymbol{GA}_5$ are shown in Table 4, in which the codes from S1 to S11 are assigned to the 11 experimental data sets in Table 1 from top to bottom. Column 3 in Table 3 shows the most suitable kernel in the 06 experimented kernels, the number of chosen factors, and the RMSE of the forecast model built on these chosen factors.

Table 3 shows that for the EXP data set, with the maximum optimal lag determined as in [34] and to be 6, it is not possible to perform regressions for estimating forecast models based on Eq. (8) on factors extracted by the PCA, SPCA, RSPCA, and ROBSPCA methods because 60 data observations are not enough degrees of freedom to perform. However, if using the KTPCA# method to reduce the variable dimension of this data set, this limitation is overcome.

Table 3 also shows that for each of the 11 experimental data sets, it is always possible to find the most suitable kernel so that the RMSE of the forecast model of the target

Table 3. The dimension reduction performance of the KTPCA# method

Datasets	Methods	KTPCA#	PCA	SPCA	RSPCA	ROBSPCA
EXP	No. of factors & kernel	GA_5, 6	14	10	10	10
	RMSE	**0.0104**	NA	NA	NA	NA
VN30	No. of factors & kernel	GA_3, 14	14	14	14	15
	RMSE	**0.1819**	0.1895	0.1968	0.1968	0.2054
CPI	No. of factors & kernel	GA_4, 6	4	4	4	4
	RMSE	**0.4452**	1.4836	1.0659	1.0673	1.0659
VIP	No. of factors & kernel	PL_1, 4	4	4	4	4
	RMSE	**672.66**	715.96	826.28	1373.57	2642.83
Res. Building	No. of factors & kernel	GA_4, 2	1	1	1	1
	RMSE	**919.9**	1152.4	1152.5	1152.5	1151.2
S&P500	No. of factors & kernel	GA_4, 2	1	1	1	1
	RMSE	**61.60**	161.415	161.441	161.441	161.441
DJI	No. of factors & kernel	PL_0, 1	1	1	1	1
	RMSE	**91.82**	**91.82**	309.24	309.24	309.23
NASDAQ	No. of factors & kernel	PL_1, 1	1	1	1	1
	RMSE	**81.05**	365.97	85.47	85.47	85.46
Air Quality	No. of factors & kernel	GA_4, 5	1	1	1	1
	RMSE	**50.297**	71.459	71.499	71.499	71.427
App. Energy	No. of factors & kernel	GA_4, 6	3	3	3	3
	RMSE	**98.81**	101.74	101.76	101.76	101.75
SuperCon.	No. of factors & kernel	GA_4, 2	2	2	2	2
	RMSE	**26.094**	27.314	27.332	27.332	27.319

variable on factors extracted by the KTPCA method is equal to or less than the RMSE of forecast models built on factors extracted by the PCA, SPCA, RSPCA, and ROBSPCA methods.

Table 4. Values of the parameter ρ^2 in the $\boldsymbol{GA}_4$ and $\boldsymbol{GA}_5$ Gaussian kernels

Datasets	*S1*	*S2*	*S3*	*S4*	*S5*	*S6*	*S7*	*S8*	*S9*	*S10*	*S11*
$\boldsymbol{GA}_4$	$e^{-0.693}$	$e^{6.9}$	$e^{0.7}$	$e^{34.1}$	e^{22}	e^{8}	e^{8}	e^{8}	e^{18}	$e^{12.8}$	$e^{18.5}$
$\boldsymbol{GA}_5$	$e^{-0.183}$	$e^{7.5}$	e^{2}	$e^{35.3}$	$e^{27.2}$	$e^{15.7}$	$e^{15.5}$	$e^{13.5}$	e^{20}	$e^{24.5}$	$e^{22.8}$

4.2 For Data Sets of Predictors at Mixed Sampling Frequencies

In this section, the regression model used to build nowcast models is the factor BE model, the factor U-MIDAS model, and some other restricted MIDAS models, including the STEP-MIDAS, PAW-MIDAS, and EAW-MIDAS models.

Experimental Datasets

Datasets used for experiments are presented in Table 5. Specifically, they include 07 datasets in the UCI-Machine Learning Repository introduced in Table 1 and three real-world data sets of Vietnam's economy in which the CPI data set is in Table 1, and the RGDP and IIP data sets are new. With the data sets already in Table 1, the target variables in these datasets are aggregated at a lower frequency so that they are mixed sampling frequency data sets. The value of the aggregated target variable is determined in one of two ways: the first way is the arithmetic average of S values of this variable, and the second way is the sum of the S values, where S is the number of high-frequency values for each low-frequency value. It is different for different data sets, as shown in Table 5. Precisely, the value of the aggregated target variable of the Air Quality, Residential Building, S & P 500, DJI, Nasdaq, Super Conductivity, and CPI data sets is calculated according to the first way, while for the Appliances Energy data set, it is calculated according to the second way.

Furthermore, the article has assumed that the number of working days in the months is the same and equals 20 days per month. This assumption is close to the number of working days in the months. So, in Table 5, when S equals 20, we understand that the target variable is at the monthly frequency, while the predictors are at the daily frequency.

Table 5. The statistical characteristics of experimental datasets

Statistical Characteristics	RGDP	CPI	IIP	Air Quality	App. Energy
Characteristics of dataset	Time-series	Time-series	Time-series	Time-series	Time-series
Variable Characteristics	Real	Real	Real	Real	Real

(*continued*)

Table 5. (*continued*)

Statistical Characteristics	RGDP	CPI	IIP	Air Quality	App. Energy
Number of low-frequency variables	3	3	1	1	1
Number of high-frequency variables	87	102	42	12	27
Total number of observations	72	72	1840	9348	19704
Number of low-frequency observations	24	24	92	779	3284
S - the number of high-frequency values for a low-frequency value[2]	3	3	20	12	6
Missing data	No	No	Yes	Yes	No
The target variable	The growth rate of GDP	Consumer Price Inflation	Index of Industrial production	The Air CO	Energy use of Appliances
Statistical Characteristics	Res. Build	S&P 500	DJI	NASDAQ	SuperCond
Characteristics of dataset	cross data	Time-series	Time-series	Time-series	cross data
Variable Characteristics	Real	Real	Real	Real	Real
Number of low-frequency variables	1	1	1	1	1
Number of high-frequency variables	27	52	81	81	81
Total number of observations	366	1760	1760	1760	21260
Number of low-frequency observations	122	88	88	88	1063

(*continued*)

Table 5. (*continued*)

Statistical Characteristics	RGDP	CPI	IIP	Air Quality	App. Energy
S - the number of high-frequency values for each low-frequency value	3	20	20	20	20
Missing data	No	Yes	Yes	Yes	No
The target variable	Sales Prices	S&P500 Index	DJI index	NASDAQ Index	Critical temperature

[2]: The total number of observations (or the number of high-frequency observations) = S * the number of low-frequency observations.

In the RGDP and CPI datasets, there are several other predictors at the same frequency as the target variable, and they are at the quarterly frequency.

4.3 Experimental Method

To build nowcast models, first, the time series target variable at a low frequency, the predictors at the same frequency as the target variable, and the factors extracted from higher frequency predictors are transformed into stationary time series. The criterion for selecting the number of extracted high-frequency factors is also their cumulative eigenvalues percentage [28].

Nowcast models are estimated under ideal cases, namely: for nowcast models built based on the factor BE model, the optimal common lag of the target variable and predictors are determined precisely using the AIC and all variables in the built nowcast models are statistically significant, at least at the $< 10\%$ level. For nowcast models built based on the factor MIDAS models, the optimal common lag of the target variable and predictors at the same frequency as the target variable is determined as in the nowcast model built based on the BE model. In contrast, the optimal lags of different factors at higher frequencies are generally different and determined as follows: first, determine the optimal common lag based on the RMSE criterion, then determine the individual optimal lag for each factor using the RMSE criterion. This approach is always suitable for the STEP-MIDAS and PAW-MIDAS models but for the EAW-MIDAS and U-MIDAS models, it is only suitable if the number of the factors in nowcast models is pretty small. On the contrary, that is, the number of factors in the model is not small, the article determines a maximum optimal common lag for all high-frequency factors by using the RMSE criterion.

To compare the variable dimension reduction performance of the KTPCA# method with the PCA, SPCA, RSPCA, and ROBSPCA methods, for the KTPCA# method, the article also only experiments with the 06 kernels mentioned above. The article uses the packet "Midas-r" in R.CRAN [35] to build nowcast models on mixed sampling frequency data sets.

4.4 Results

The average minimum distance between two data vectors of the 08 experimental datasets in Table 1 is presented in Table 2, while for the RGDP and IIP data sets, this value corresponds to $\rho^2 = e^{1.464}$ and $\rho^2 = e^{8.978}$.

With the same cumulative eigenvalue minimum threshold of 75% for all the five variable dimension reduction methods, the experimental datasets, and the five regression models: BE, STEP-MIDAS, PAW-MIDAS, EAW-MIDAS, and U-MIDAS, the RMSE of built nowcast models on the chosen extracted factors is shown in Table 6. This table includes five subtables 6a, 6b, 6c, 6d, and 6e containing the RMSE of nowcast models built based on the factor BE, STEP-MIDAS, PAW-MIDAS, EAW-MIDAS, and U-MIDAS models, respectively. Here, factors are extracted from the ten aforementioned experimental datasets using the PCA, SPCA, RSPCA, ROBSPCA, and KTPCA# methods. Here SET1,…, SET10 is another writing way for the 10 data sets in Table 4 in order from left to right and from top to bottom.

Table 6 shows that for all the factor regression models and the all experimental datasets, it is always possible to choose a suitable kernel so that the RMSE of the nowcast model built on factors extracted by the KTPCA method corresponding to this

Table 6. Performance of the KTPCA# method for mixed sampling frequency data sets

6a. BE	PCA	SPCA	RSPCA	ROBSPCA	KTPCA#	**6b.STEP**	PCA	SPCA	RSPCA	ROBSPCA	KTPCA#
SET1	**0.000493**	0.000788	0.00079	0.000788	**0.000493**	SET1	**0.00744**	0.009727	0.009722	0.009727	**0.00744**
SET2	**0.000183**	0.000485	0.00051	0.000485	**0.000183**	SET2	0.008236	0.00439	0.004387	0.00439	**0.003948**
SET3	1.348981	1.203836	1.04437	1.545299	**0.56932**	SET3	26.52232	21.39361	28.86856	28.13315	**8.78805**
SET4	0.615228	0.611051	0.6104	0.61106	**0.592861**	SET4	**0.630038**	0.63004	0.63004	**0.630038**	**0.630038**
SET5	377.6252	377.2618	377.262	377.0618	**360.131**	SET5	**385.1972**	385.68	385.68	385.3454	**385.1972**
SET6	565.5147	565.523	565.523	565.516	**513.6189**	SET6	430.8412	430.8373	430.8373	430.8397	**421.709**
SET7	**4.3074**	4.3076	4.3076	4.3076	**4.3074**	SET7	259.8844	259.8083	257.6644	259.8065	**72.7871**
SET8	57.1033	56.4321	56.4321	56.4321	**56.2975**	SET8	4101.593	4101.958	4101.958	4102.275	**1024.708**
SET9	18.5945	18.5941	18.5941	18.5489	**18.3479**	SET9	1419.767	1419.807	1419.807	1419.756	**687.2987**
SET10	13.5381	13.5397	13.5425	13.5429	**13.3662**	SET10	14.3425	14.3462	14.3462	14.3431	**13.9649**
6c.PAW	PCA	SPCA	RSPCA	ROBSPCA	**KTPCA#**	**6d.EAW**	PCA	SPCA	RSPCA	ROBSPCA	**KTPCA#**
SET1	**0.000026**	0.000208	0.000197	0.000208	**0.000026**	SET1	0.005232	0.005274	0.005277	0.005274	**0.004544**
SET2	**0.001473**	0.001833	0.001819	0.001833	**0.001473**	SET2	0.006911	0.005465	0.007418	0.005465	**0.00509**
SET3	1.1268	0.7342	0.7508	0.6208	**0.0433**	SET3	4.4983	4.7174	4.3561	4.3146	**4.1810**
SET4	0.6298	0.6293	0.6402	0.6298	**0.6174**	SET4	0.4762	0.4765	0.4765	0.4761	**0.4392**
SET5	384.4007	384.4115	384.3218	384.3270	**384.0171**	SET5	385.4549	385.4515	385.4515	385.4597	**385.000**
SET6	404.3389	399.4798	399.4798	399.4800	**399.3498**	SET6	504.9074	504.9076	504.9076	504.9069	**379.0157**
SET7	40.7019	42.8444	42.8444	42.8444	**33.6159**	SET7	**2.806**	2.953	2.953	2.953	**2.8060**
SET8	337.8048	337.8025	337.8025	337.8026	**311.3913**	SET8	240.0	239.7	239.7	239.5	**118.900**
SET9	107.9667	107.9666	107.9666	107.9666	**107.0302**	SET9	82.2279	82.1254	82.1254	82.0357	**36.3656**
SET10	13.9580	13.9580	13.9580	13.9580	**13.9485**	SET10	13.9322	13.931	13.931	13.9322	**13.9302**
6e.U	PCA	SPCA	RSPCA	ROBSPCA	KTPCA#	**6e.U**	PCA	SPCA	RSPCA	ROBSPCA	KTPCA#
SET1	0.00204	0.000951	0.000919	0.000951	**0.000699**	SET6	430.1182	430.1732	430.1732	430.1286	**389.1229**
SET2	**0.000109**	0.002515	0.002955	0.002512	**0.000109**	SET7	0.000701	5.58E-05	0.0000558	0.0000587	**0.0000546**
SET3	**0.0283**	0.9860	0.3109	0.6632	**0.0283**	SET8	2.932	2.931	2.931	2.931	**2.9300**
SET4	0.4054	0.4058	0.4058	0.4055	**0.3330**	SET9	0.8993	0.8992	0.8992	0.8992	**0.8841**
SET5	376.9851	377.4016	377.4016	376.8008	**351.2000**	SET10	14.0231	14.0219	14.0219	14.0231	**13.9115**

kernel is less than or equal to the RMSE of nowcast models built on factors extracted by the PCA, SPCA, RSPCA, and ROBSPCA methods.

5 Conclusions

This article briefly introduced the variable dimension reduction method KTPCA#. Experimental results on real data sets of predictors at the same sampling frequency and mixed sampling frequencies show that the dimensional reduction performance of this method is higher than that of the PCA, SPCA, RSPCA, and ROBSPCA methods. Here the variable dimension reduction performance of a method is measured by the RMSE of forecast/nowcast models built based on the autoregressive distributed lag model/ the factor BE, U-MIDAS, and some other restricted MIDAS models such as STEP-MIDAS, PAW-MIDAS, and EAW-MIDAS, where factors are extracted from a data set of predictors using this method. The KTPCA# method is an iterative, trial-and-error process over the kernels. It is uncomplicated and can be used to reduce the variable dimension of large data sets.

Bagging and Boosting are currently emerging as prediction/classification methods on data sets of a large number of predictors. In the upcoming study, we will compare the forecast accuracy of models built based on the dynamic factor regression model where factors are extracted by the KTPCA# method with the forecast accuracy of models built based on these methods.

References

1. Hoerl, A.E., Kennard, R.W.: Ridge regression: Biased estimation for nonorthogonal problems. Technometrics **12**(1), 55–67 (1970)
2. Marquardt, D.W., Snee, R.D.: Ridge regression in practice. Am. Stat. **29**(1), 3–20 (1975)
3. Tibshirani, R.: Regression shrinkage and selection via the lasso. J. R. Stat. Soc. Ser. B **58**(1), 267–288 (1996)
4. Zou, H., Hastie, T., Tibshirani, R.: Sparse principal component analysis. J. Comput. Graph. Stat. **15**(2), 265–286 (2006)
5. Zou, H.: The adaptive lasso and its oracle properties. J. Am. Stat. Assoc. **101**(476), 1418–1429 (2006)
6. Zou, H., Hastie, T.: "Regularization and variable selection via the elastic net." J. R. Stat. Soc. Ser. B statistical Methodol, **67**(2), 301–320 (2005)
7. Hoeting, J.A., Madigan, D., Raftery, A.E., Volinsky, C.T.: Bayesian model averaging: a tutorial (with comments by M. Clyde, David Draper and EI George, and a rejoinder by the authors. Stat. Sci. **14**(4), 382–417 (1999)
8. Kapetanios, G., Papailias, F., et al.: Big data & macroeconomic nowcasting: methodological review. Econ. Stat. Cent. Excell. Natl. Inst. Econ. Soc. Res. (2018)
9. Do Van, T.: Dimensionality reduction performance of sparse PCA methods. In: Cong Vinh, P., Huu Nhan, N. (eds.) ICTCC 2021. LNICSSITE, vol. 408, pp. 138–148. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-92942-8_12
10. Postma, E.: Dimensionality reduction : a comparative review dimensionality reduction : a comparative review. J. Mach. Learn. Res. 10, 1–35 (2007). October 2016
11. Zhong, X., Enke, D.: Forecasting daily stock market return using dimensionality reduction. Expert Syst. Appl. **67**, 126–139 (2017)

12. Shlens, J.: A tutorial on principal component analysis. *arXiv Prepr. arXiv1404.1100* (2014)
13. Schölkopf, B., Smola, A., Müller, K.-R.: Nonlinear component analysis as a kernel eigenvalue problem. Neural Comput. **10**(5), 1299–1319 (1998)
14. Schölkopf, B., Smola, A.J.: A short introduction to learning with kernels. In: Mendelson, S., Smola, A.J. (eds) Advanced Lectures on Machine Learning. Lecture Notes in Computer Science, vol. 2600, pp. 41–64. Springer, (2003). https://doi.org/10.1007/3-540-36434-X_2
15. Urasawa, S.: Real-time GDP forecasting for Japan: a dynamic factor model approach. J. Jpn. Int. Econ. **34**, 116–134 (2014)
16. Bok, B., Caratelli, D., Giannone, D., Sbordone, A.M., Tambalotti, A.: Macroeconomic nowcasting and forecasting with big data. Annu. Rev. Econom. **10**, 615–643 (2018)
17. Castle, J.L., Hendry, D.F., Kitov, O.I.: Forecasting and Nowcasting Macroeconomic Variables: A Methodological Overview (2013)
18. Bańbura, M., Rünstler, G.: A look into the factor model black box: publication lags and the role of hard and soft data in forecasting GDP. Int. J. Forecast. **27**(2), 333–346 (2011)
19. Foroni, C., Marcellino, M.: A comparison of mixed frequency approaches for nowcasting Euro area macroeconomic aggregates. Int. J. Forecast. **30**(3), 554–568 (2014)
20. Bai, J., Ghysels, E., Wright, J.H.: State space models and MIDAS regressions. Econom. Rev. **32**(7), 779–813 (2013)
21. Kim, H.H., Swanson, N.R.: Mining big data using parsimonious factor, machine learning, variable selection and shrinkage methods. Int. J. Forecast. **34**(2), 339–354 (2018)
22. Geweke, J.: The dynamic factor analysis of economic time series. Latent Var. socio-economic Model (1977)
23. Panagiotelis, A., Athanasopoulos, G., Hyndman, R.J., Jiang, B., Vahid, F.: Macroeconomic forecasting for Australia using a large number of predictors. Int. J. Forecast. **35**(2), 616–633 (2019)
24. Yu, Y., Samworth, R.J.: Discussion of Large Covariance Estimation by Thresholding Principal Orthogonal Complements by Fan, Liao and Mincheva (2013)
25. Stock, J.H., Watson, M.W.: Forecasting using principal components from a large number of predictors. J. Am. Stat. Assoc. **97**(460), 1167–1179 (2002)
26. Foroni, C., Marcellino, M.G.: A survey of econometric methods for mixed-frequency data (2013). SSRN 2268912
27. Ghysels, E., Kvedaras, V., Zemlys, V.: Mixed frequency data sampling regression models: the R package midasr. J. Stat. Softw. **72**(1), 1–35 (2016)
28. Zhang, Y., Li, S., Teng, Y.: Dynamic processes monitoring using recursive kernel principal component analysis. Chem. Eng. Sci. **72**, 78–86 (2012)
29. Kim, K.I., Franz, M.O., Scholkopf, B.: Iterative kernel principal component analysis for image modeling. IEEE Trans. Pattern Anal. Mach. Intell. **27**(9), 1351–1366 (2005)
30. Rathi, Y., Dambreville, S., Tannenbaum, A.: Statistical shape analysis using kernel PCA. In: Image Processing: Algorithms and Systems, Neural Networks, and Machine Learning, vol. 6064, p. 60641B (2006)
31. Erichson, N.B., Zheng, P., Manohar, K., Brunton, S.L., Kutz, J.N., Aravkin, A.Y.: Sparse principal component analysis via variable projection. SIAM J. Appl. Math. **80**(2), 977–1002 (2020)
32. Karatzoglou, A., Smola, A., Hornik, K., Zeileis, A.: kernlab-an S4 package for kernel methods in R. J. Stat. Softw. **11**(9), 1–20 (2004)
33. "UCI-Machine Learning Repository."
34. Wooldridge, J.M.: Introductory Econometrics: A Modern Approach. Nelson Education (2016)
35. Kedaras, V.K., Zemlys, V., Imports, M., NumDeriv, M.: Package midasr (2021)