Erasmus University Rotterdam
Erasmus School of Economics
Bachelor Thesis Econometrics and Operations Research

Real-time inflation forecasting using dimension reduction techniques and shrinkage methods

Yanting Wang (576204)

Supervisor:	Eoghan O'Neill
Second assessor:	Andreas Pick
Date final version:	1st July 2024

Abstract

This paper explores real-time inflation forecasting by addressing high-dimensional data and non-linearities in the dataset. It employs various Principal Component Analysis (PCA) methods and shrinkage techniques to tackle multicollinearity and overfitting. Dimension reduction methods include linear, squared, quadratic, and kernel PCA. Shrinkage methods involve the adaptive Minnesota prior, ridge regression, Least Absolute Shrinkage and Selection Operator (Lasso), Elastic Net, and kernel ridge regression. Results show that quadratic or kernel PCA combined with the Minnesota prior or Elastic Net outperform benchmark AR models for one-month-ahead forecasts, while for one-quarter-ahead forecasts, quadratic or squared PCA with the Minnesota prior or ridge regression excel. However, robustness checks indicate variability in performance across non-linear PCA methods, with models using squared or quadratic PCA being less robust than those using kernel PCA. Overall, ridge and kernel ridge regressions demonstrate greater robustness in one-quarter-ahead forecasts, suggesting a trade-off between accuracy and robustness.

1 Introduction

Forecasting real-time macroeconomic conditions is a compelling topic for both economists and econometricians. Building such a model involves intricacies as it could be correlated with numerous potential economic factors, which brings out significant challenges. In the study by Stock & Watson (1999), they forecast inflation more accurately by employing 168 different economic-related data, instead of relying solely on the unemployment rate. This indicates that including a larger dimensional dataset might benefit the prediction. However, using a high-dimensional dataset could lead to potential problems in real practice. The process could be computationally expensive and overfitting would undermine the validity and accuracy of final results. This, together with the increasing interest in forecasting macroeconomic data sparked a surge of relevant studies utilising machine learning and other dimension reduction techniques (Bai & Ng, 2008; Goulet Coulombe et al., 2022; Hauzenberger et al., 2023; Kim & Swanson, 2018).

Another challenge is that the previous traditional models assume linearity in the observed dataset, while this is not the case in empirical work. Stock & Watson (1998) point out that allowing for non-linearities could benefit models that are tightly parameterised, in other words, models expected to have a limited number of variables. Modern studies typically aim to build a model with informative but parsimonious factors, suggesting that possible non-linearities need to be considered.

My paper concentrates on the above-discussed challenges, which are problems rooted in the high-dimensional data and the non-linearities. Therefore, I use different techniques to develop models that can effectively capture complex patterns in economic data while maintaining parsimony and improving forecast accuracy. This study focuses on two key aspects: dimension reduction and shrinkage methods. Inspired by Hauzenberger et al. (2023), my paper also employs the adaptive Minnesota prior and various types of PCA. In addition to these, I incorporate other regularisation methods.

PCA is commonly applied to process and transform macroeconomic data. In this paper, besides the traditional linear PCA, I also employ other non-linear PCA techniques including squared PCA, quadratic PCA, and kernel PCA (Gaussian and polynomial kernels) to model non-linearities (see, e.g., Bai & Ng, 2008; Hauzenberger et al., 2023; Ludvigson & Ng, 2009).

To further address multicollinearity and improve computational efficiency, shrinkage methods are applied after the dimension reduction. These include shrinkage priors (Chan et al., 2020; Hauzenberger et al., 2023) and penalised regressions (Bai & Ng, 2008; Kim & Swanson, 2018). Classic penalisation methods such as ridge regression, Lasso, and Elastic Net are incorporated. Additionally, my paper uses kernel ridge regression to capture non-linearity within the dataset (Exterkate et al., 2016). All models are compared against the benchmark autoregressive model (AR) model using the Diebold-Mariano (DM) test and the Root Mean Square Error (RMSE).

Loosening assumptions on linearity, this paper proposes models combining both dimension reduction and shrinkage methods and thus, my **research question** is formulated as:

How can dimension reduction and shrinkage methods improve the accuracy of real-time inflation forecasting?
with the following subquestions,

- *How do linear and non-linear dimension reduction techniques affect the accuracy of real-time inflation forecasts?*
- *Which combinations of dimension reduction techniques and shrinkage methods are more effective in forecasting real-time inflation?.*

This paper makes twofold extra contributions to the existing literature. While many previous works focus on one single method to improve the model performance, such as Exterkate et al. (2016) on kernel PCA and Chan (2021) on the Minnesota shrinkage prior, my approach sequentially applies data transformation and shrinkage methods. My paper builds on the work by Hauzenberger et al. (2023), who apply various non-linear dimension reduction techniques and shrinkage priors, including different types of PCA as well as other machine learning techniques such as diffusion maps, autoencoders, and local linear embeddings. My paper narrows down the scope by solely focusing on PCA. I extend their study by incorporating additional regularisation methods mentioned in other studies (Coulombe et al., 2021; Kim & Swanson, 2018). Moreover, my paper introduces a new combination of PCA and kernel ridge regression.

Additionally, my study updates the dataset to include the most recent data up to April 2024, compared to Hauzenberger et al. (2023), which only includes data up to January 2021. The inclusion of post-COVID data is expected to provide valuable new insights and a more comprehensive understanding of underlying trends.

This paper finds that, in the main-sample forecast using pre-COVID data, combining quadratic or kernel PCA with the Minnesota prior or Elastic Net performs best in the one-month-ahead forecast. For one-quarter-ahead forecasts, quadratic or squared PCA combined with the Minnesota prior or ridge regression outperforms the AR model. However, many models, particularly those using linear PCA with the Minnesota prior or Lasso regression, underperform in the DM test. These findings support the assumption that non-linear PCA outperforms linear PCA, which is validated again in the robustness check. Despite these insights, no definitive best combination of dimension reduction techniques and shrinkage methods is identified. Contradictory to the main-sample forecast, the robustness check suggests that incorporating squared or quadratic PCA may perform worse than other non-linear PCA methods, and using kernel PCA could achieve better performance. Furthermore, models using ridge or kernel ridge regressions are most robust for one-quarter-ahead forecasts. The decision to choose the preferred model specification involves a trade-off between accuracy and robustness.

The rest of the paper is structured as follows. Section 2 provides an extensive review of the literature. Section 3 discusses the dataset used in this study. Section 4 explains the methodology in detail. Finally, I present the results and discuss the conclusions.

2 Literature

There is extensive research that exists on macroeconomic forecasting (see, e.g., Hauzenberger et al., 2023; Kim & Swanson, 2018; Sermpinis et al., 2014; Smeekes & Wijler, 2018; Stock & Watson, 1999). Despite the various problematic aspects of this task, this paper focuses on two main popular topics. The first is to model the non-linearity. Bai & Ng (2008) shows that non-linear models using squared and quadratic principal components (PCs) often outperform linear models. Goulet Coulombe et al. (2022) also indicate in their study that including non-linearity is a "game changer" for forecasting, where they propose two commonly used approaches, kernel ridge regression and random forests, and the former is applied in this paper.

The second concern is to deal with high-dimensional data, which not only increases computation time but also introduces possible multicollinearities. Researchers aim to only retain the variables that are informative and relevant to the predicted target and to develop relatively simple models using parsimonious factors (see, e.g., Bai & Ng, 2006, 2008; Hauzenberger et al., 2023; Kim & Swanson, 2018; Ludvigson & Ng, 2009; Stock & Watson, 2002). PCA remains a widely used technique in these studies.

There are many variations of PCA derived from the basic linear one (Hotelling, 1933), which reduces the original data space to a lower dimension with a linear function. While the linear PCA could be useful in many circumstances, it fails to capture more complicated real-life data. This is why non-linear PCA is needed, as this technique allows for both linearity and non-linearity between the original data and transformed latent factors. Different types of non-linear PCA have been developed. Ludvigson & Ng (2009) apply both squared and cubed factors in their study for predicting excess bond returns as they effectively reduce the Bayesian Information Criteria (BIC). Bai & Ng (2008) employ both squared and quadratic PCs for inflation forecasting, and they outperform one another interchangeably over different sample sizes. In this paper, I use both squared PCA and quadratic PCA while also performing the linear PCA for comparison.

In addition, another form of PCA is applied in my paper. When mapping the original dataset into higher dimensions, problems could occur. Since the dataset is of a higher dimension, it is computationally costly if the dimension is too high. Proposed by Schölkopf et al. (1998), kernel PCA successfully solves this problem, where there is no need to calculate the data transformation explicitly, instead, the computation is implicitly finished using the kernel trick. Nahil & Lyhyaoui (2018) use different kernels to predict the short-term stock price indices and show that kernel PCA outperforms other methods.

These PCA methods help improve efficiency when it comes to "big data" or "data mining", but the size of parameters could still be large and possible problems including overfitting and multicollinearity could still exist. As the intersection of machine learning with econometric forecasting has become increasingly popular in recent studies, other researchers propose to use machine learning techniques to solve this problem further (see, e.g., Bai & Ng, 2008; Hauzenberger et al., 2023; Kim & Swanson, 2018; Korobilis & Pettenuzzo, 2016; Nahil & Lyhyaoui, 2018; Sermpinis et al., 2014). Therefore, this paper also considers to include a series of shrinkage methods.

The first approach is to include the shrinkage prior. Sims (1980) points out that the a priori restrictions can lead to significant estimation errors in out-of-sample forecasting. He proposes the use of Bayesian methods to shrink "unconstrained coefficients" in empirical macroeconomics effectively. Litterman (1979) first introduces the Minnesota prior, which serves as a foundation for later research. Giannone et al. (2015) create a new prior by combining different priors including the Minnesota prior, while other researchers like Korobilis & Pettenuzzo (2016) and Chan et al. (2020) develop new "adaptive" Minnesota priors. This paper uses the adaptive Minnesota prior suggested by Chan et al. (2020), which is also applied later by Hauzenberger et al. (2023).

Another popular technique is penalised regression, with Ridge, Lasso, and Elastic Net being the three common methods. Previous studies show that forecast accuracy is improved if penalised regressions are conducted after reducing the data dimension (Bai & Ng, 2008; Kim & Swanson, 2018; Smeekes & Wijler, 2018). My paper adds value by including kernel ridge regression besides these three methods. Building upon the standard ridge regression, the kernel ridge regression maps the dataset to a possible higher dimension using the kernel trick before directly operating on the original feature space. The non-linearities in the initial regressors are considered and thus, improve the prediction accuracy, especially for macroeconomic data (Exterkate et al., 2016; Sermpinis et al., 2014). I follow the steps described in Exterkate et al. (2016) to build the kernel ridge regression using the Gaussian kernel. As for the choice of hyperparameters, my paper uses the popular k-fold cross-validation (CV) approach as suggested in many papers (Coulombe et al., 2021; Exterkate et al., 2016; Goulet Coulombe et al., 2022; Hauzenberger et al., 2023; Kim & Swanson, 2018). The hyperparameters are updated by k-fold CV in each rolling window.

My paper introduces two extra contributions to the existing literature. Many previous works focus primarily on improving models with one single technique. For example, Exterkate et al. (2016) concentrate solely on kernel PCA, while Chan (2021) investigates the Minnesota shrinkage prior in depth. In contrast, I sequentially utilise two distinct techniques, PCA and shrinkage methods. Hauzenberger et al. (2023) build their models using a similar logic but with different approaches. They apply horseshoe and Minnesota shrinkage priors after the dimension reduction. My paper, however, extends their work by considering additional regularisation methods mentioned in other works (Coulombe et al., 2021; Kim & Swanson, 2018). Kim & Swanson (2018) only use the most classic penalised regressions and PCA but do not explore kernel PCA or kernel ridge regression. In contrast, my study integrates and compares these advanced techniques.

Moreover, this paper extends the dataset used in the analysis to include the most recent data up to April 2024, in contrast to the work in Hauzenberger et al. (2023), which only includes data up to January 2021. The inclusion of post-Covid data is expected to provide valuable new insights and a more comprehensive understanding of the underlying trends.

3 Data

The data are retrieved from the US Federal Reserve Economic Data (FRED-MD), where monthly updated macroeconomic data can be obtained in real-time (McCracken & Ng, 2016). This database is easily accessible and designed for big data research. I retrieve all available vintages dating from September 1999 to April 2024. Each vintage records a set of covariates beginning from January 1959 until the latest month. More details about the variables are provided in Appendix A. Note that inflation is not directly retrievable and must be calculated using the Consumer Price Index (CPI), and this is discussed later in Section 4.1.

Before transforming the data, several preprocessing steps are necessary. There are two kinds of missing data in general. First, the earliest months do not record all covariates, so any months with missing data are excluded. Therefore, my dataset begins in January 1960. Second, some variables were added or deleted during the historical vintages. To maintain consistency throughout all vintages, the scope of all possible variables is determined by the current vintage (see the table in Appendix A), and there are 103 variables in total.

Moreover, each variable must be transformed to achieve stationarity. The transformation rules are based on the official suggestion (McCracken & Ng, 2016). In addition, all explanatory and dependent variables will be demeaned and standardized to unit sample variance before performing dimension reduction.

After the above procedures, the dimension reduction is conducted, and the number of PCs is set to 5, 10, and 15. Regarding the forecasting model, both inflation and exogenous variables include their lags. In empirical practice, one year of lagged information is included, i.e., 12 lags. Also, the forecast horizon is set to one month and one quarter, i.e. $h \in \{1, 3\}$. The size of the rolling window is set to 240 months (20 years), and the main forecast sample is set as the pre-Covid period.

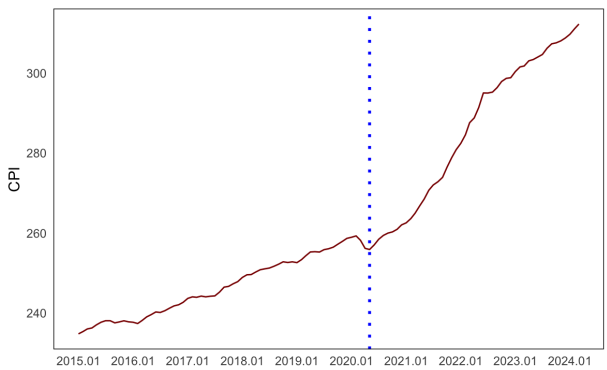

Figure 1: Monthly CPI from 2015 to 2024

The monthly trend of CPI data from 2015 to 2024 is analysed in figure 1 to determine a clear breakpoint regarding the start of COVID-19. As shown in the graph, before May 2020 (indicated by the blue dotted line), the CPI increased relatively smoothly. However, after May 2020, there was a temporary drop followed by a steep increase. Since then, the slope of the trend has remained pronounced in contrast to the pre-COVID period, indicating that the pandemic has a long-lasting effect on the CPI. Therefore, the main model predicts the period from June 2000 to May 2020. Accordingly, the initial window starts in June 1980 and ends in May 2000. The last window starts in May 2000 and ends in April 2020. In total, there are 240 windows.

Another small sample is used to forecast the inflation from June 2020 to December 2023 for further robustness checks, with 43 rolling windows in total. The initial window is from June 2000 to May 2020, and the final one is from December 2003 to November 2023. Here, again, the window size is 240 months.

4 Methodology

4.1 Forecasting Inflation

In this section, I first define the forecasted index, inflation. As suggested by Hauzenberger et al. (2023), the inflation is calculated as

$$y_{t+h} = \log\left(\frac{CPI_{t+h}}{CPI_t}\right) - \log\left(\frac{CPI_t}{CPI_{t-1}}\right). \tag{1}$$

with h indicating the forecast horizon. The forecast model will employ a rolling window forecasting method, with a size of L months. The fully real-time forecasts are benchmarked against the pseudo-real-time forecasts. The former uses data available at a specific point, reflecting real-time conditions, while the latter incorporates subsequent revisions. This comparison can show the impact of data revisions. A generalised h-step ahead forecast for inflation, given the information up until period t, is formulated as follows,

$$Y_{t+h} = u_t'\beta_{t+h} + \epsilon_{t+h}, \quad \epsilon_{t+h} \sim N(0, \sigma^2), \tag{2}$$

where u_t is a set of m variables including latent factors together with lags of y_t at time t. The errors are homoskedastic and the variance is constant over time.

For a detailed specification of u_t', it includes m variables, $m = q + p$, where p represents the number of lagged y_t variables, and q represents the dimension for latent variables obtained by dimension reduction techniques. Suppose the original dataset $X_t = \left(s_{t-p+1}', \ldots, s_t'\right)'$, where s_t is the observation at time t, with k_0 covariates in the given dataset, and X_t contains current s_t and $p-1$ lags of it. Therefore, X_t is composed of K variables with $K = k_0 \times p$. Latent factors are obtained by applying the dimension reduction techniques (see next Section 4.2).

4.2 Dimension Reduction Techniques

Suppose the observed data set X is a $T \times K$ matrix, which contains information of K dimensional variables over T time periods, where $X = \{X_1, \ldots, X_T\}'$, and X_t is the observation at time t, $t = 1, \ldots, T$. To perform dimensional reduction, there exists a function $f(\cdot)$, such that $Z = f(X)$, and $f : \mathbb{R}^{T\times K} \to \mathbb{R}^{T\times q}$. The ultimate goal is to reduce the dimension of X such that the transformed matrix Z is of q dimensions and q is much smaller than K (i.e., $q \ll K$). To achieve this, this paper applies different variants of PCA, including basic linear PCA, squared

PCA, quadratic PCA, and kernel PCA.

Since this paper considers not only linear data transformations but also non-linear ones, X can be possibly mapped to a higher dimension to capture the non-linearities. Let $g(\cdot)$ be the function that maps X to the matrix W, i.e., $W = g(X)$. The resulting matrix W is of dimension $T \times d$, where d is greater than or equal to K. The covariance matrix of W is denoted by Ω, and the function $V(\cdot)$ is used to extract the eigenvectors of Ω, represented as $V(\Omega)$. The general formula for obtaining the final transformed data Z can be written as follows,

$$Z = f(X) = g(X)V(W'W) = WV(\Omega) \tag{3}$$

Note that the eigenvectors are obtained by first applying the singular value decomposition (SVD) of the covariance matrix Ω and then choosing the eigenvectors based on the first top eigenvalues. By defining $g(\cdot)$ and $f(\cdot)$ functions differently, different PCs can be obtained.

4.2.1 Linear, Squared and Quadratic PCs

In the setting of linear PCA, the relationship simplifies to $W = X$, suggesting that g is the identity function. In this case, Ω represents the covariance matrix of X. Essentially, Z is obtained by projecting the original dataset through a linear transformation.

For both squared and quadratic PCs, g becomes a non-linear function. In squared PCs, $g(X) = X \odot X = W$, where $\odot$ denotes the Hadamard product for element-wise multiplication. For simplicity, $X \odot X$ is denoted by X^2. Consequently, Ω then becomes as

$$\Omega = W'W = \left(X^2\right)' \left(X^2\right).$$

Regarding the quadratic PCs, function g maps the data into a new feature space that concatenates original data and the element-wise squared X, i.e. $g(X) = (X, X^2) = W$. The covariance matrix Ω is formulated as

$$\Omega = W'W = \begin{bmatrix} X' \\ (X^2)' \end{bmatrix} \begin{bmatrix} X & X^2 \end{bmatrix} = \begin{bmatrix} X'X & X'X^2 \\ (X^2)'X & (X^2)'X^2 \end{bmatrix},$$

where the W is a $T \times 2K$ matrix, and naturally, the Ω is of dimension $2K \times 2K$.

4.2.2 Kernel PCs

By using a kernel function, the kernel PCA implicitly transforms the data points into a higher-dimensional feature space in a non-linear style. This paper considers two different kernels, the Gaussian kernel and the polynomial kernel.

Similar to the above-discussed approaches, after transforming the data, SVD will be applied to obtain the corresponding eigenvectors, however, since the transformation is undergone implicitly, in this case, we do not calculate the covariance matrix explicitly. Define $\Omega = \kappa$, and the κ is a $K \times K$ kernel matrix, of which each element $\kappa_{i,j} = \kappa(x_{\cdot i}, x_{\cdot j})$. $x_{\cdot i}$ and $x_{\cdot j}$ are the columns of dataset X, with $i, j = 1, \ldots, K$.

The Gaussian kernel function is expressed as,

$$\kappa(x_{\cdot i}, x_{\cdot j}) = \exp\left(-\frac{\|x_{\cdot i} - x_{\cdot j}\|^2}{2\sigma_1^2}\right),$$

where $\|\cdot\|$ is the Euclidean distance function and the σ_1, as a tuning parameter, controls the shape of the Gaussian function. Following Hauzenberger et al. (2023) and Exterkate et al. (2016), σ_1 is set as $\sigma_1 = \sqrt{C_K}/\pi$, with C_K as the 95th percentile of the χ^2 distribution with K degrees of freedom. The polynomial kernel function is expressed as,

$$\kappa(x_{\cdot i}, x_{\cdot j}) = \left(\frac{x'_{\cdot i} x_{\cdot j}}{\sigma_2} + 1\right)^2,$$

with $\sigma_2 = \sqrt{(K+2)/2}$.

4.3 Shrinkage Methods

4.3.1 Adaptive Minnesota Prior

Besides reducing the data dimension, this paper considers different shrinkage methods to improve the computational efficiency further. The first one is the adaptive Minnesota prior. Based on the work from Chan (2021) and Hauzenberger et al. (2023), suppose β as the coefficient vector for the forecasting model described in equation 2, $\beta = (\beta'_1, \ldots, \beta'_m)$, m as the number of regressors. I assume β has a multivariate normal distribution,

$$\beta | V \sim N(0, V).$$

V is a $m \times m$ dimensional prior variance-covariance matrix and it is a diagonal matrix, with each diagonal element denoting as v_j^2. For the adaptive Minnesota prior, the structure of v_j^2 is

$$v_j^2 = \eta_j^2 c_j^2,$$

with

$$\eta_j^2 = \begin{cases} \eta_1^2 & \text{for time-invariant coefficients related to the own lags of the dependent variable (e.g., inflation)} \\ \eta_2^2 & \text{for time-invariant coefficients related to exogenous factors} \\ \eta_3^2 & \text{for state innovation standard deviations,} \end{cases}$$

$$c_j^2 = \begin{cases} \frac{1}{l^2} & \text{for coefficients associated with the own lags of inflation} \quad (l = 1, \ldots, p) \\ \frac{\hat{\sigma}_\pi^2}{\hat{\sigma}_k^2} & \text{for coefficients associated with the } k\text{th exogenous factor} \quad (k = 1, \ldots, q), \end{cases}$$

where η_j^2 is the global shrinkage parameter and c_j^2 is the local scaling parameter. $\hat{\sigma}_\pi^2$ and $\hat{\sigma}_k^2$ denote the OLS variances of an AR(1) model for inflation and for the k-th exogenous factor. Here the local scaling parameters are assumed fixed and known and the global shrinkage parameters follow a hierarchical prior structure and are standard half-Cauchy distributed. Later for the posterior inference, the emphasis is on updating global hyperparameters and their auxiliary quantities using the Markov Chain Monte Carlo simulations.

4.3.2 Penalised Regressions

In addition to the shrinkage prior, several types of penalised regressions are considered: ridge regression, Lasso, Elastic Net, and kernel ridge regression. The first three are widely used to address possible multicollinearity in the high-dimensional data but with slight differences. Ridge regression is an L2 regularisation, which adds a penalty to all coefficients and shrinks them towards zero. Given that U is a $T \times q$ regressor matrix after the data transformation and β is the coefficient matrix as described in equation 2, the loss function of the ridge penalisation is expressed as,

$$\min_{\beta} \left((Y - U\beta)'(Y - U\beta) + \lambda_1 \sum_{j=1}^{q} \beta_j^2 \right). \tag{4}$$

Lasso regression uses L1 regularization, which penalizes the absolute value of the coefficients, potentially driving some of them to exactly zero. The objective function is defined as,

$$\min_{\beta} \left((Y - U\beta)'(Y - U\beta) + \lambda_2 \sum_{j=1}^{q} |\beta_j| \right). \tag{5}$$

Whereas Elastic Net combines both L1 and L2 regularisations, incorporating the benefits of both methods, and its loss function is formulated as,

$$\min_{\beta} \left((Y - U\beta)'(Y - U\beta) + \lambda_3 \sum_{j=1}^{q} |\beta_j| + \lambda_4 \sum_{j=1}^{q} \beta_j^2 \right). \tag{6}$$

Here, λ_1, λ_2, λ_3, λ_4 are tuning parameters that control the strength of the regularisation. These tuning parameters are selected by applying a k-fold CV in each rolling window to ensure the best model performance.

The last penalised regression method is kernel ridge regression. Built upon the ridge function, this method is able to handle non-linearities with the use of kernel trick. Define the Kernel matrix as H and each element $h_{i,j}$ is defined by the Gaussian kernel function,

$$h_{i,j} = h(u_i, u_j) = \exp\left(-\frac{\|u_i - u_j\|^2}{2\sigma^2} \right), \quad i, j = 1, \ldots, T,$$

with u_i and u_j being rows of the regressor matrix U. The objective function of the kernel ridge regression is,

$$\min_{\alpha} \left((Y - H\alpha)'(Y - H\alpha) + \lambda_5 \sum_{i=1}^{T} \sum_{j=1}^{T} \alpha_i h_{ij} \alpha_j \right), \tag{7}$$

and the α is obtained by solving $(H + \lambda I)\alpha = Y$. To make prediction, using new dataset u_{new}, compute the kernel matrix H_{new} between the training data U and u_{new}. The prediction for the new data is then as $\hat{Y}_{\text{new}} = H_{\text{new}}^{\top}\alpha$. Two tuning parameters are involved in the kernel ridge regression, λ_5 being the regularisation parameter and σ controls kernel width. Applying the k-fold CV, these two hyperparameters are selected from grids in each window, $\lambda_5 \in \left\{\frac{1}{8}\lambda_0, \frac{1}{4}\lambda_0, \frac{1}{2}\lambda_0, \lambda_0, 2\lambda_0\right\}$, $\sigma \in \left\{\frac{1}{2}\sigma_0, \sigma_0, 2\sigma_0, 4\sigma_0, 8\sigma_0\right\}$. λ_0 and σ_0 are defined as,

$$\lambda_0 = \frac{1 - \hat{R}^2}{\hat{R}^2}, \quad \sigma_0 = \frac{\sqrt{C_q}}{\pi},$$

where C_q is the 95th percentile of χ^2 distribution with q degrees of freedom. The $\hat{R}^2$ is the R^2 value for the regression of y on the first principle component of the covariate matrix U (Exterkate et al., 2016).

4.4 Model Specification and Evaluation

The models are specified in two ways. The first type of model specification involves transforming the data using different dimension reduction techniques discussed in Section 4.2, followed by including the Minnesota shrinkage prior in the forecast model. In the second type, I use penalised regression techniques after reducing the dimensionality of the data. Both approaches employ dimension reduction to handle high-dimensional data effectively while also incorporating shrinkage methods to shrink large-sized parameters further.

To evaluate the model performance, I benchmark against the AR(p) model, which regresses the dependent variable y on its own lags. The DM test (Diebold & Mariano, 1995) is employed to compare the predictive accuracy between the benchmark and alternative models. Additionally, the RMSE is used as another evaluation metric (see,e.g., Coulombe et al., 2021; Hauzenberger et al., 2023). To assess the robustness of the models against turbulent changes in inflation, the forecast errors are compared across different models specifically after the outbreak of COVID-19.

5 Results

5.1 Forecasts for Pre-Covid period

This section presents the model performance across different specifications. The main focus is to evaluate the prediction accuracy of models that combine various PCA methods and machine learning techniques. As discussed in Section 3, my models first forecast the inflation before June 2020 as the main-sample forecast. There are two different forecast horizons: one month ahead and one quarter ahead. For each forecast horizon, there are two types of forecasts, fully real-time predictions and pseudo-real-time predictions. Fully real-time forecasts use the data available up to the time of the forecast, simulating real-time conditions. In contrast, pseudo-real-time forecasts incorporate revised and more complete data, serving as a benchmark to evaluate the performance of real-time forecasts.

Table 1 displays the RMSEs for the one-month-ahead forecast with real-time data. The first row presents the actual RMSE for the AR models, while the remaining values stand for the relative RMSE of each model compared to the AR model. The asterisks present the p-value from the DM test, showing if the prediction accuracy of the indicated model significantly differs from that of the AR model. Overall, the forecast performance does not improve significantly by utilising different PCA techniques. The DM test shows that there is no notable difference between the AR model and other model specifications, except for the model using linear PCA. The p-value for the model using linear PCA with 15 factors and ridge regression is smaller than 0.05, and the RMSE is higher than that of the AR model, which shows that only applying linear transformations in the dataset could undermine the forecast accuracy.

However, when considering RMSE as the only evaluation metric, the models using 10 or 15 quadratic PCs and Elastic Net outperform the AR model. Their RMSEs are 0.983 and 0.991 times that of the AR model respectively. Furthermore, models that combine kernel PCs with the Minnesota prior or Elastic Net typically perform better. For example, when the model includes 5 Gaussian kernel PCs in the Minnesota prior model, it achieves the lowest RMSE value than using other forms of PCs, which is 0.997 times the value in the AR model.

Table 1: One-month-ahead forecast (real-time)

	MIN	Ridge	Lasso	Elastic Net	Kernel Ridge
AR	1.197	1.210	1.214	1.214	1.240
p = 5					
Linear PCA	1.004	1.055*	1.126	1.076	1.034
Squared PCA	1.070	1.106	1.362	1.255	1.023
Quadratic PCA	1.094	1.077	1.229	1.158	1.026
Gauss. kernel PCA	**0.997**	1.002	1.026	1.015	1.019
Poly. kernel PCA	1.002	1.007	1.102	1.008	1.020
p =10					
Linear PCA	1.002	1.118	1.214	1.151	1.034
Squared PCA	1.301	1.007	1.029	1.040	1.023
Quadratic PCA	1.017	1.004	1.007	**0.983**	1.026
Gauss. kernel PCA	1.002	1.006	1.068	1.009	1.019
Poly. kernel PCA	1.001	1.008	1.056	1.009	1.020
p =15					
Linear PCA	1.001	1.108**	1.043	1.104	1.034
Squared PCA	1.060	1.038	1.030	1.017	1.023
Quadratic PCA	1.070	1.018	1.089	**0.991**	1.026
Gauss. kernel PCA	1.001	1.017	1.181	1.020	1.019
Poly. kernel PCA	**0.999**	1.007	1.022	**0.996**	1.020

Note: The first row presents the actual RMSE for the AR models, while the remaining values represent the relative RMSE of each model compared to the AR model. Asterisks indicate the p-value significance levels for the DM test, with *p<0.1; **p<0.05; ***p<0.01. This shows whether the forecast errors of the AR model and the specified model are statistically different. "MIN" stands for Minnesota prior.

While combining kernel ridge regression with different PCA techniques does not create any models that outperform the benchmark model, it generally yields relatively lower RMSE values compared to models using ridge or Lasso regression. The other two models have slightly higher

RMSEs, but in line with the DM test, they are not dramatically different from the benchmark model. It is also worth mentioning that, though in the DM test, linear PCA performs the worst, it does not always generate the highest RMSE among all PCA methods. For instance, if using the Elastic Net, the model applying linear PCA with 5 factors has a relative RMSE of 1.076 compared to the benchmark, while applying squared PCA with 5 factors has a higher relative RMSE of 1.255, 1.166 times the former.

Next, Table 2 presents the pseudo-one-month-ahead forecast, where the data are revised over certain periods and considered to be more accurate. Again, when examining the DM test results, models using linear PCA combined with ridge regression exhibit significantly larger forecast errors against the AR model. Moreover, if applying the Lasso regularisation, the model uses the Gaussian kernel PCA with 5 latent factors and the model uses the polynomial kernel PCA with 10 factors both have p-values lower than 0.1 in the DM test, suggesting that under these two circumstances, the predictive performance is significantly worse than that of the AR model.

Table 2: One-month-ahead forecast (pseudo)

	MIN	Ridge	Lasso	Elastic Net	Kernel Ridge
AR	1.121	1.133	1.143	1.140	1.144
p = 5					
Linear PCA	1.005	1.063*	1.147	1.088	1.031
Squared PCA	1.079	1.109	1.386	1.271	1.028
Quadratic PCA	1.101	1.078	1.241	1.169	1.031
Gauss. kernel PCA	**0.999**	1.001	1.036*	1.018	1.022
Poly. kernel PCA	1.003	1.007	1.115	1.012	1.023
p = 10					
Linear PCA	1.003	1.133	1.242	1.168	1.031
Squared PCA	1.031	**0.998**	1.018	1.031	1.028
Quadratic PCA	1.016	**0.993**	**0.997**	**0.971**	1.031
Gauss. kernel PCA	1.003	1.007	1.078	1.008	1.022
Poly. kernel PCA	1.002	1.009	1.069*	1.009	1.024
p = 15					
Linear PCA	1.002	1.112**	1.043	1.116	1.031
Squared PCA	1.064	1.030	1.025	1.011	1.028
Quadratic PCA	1.078	1.005	1.094	**0.980**	1.031
Gauss. kernel PCA	1.001	1.011	1.192	1.021	1.022
Poly. kernel PCA	1.000	1.000	1.017	**0.994**	1.024

Note: The first row presents the actual RMSE for the AR models, while the remaining values represent the relative RMSE of each model compared to the AR model. Asterisks indicate the p-value significance levels for the DM test, with *p<0.1; **p<0.05; ***p<0.01. This shows whether the forecast errors of the AR model and the specified model are statistically different. "MIN" stands for the Minnesota prior.

However, in Table 2, more models beat the benchmark than in Table 1. Models using 10 quadratic PCs in the ridge, Lasso or Elastic Net regression achieve a lower RMSE than in the AR model, with values 0.993, 0.997 or 0.971 times that of the benchmark. The model using squared PCA combined with ridge regression outperforms the AR model as well. Moreover, using kernel PCs with the Minnesota prior or Elastic Net also beats the corresponding AR model. In general, the Lasso regression with different types of PCA performs poorer than other models.

Table 3: One-quarter-ahead forecast (real-time)

	MIN	Ridge	Lasso	Elastic Net	Kernel Ridge
AR	1.059	1.080	1.074	1.069	1.181
p = 5					
Linear PCA	1.046**	1.091	1.127**	1.110	1.028
Squared PCA	**0.942**	**0.973**	1.211	1.091	1.049
Quadratic PCA	**0.955**	**0.992**	1.190	1.115	1.050
Gauss. kernel PCA	1.045*	1.023	1.120**	1.047	1.003
Poly. kernel PCA	1.031	1.025	1.122**	1.053	1.002
p =10					
Linear PCA	1.037*	1.156	1.202***	1.181	1.028
Squared PCA	1.025	**0.991**	1.084	1.083	1.049
Quadratic PCA	1.080	1.005	1.145	1.064	1.050
Gauss. kernel PCA	1.035**	1.029	1.170***	1.062	1.003
Poly. kernel PCA	1.030**	1.029	1.181***	1.065	1.002
p =15					
Linear PCA	1.037**	1.195	1.188	1.185	1.028
Squared PCA	1.043	**0.996**	1.061	1.036	1.049
Quadratic PCA	1.069	**0.978**	1.049	**0.985**	1.050
Gauss. kernel PCA	1.036**	1.028	1.184***	1.049	1.003
Poly. kernel PCA	1.028**	1.027	1.148**	1.034	1.002

Note: The first row presents the actual RMSE for the AR models, while the remaining values represent the relative RMSE of each model compared to the AR model. Asterisks indicate the p-value significance levels for the DM test, with $^{*}p<0.1$; $^{**}p<0.05$; $^{***}p<0.01$. This shows whether the forecast errors of the AR model and the specified model are statistically different. "MIN" stands for the Minnesota prior.

Table 3 demonstrates the results for real-time forecasting if the forecast horizon is set to one quarter. The DM tests show that there are more underperformed model specifications against the AR model compared to the one-month-ahead results shown in Table 1. Similar to the one-month-ahead forecast, linear PCA combined with different shrinkage methods results in worse performance against the AR model concerning the DM test. However, kernel PCA methods, including Gaussian kernel PCA and polynomial kernel PCA, combined with the Minnesota prior or Lasso regression also present poorer results. This might be because the one-quarter-ahead forecast introduces more uncertainty and complexity, bringing more challenges for sophisticated models. The AR model, due to its simplicity and robustness, might handle these fluctuations better. In contrast, those more complex models with inappropriate regularisations could overfit the short-term data, making them less effective at capturing longer-term trends and variability.

Regarding RMSE values, the combination of squared PCs with either the Minnesota prior or ridge regression beats the benchmark. For instance, in the Minnesota prior model, applying 5 squared PCs leads to an RMSE value that is 0.942 times that of the AR model. Additionally, the model utilising the Minnesota prior, ridge regression or Elastic Net, combined with quadratic PCA outperforms the AR model when including 5 or 15 latent factors. Comparing the performance between different regularisations, models applying the Minnesota prior have better results, while the Lasso generally leads to poorer performance.

Table 4: One-quarter-ahead forecast (pseudo)

	MIN	Ridge	Lasso	Elastic Net	Kernel Ridge
AR	1.054	1.042	1.056	1.045	1.289
p = 5					
Linear PCA	1.057***	1.103	1.142***	1.123	1.038
Squared PCA	**0.939**	**0.957**	1.201	1.082	1.036
Quadratic PCA	**0.953**	**0.984**	1.184	1.112	1.039
Gauss. kernel PCA	1.056***	1.026	1.134**	1.056	1.035
Polynomial PCA	1.042	1.028	1.136**	1.062	1.035
p =10					
Linear PCA	1.047**	1.174	1.219***	1.200	1.039
Squared PCA	1.022***	**0.986**	1.090	1.088	1.036
Quadratic PCA	1.076	**0.988****	1.134	1.050	1.039
Gauss. kernel PCA	1.045***	1.032	1.186***	1.071	1.035
Polynomial PCA	1.041***	1.033	1.197**	1.074	1.035
p =15					
Linear PCA	1.048**	1.219	1.211***	1.210	1.039
Squared PCA	1.055	1.010	1.082	1.064	1.036
Quadratic PCA	1.085	**0.992**	1.070	1.007	1.039
Gauss. kernel PCA	1.047***	1.031	1.200***	1.058	1.035
Polynomial PCA	1.039***	1.031	1.170***	1.043	1.035

Note: The first row presents the actual RMSE for the AR models, while the remaining values represent the relative RMSE of each model compared to the AR model. Asterisks indicate the p-value significance levels for the DM test, with *p<0.1; **p<0.05; ***p<0.01. This shows whether the forecast errors of the AR model and the specified model are statistically different. "MIN" stands for the Minnesota prior.

As for the pseudo-real-time forecast using the revised data in Table 4, it shares similar patterns with the real-time one. First, comparing the DM test, linear PCA and kernel PCA combined with the Minnesota prior or Lasso regression lead to forecast errors that are significantly different from those of the AR model. Based on their RMSE values, those models underperform the benchmark model. Applying squared or quadratic PCA with the shrinkage methods (the Minnesota prior and ridge regression) performs better than the AR model.

In summary, regarding data transformation methods, quadratic PCA generally produces better forecasting results, especially when combined with the Minnesota prior, ridge, or Elastic Net regularisation. In the real-time one-month-ahead forecast, combining the Minnesota prior or Elastic Net with the kernel PCA also exhibits good performance. However, most of the DM tests are insignificant, even though some models achieve a lower RMSE than that of the benchmark. In this paper, most of the significant DM tests indicate the inferior performance of the suggested model against the AR model. One possible explanation could be that, despite the small forecast errors, the variance of these errors remains large within these models, leading to an insignificant DM score. This highlights a limitation of my models: while they can reduce forecast error magnitude, they do not consistently reduce forecast error variance.

Forecasts using the vintage with revised data have better performance than the real-time one, with more model specifications outperforming the benchmark. This is mainly seen in combinations of squared or quadratic PCA with the Minnesota prior, ridge regression, or Elastic Net. This is expected since the revised data are cleaner and have fewer anomalies, making the prediction more accurate.

Interestingly, our model specifications perform better in the one-quarter ahead forecast than in the one-month ahead forecast in terms of RMSE scores, indicating that some models can handle turbulence in the long forecast horizon. However, if comparing the DM test, in the one-quarter-ahead forecast, more models perform worse than the benchmark model. This inferior performance is particularly noticeable in kernel PCA models with Lasso regression or the Minnesota prior.

Furthermore, upon choosing the proper type of regularisation, the Minnesota prior, ridge regression and Elastic Net are ideal options. Models adopting two less well-performing penalised regressions, Lasso and kernel ridge regression, hardly outperform the benchmark model. Note that while combining PCA with kernel ridge regression underperforms, it is not significantly different from the corresponding AR model based on the DM test. Conversely, especially in the one-quarter-ahead forecast, models using Lasso and PCA significantly underperform the AR model most of the time. This suggests that combining PCA methods with Lasso may be inefficient.

One possible reason could be that the number of PCs extracted is still relatively small, which might not provide enough variability for the Lasso regularisation to be effective. Lasso regularisation tends to perform better when there is a larger set of predictors to select from, as it works by shrinking some coefficients to zero and selecting a subset of variables. With a limited number of PCs, Lasso may not be able to perform this selection effectively, leading to inferior model performance. An alternative explanation could be that Lasso works well with sparse data even with limited data. In this sense, incorporating more latent factors would not help, as Lasso's advantage comes from dealing with sparse data. Hence, if our dataset is not sparse, including more PCs does not necessarily improve the performance either.

5.2 Robustness Check

In this section, I test the robustness of the proposed models to COVID-19 by applying them to forecast inflation after May 2020. Each model includes 15 latent factors and makes one-month-ahead and one-quarter-ahead forecasts respectively.

Figure 2 shows the forecast errors for all model specifications with the forecast horizon of 1. The graph demonstrates the fluctuating trends in the forecast errors from June 2020 to June 2021, especially for models using linear data transformation with ridge, Lasso or Elastic Net regularisation. This volatility is expected due to the outbreak of COVID-19, which introduced many complicated political and financial factors that disrupted the macroeconomic environment, making it more unpredictable. There are also a few noticeable spikes from April 2022 to September 2022. After this period, all models seem more stable, exhibiting smaller variances in forecast errors, and there is no distinct difference between PCA methods regardless of the shrinkage method used.

While all models illustrate a similar trend, there are still some differences. Even though all models are volatile at the start of the pandemic, linear PCA predicts the worst, and the squared PCA also results in large variance in forecast errors. In contrast, kernel PCA has better forecast performance, characterised by more consistent errors. Note that this finding does not apply to models utilising the Minnesota prior or kernel ridge regression, as there is no such significant difference between different PCA methods. In general, models applying different regularised regressions exhibit similar forecast performance. Despite the fluctuating errors, the variance is still considered small, indicating that these models are robust to the turbulent changes during COVID-19.

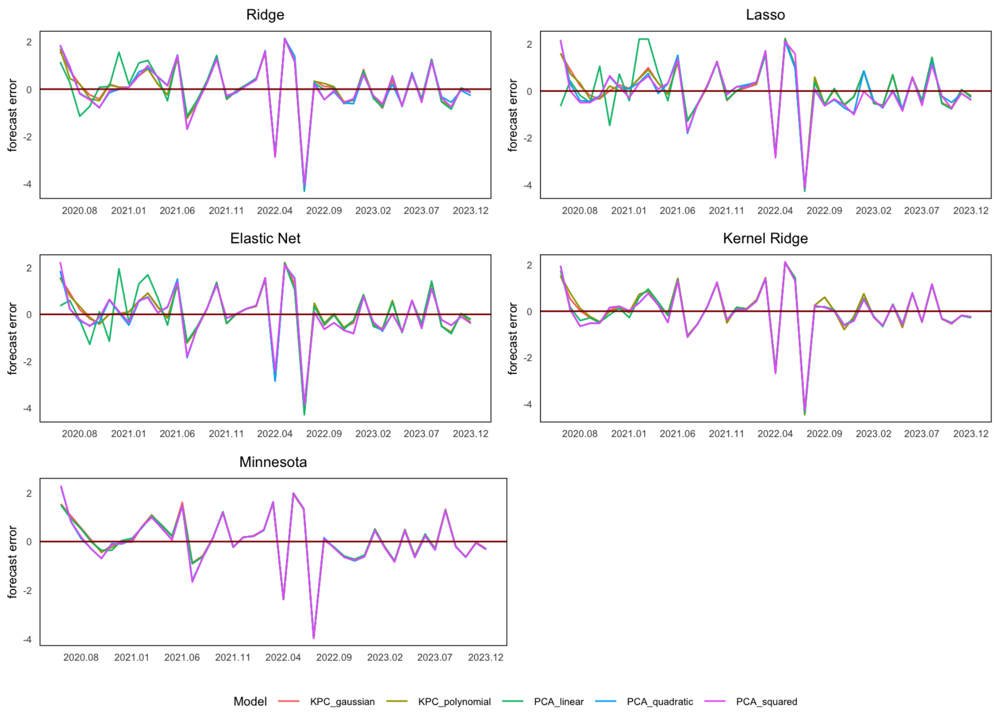

Figure 2: Robustness check for one-month-ahead forecast

The forecast errors become more pronounced in some models when conducting the one-quarter-ahead forecast, as illustrated in Figure 3. Similar to the trend in Figure 2, there are significant fluctuations in the early period (months between June 2020 and June 2021). However, compared to the one-step-ahead prediction, the trends of forecast errors are more distinct across different model specifications. Among all, models with Lasso regression and the Minnesota prior forecast inflation with significantly large forecast errors in the early period, but errors gradually stabilise and converge to zero afterwards. Although models with kernel ridge regression exhibit volatile errors with some peaks, the deviations are relatively small and decrease over time. Elastic Net with PCA methods also shows moderate deviations initially, followed by reducing errors in 2023.

The linear PCA is the least stable, producing a wide range of deviations in forecast errors, making it the least preferred method in the early period. During the same period, the squared PCA and quadratic PCA also result in a large deviation of the forecast errors. This finding is interesting, since in the one-quarter-ahead main-sample forecast, these two PCA methods with the Minnesota prior and ridge regression often beat the AR model. In contrast, two types of Kernel PCA, the Gaussian kernel PCA and the polynomial kernel PCA, lead to more stable and robust models than other PCA techniques. This is especially true during the early period when the overall trend is more turbulent. Last but not least, compared to errors in the one-month-ahead forecast, the ones in the one-quarter-ahead are more consistent after 2023, showing a smoother trend.

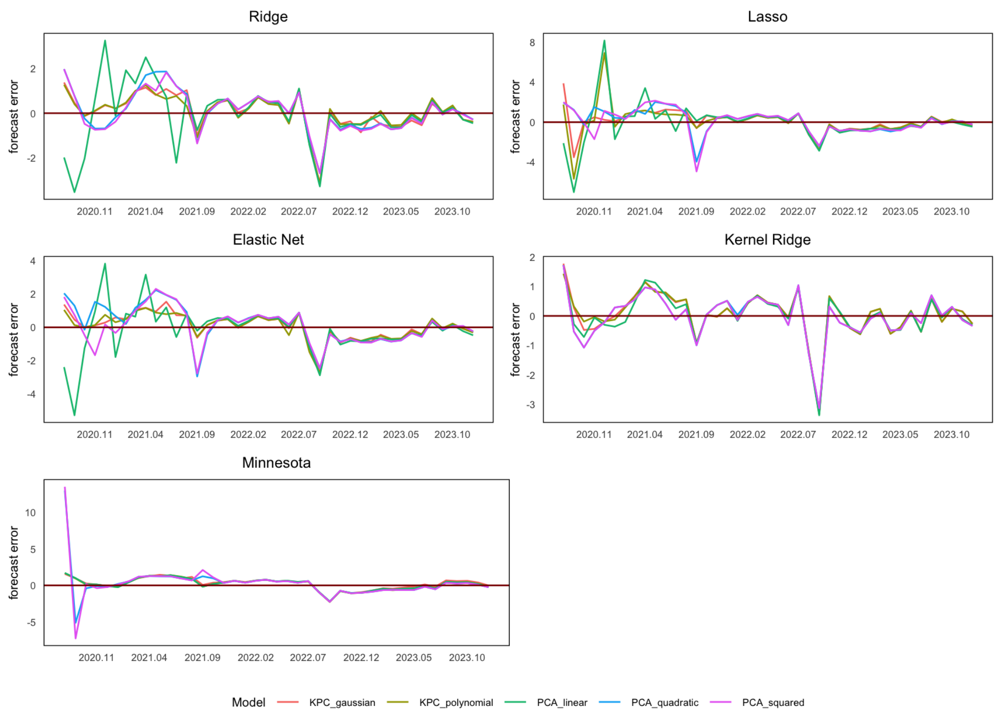

Figure 3: Robustness check for one-quarter-ahead forecast

To sum up, in both one-month ahead and one-quarter-ahead forecasts, results indicate that all models start with less accuracy but stabilise over time. This pattern is characterised by highly volatile forecast errors initially, which converge to zero afterwards. Some specifications are relatively inferior. The squared PCA and linear PCA, whose combinations of either regularisation always lead to a higher variance of forecast errors. In contrast, utilising the kernel PCA could yield better results.

Despite the commonalities between predictions across different forecast horizons, differences still exist. In the one-month-ahead forecast, the deviation of errors is small, making the models robust. However, in the one-quarter-ahead forecast, models using Lasso regression or the Minnesota prior exhibit more volatile forecast errors. This is in line with the main-sample forecast, where most models under these two shrinkage methods underperform the benchmark regarding the DM test. In contrast, models using ridge, Elastic Net, or kernel ridge regression are considered more robust. Regardless of the type of regularisation applied, using PCs with linear, squared, or quadratic transformation undermines the robustness of the model in the one-quarter-ahead forecast.

6 Conclusion

This paper studies the virtues of combining different data reduction techniques and shrinkage methods that could effectively improve real-time inflation forecasting. I use both linear and non-linear PCA to transform data. The non-linear PCA involve squared PCA, quadratic PCA, and kernel PCA (Gaussian and polynomial). This paper also considers different shrinkage methods, the first is the Minnesota shrinkage prior and the second is the penalised regression. The latter applies classic penalisation methods (ridge, Lasso, and Elastic Net regressions) along with the kernel ridge regression, which uses the kernel function to map the original dataset into a higher dimension in a non-linear way. Different combinations provide different model specifications. To evaluate the model performance, I use the DM test and the RMSE metric, with the AR model as the benchmark.

The main-sample forecast uses data before the pandemic. In the real-time one-month-ahead forecast, employing quadratic PCA or kernel PCA with the Minnesota prior or Elastic Net performs better than the AR model. In one-quarter-ahead forecasts, models using squared or quadratic PCs with the Minnesota prior or ridge regularisation outperform the AR model in terms of RMSE. However, many models, especially those using linear PCA or kernel PCA with the Minnesota prior or Lasso regression, underperform according to the DM test, which might be because of increased uncertainty with a larger forecast horizon. The poor performance due to the use of linear PCA aligns with my assumption that the linear data transformation is inferior to the non-linear one when encountering more complex tasks. The robustness check confirms this again by showing the linear PCA with either regularisation method always generates higher-volatile forecast errors.

However, the above findings do not suffice enough to determine the most preferred combinations of dimension reduction techniques and shrinkage methods. In fact, there are no definite answers in this paper. The robustness check shows that including squared or quadratic PCs could lead to poorer forecast performance than other non-linear PCA methods, contradictory to the findings in the main-sample forecast. Also, in the one-quarter-ahead, only the ridge and kernel ridge regressions are considered robust. In this case, during turbulent periods, including kernel PCs with a ridge or kernel ridge regression might be a better choice. Nevertheless, the final decision needs to balance between accuracy and robustness.

Inspired by Hauzenberger et al. (2023), this paper narrows down the scope of dimension reduction techniques by only focusing on different types of PCA, in line with their results, including squared or quadratic PCs in the regression using the Minnesota prior in general outperform other types of PCA. Building upon their findings, this paper demonstrates that by including more shrinkage methods, using Elastic Net regression for the month-ahead forecast or ridge regression for the one-quarter-ahead forecast can also generate results that outperform the AR model, with absolute RMSE values sometimes even lower than those achieved using the Minnesota prior.

Moreover, since I adopt more recent data, the robustness check can use samples starting from and after the pandemic, which provides more insights into the model performance. It is important to note that the winning models in the main-sample forecast are sometimes less robust than others, and this paper suggests some more robust models instead of incorporating the shrinkage prior, among which the ridge and kernel ridge regressions combined with the PCA technique show better performance.

This paper suggests a combination of various types of PCA and kernel ridge regression. Although this model does not prove to be the most efficient in the main-sample forecast, it does show robustness against inflation changes during the pandemic. Further studies can enhance this new model specification by improving the k-fold cross-validation for tuning parameters. Defining a more appropriate search grid for the tuning parameters could enhance performance accuracy. Another approach could be considering other types of PCA, such as sparse PCA, to refine the model.

Moreover, as discussed before in section 5, this study is limited by the fact that most DM tests are not significant even if the specified model achieves lower RMSE than the AR model. One possible explanation is that, while achieving small forecast errors, the variance of these errors remains large within models, resulting in insignificant DM scores. This highlights a limitation of my models: they effectively reduce forecast errors but do not consistently reduce error volatility.

Moving forward, this paper only tests models for forecasting inflation. Other macroeconomic data could also be considered. For example, Kim & Swanson (2018) also forecast the unemployment rate, producer price index, and more. This broader scope could provide a more comprehensive evaluation of the model's performance.

References

Bai, J. & Ng, S. (2006). Evaluating latent and observed factors in macroeconomics and finance. *Journal of Econometrics*, *131*(1-2), 507–537.

Bai, J. & Ng, S. (2008). Forecasting economic time series using targeted predictors. *Journal of Econometrics*, *146*(2), 304–317.

Chan, J. C. (2021). Minnesota-type adaptive hierarchical priors for large bayesian vars. *International Journal of Forecasting*, *37*(3), 1212–1226.

Chan, J. C., Eisenstat, E. & Strachan, R. W. (2020). Reducing the state space dimension in a large tvp-var. *Journal of Econometrics*, *218*(1), 105–118.

Coulombe, P. G., Leroux, M., Stevanovic, D. & Surprenant, S. (2021). Macroeconomic data transformations matter. *International Journal of Forecasting*, *37*(4), 1338–1354.

Diebold, F. X. & Mariano, R. S. (1995). Comparing predictive accuracy. *Journal of Business & Economic Statistics*, *13*(3), 253–263.

Exterkate, P., Groenen, P. J., Heij, C. & van Dijk, D. (2016). Nonlinear forecasting with many predictors using kernel ridge regression. *International Journal of Forecasting*, *32*(3), 736–753.

Giannone, D., Lenza, M. & Primiceri, G. E. (2015). Prior selection for vector autoregressions. *Review of Economics and Statistics*, *97*(2), 436–451.

Goulet Coulombe, P., Leroux, M., Stevanovic, D. & Surprenant, S. (2022). How is machine learning useful for macroeconomic forecasting? *Journal of Applied Econometrics*, *37*(5), 920–964.

Hauzenberger, N., Huber, F. & Klieber, K. (2023). Real-time inflation forecasting using non-linear dimension reduction techniques. *International Journal of Forecasting*, *39*(2), 901–921.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, *24*(6), 417.

Kim, H. H. & Swanson, N. R. (2018). Mining big data using parsimonious factor, machine learning, variable selection and shrinkage methods. *International Journal of Forecasting*, *34*(2), 339–354.

Korobilis, D. & Pettenuzzo, D. (2016). Adaptive minnesota prior for high-dimensional vector autoregressions.

Litterman, R. (1979). *Techniques of forecasting using vector autoregressions* (Working Paper No. 115). Federal Reserve of Minneapolis.

Ludvigson, S. C. & Ng, S. (2009). Macro factors in bond risk premia. *The Review of Financial Studies*, *22*(12), 5027–5067.

McCracken, M. W. & Ng, S. (2016). Fred-md: A monthly database for macroeconomic research. *Journal of Business & Economic Statistics*, *34*(4), 574–589.

Nahil, A. & Lyhyaoui, A. (2018). Short-term stock price forecasting using kernel principal component analysis and support vector machines: the case of casablanca stock exchange. *Procedia Computer Science*, *127*, 161–169.

Schölkopf, B., Smola, A. & Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, *10*(5), 1299–1319.

Sermpinis, G., Stasinakis, C., Theofilatos, K. & Karathanasopoulos, A. (2014). Inflation and unemployment forecasting with genetic support vector regression. *Journal of Forecasting*, *33*(6), 471–487.

Sims, C. A. (1980). Macroeconomics and reality. *Econometrica: journal of the Econometric Society*, 1–48.

Smeekes, S. & Wijler, E. (2018). Macroeconomic forecasting using penalized regression methods. *International journal of forecasting*, *34*(3), 408–430.

Stock, J. H. & Watson, M. W. (1998). *A comparison of linear and nonlinear univariate models for forecasting macroeconomic time series.* National Bureau of Economic Research Cambridge, Mass., USA.

Stock, J. H. & Watson, M. W. (1999). Forecasting inflation. *Journal of monetary economics*, *44*(2), 293–335.

Stock, J. H. & Watson, M. W. (2002). Forecasting using principal components from a large number of predictors. *Journal of the American statistical association*, *97*(460), 1167–1179.

A Description on the dataset

Table 5: Variable Description

id	tcode	variable	description
1	5	RPI	Real Personal Income
2	5	W875RX1	Real personal income ex transfer receipts
3	5	CMRMTSPLx	Real Manu. and Trade Industries Sales
4	5	RETAILx	Retail and Food Services Sales
5	5	INDPRO	IP Index
6	5	IPFPNSS	IP: Final Products and Nonindustrial Supplies
7	5	IPFINAL	IP: Final Products (Market Group)
8	5	IPCONGD	IP: Consumer Goods
9	5	IPMAT	IP: Materials
10	5	IPMANSICS	IP: Manufacturing (SIC)
11	2	CUMFNS	Capacity Utilization: Manufacturing
12	5	CLF16OV	Civilian Labor Force
13	5	CE16OV	Civilian Employment
14	2	UNRATE	Civilian Unemployment Rate
15	2	UEMPMEAN	Average Duration of Unemployment (Weeks)
16	5	UEMPLT5	Civilians Unemployed - Less Than 5 Weeks
17	5	UEMP5TO14	Civilians Unemployed for 5-14 Weeks
18	5	UEMP15OV	Civilians Unemployed - 15 Weeks and Over
19	5	UEMP15T26	Civilians Unemployed for 15-26 Weeks
20	5	UEMP27OV	Civilians Unemployed for 27 Weeks and Over
21	5	CLAIMSx	Initial Claims
22	5	PAYEMS	All Employees: Total nonfarm
23	5	USGOOD	All Employees: Goods-Producing Industries
24	5	CES1021000001	All Employees: Mining and Logging: Mining
25	5	USCONS	All Employees: Construction

id	tcode	variable	description
26	5	MANEMP	All Employees: Manufacturing
27	5	DMANEMP	All Employees: Durable goods
28	5	NDMANEMP	All Employees: Nondurable goods
29	5	SRVPRD	All Employees: Service-Providing Industries
30	5	USWTRADE	All Employees: Wholesale Trade
31	5	USTRADE	All Employees: Retail Trade
32	5	USFIRE	All Employees: Financial Activities
33	5	USGOVT	All Employees: Government
34	1	CES0600000007	Avg Weekly Hours : Goods-Producing
35	2	AWOTMAN	Avg Weekly Overtime Hours : Manufacturing
36	1	AWHMAN	Avg Weekly Hours : Manufacturing
37	4	HOUST	Housing Starts: Total New Privately Owned
38	4	HOUSTNE	Housing Starts, Northeast
39	4	HOUSTMW	Housing Starts, Midwest
40	4	HOUSTS	Housing Starts, South
41	4	HOUSTW	Housing Starts, West
42	4	PERMIT	New Private Housing Permits (SAAR)
43	4	PERMITNE	New Private Housing Permits, Northeast (SAAR)
44	4	PERMITMW	New Private Housing Permits, Midwest (SAAR)
45	4	PERMITS	New Private Housing Permits, South (SAAR)
46	4	PERMITW	New Private Housing Permits, West (SAAR)
47	5	AMDMNOx	New Orders for Durable Goods
48	5	ANDENOx	New Orders for Nondefense Capital Goods
49	5	AMDMUOx	Unfilled Orders for Durable Goods
50	5	BUSINVx	Total Business Inventories
51	2	ISRATIOx	Total Business: Inventories to Sales Ratio
52	6	M1SL	M1 Money Stock
53	6	M2SL	M2 Money Stock
54	5	M2REAL	Real M2 Money Stock
55	6	TOTRESNS	Total Reserves of Depository Institutions
56	7	NONBORRES	Reserves Of Depository Institutions
57	6	BUSLOANS	Commercial and Industrial Loans
58	6	REALLN	Real Estate Loans at All Commercial Banks
59	6	NONREVSL	Total Nonrevolving Credit
60	2	CONSPI	Nonrevolving consumer credit to Personal Income
61	5	S.P 500	S.P's Common Stock Price Index: Composite
62	2	S.P div yield	S.P's Composite Common Stock: Dividend Yield
63	5	S.P PE ratio	S.P's Composite Common Stock: Price-Earnings Ratio
64	2	FEDFUNDS	Effective Federal Funds Rate
65	2	CP3Mx	3-Month AA Financial Commercial Paper Rate
66	2	TB3MS	3-Month Treasury Bill:

id	tcode	variable	description
67	2	TB6MS	6-Month Treasury Bill:
68	2	GS1	1-Year Treasury Rate
69	2	GS5	5-Year Treasury Rate
70	2	GS10	10-Year Treasury Rate
71	2	AAA	Moody's Seasoned Aaa Corporate Bond Yield
72	2	BAA	Moody's Seasoned Baa Corporate Bond Yield
73	1	COMPAPFFx	3-Month Commercial Paper Minus FEDFUNDS
74	1	TB3SMFFM	3-Month Treasury C Minus FEDFUNDS
75	1	TB6SMFFM	6-Month Treasury C Minus FEDFUNDS
76	1	T1YFFM	1-Year Treasury C Minus FEDFUNDS
77	1	T5YFFM	5-Year Treasury C Minus FEDFUNDS
78	1	T10YFFM	10-Year Treasury C Minus FEDFUNDS
79	1	AAAFFM	Moody's Aaa Corporate Bond Minus FEDFUNDS
80	1	BAAFFM	Moody's Baa Corporate Bond Minus FEDFUNDS
81	5	TWEXAFEGSMTHx	Trade Weighted U.S. Dollar Index
82	5	EXSZUSx	Switzerland / U.S. Foreign Exchange Rate
83	5	EXJPUSx	Japan / U.S. Foreign Exchange Rate
84	5	EXUSUKx	U.S. / U.K. Foreign Exchange Rate
85	5	EXCAUSx	Canada / U.S. Foreign Exchange Rate
86	6	OILPRICEx	Crude Oil, spliced WTI and Cushing
87	6	PPICMM	PPI: Metals and metal products:
88	6	CPIAUCSL	CPI : All Items
89	6	CPIAPPSL	CPI : Apparel
90	6	CPITRNSL	CPI : Transportation
91	6	CPIMEDSL	CPI : Medical Care
92	6	CUSR0000SAC	CPI : Commodities
93	6	CUSR0000SAS	CPI : Services
94	6	CPIULFSL	CPI : All Items Less Food
95	6	CUSR0000SA0L5	CPI : All items less medical care
96	6	CES0600000008	Avg Hourly Earnings : Goods-Producing
97	6	CES2000000008	Avg Hourly Earnings : Construction
98	6	CES3000000008	Avg Hourly Earnings : Manufacturing
99	2	UMCSENTx	Consumer Sentiment Index
100	6	MZMSL	MZM Money Stock
101	6	DTCOLNVHFNM	Consumer Motor Vehicle Loans Outstanding
102	6	DTCTHFNM	Total Consumer Loans and Leases Outstanding
103	6	INVEST	Securities in Bank Credit at All Commercial Banks

Note. tcode represents the transformation of the corresponding series: (1) no transformation, (2) Δx_t, (3) $\Delta^2 x_t$, (4) $\log(x_t)$, (5) $\Delta \log(x_t)$, (6) $\Delta^2 \log(x_t)$, and (7) $\Delta \left(\frac{x_t}{x_{t-1}} - 1.0 \right)$.

Adapted Source: US Federal Reserve Economic Data

B Programming code

This section provides a short description of the replication package and instructions on how to use the code.

In the replication package, there are the following R scripts: `Preprocessing`, `estim_file`, `estim_extension`, `data_setup`, `tiv_estim`, `aux_fct`, and `robustness check`. Note that the file `Preprocessing` is credited to Rens den Heeten. Files `estim_file`, `estim_extension`, `data_setup`, `tiv_estim`, and `aux_fct` are adapted from the original replication package provided by Hauzenberger et al. (2023).

1. `Preprocessing`: The first step is to run the `Preprocessing` script. This script cleans the original data and transforms it based on the transformation rules provided by Fred-MD to ensure stationarity. This script prepares a cleaned dataset containing all vintages for further transformation. Note that all historical vintages are downloaded from the official website, as detailed in McCracken & Ng (2016).

2. `estim_file`: To replicate the model mentioned in Hauzenberger et al. (2023), run the main estimation script `estim_file`. After execution, RMSE values and DM test results for different model specifications are obtained. Note that some parameters, such as the forecast horizon, need to be adjusted for different specifications.

3. `estim_extension`: The `estim_extension` script is for running the extension models described in this paper. Again, parameters such as the type of shrinkage method and the forecast horizon need to be adjusted for different models.

4. `robustness check`: This file is to process and plot the results after running the robustness check.

5. `data_setup`: The `data_setup` script is for setting up the data by reducing its dimensionality and deriving the corresponding response and explanatory variables.

6. `tiv_estim`: The `tiv_estim` script is specifically for models using the Minnesota prior.

7. `aux_fct`: The `aux_fct` script contains most of the functions used in other files, including functions for RMSE calculation, ridge regression, kernel ridge regression, etc.

Except for the first four files, other files do not need to be run manually. This overview explains their purpose for simplicity and ensures that users can follow the correct sequence of steps to replicate the results.