A two-step forecasting approach via cluster analysis

A thesis presented for the bachelor degree of
Econometrics & Operations Research
and
Economics & Business Economics

Author
Patrick Sassenus
Supervisor
Prof. Dr. Michel van de Velden
Second Assessor
Dr. Carlo Cavicchia

Erasmus School of Economics
Erasmus University Rotterdam
Netherlands
July 2020

The views stated in this thesis are those of the author and not necessarily those of the supervisor, second assessor, Erasmus School of Economics or Erasmus University Rotterdam.

Abstract

In this research, we propose a novel procedure for forecasting when there is little data available. This method could be interesting after an event that is suspected to have changed the parameters in the model. Our two-step procedure allows for the possibility to only use the most recent data due to information sharing between groups. Our procedure uses cluster analysis to establish these groups among related problem instances. The groups are then jointly estimated with the joint lasso. Which is a penalizing framework containing both a lasso term and a term that penalizes differences between group-specific parameters. Before we started the procedure, we compared several stability algorithms using simulated data. Ultimately, the bootstrap validation algorithm was used as the stability algorithm to estimate the number of clusters k-means should partition the data into. The results acquired show possible gains for using this two-step procedure.

Acknowledgments

The author would like to thank Prof. Dr. Michel van de Velden for supervising this thesis. His constructive comments led to a substantially improved thesis and his advise prevented the author from diverting of a clear path. Furthermore, credits have to be given to Tibshirani and Walther (2005) and Fu and Perry (2020) for making their R packages public which were used by the author. As for all the other methods and algorithms, they were written by the author himself following the cited papers.

Contents

1. Introduction

Structural changes pose a problem for the forecasting abilities of regression models. Disrupting events (e.g. wars, pandemics and economic crises) are certainly not a thing of the past. Globalization and rapid technological advancements, disruptions in themselves, might even increase the rate of this structural instability or at least increase the size of their effect. Even if non-permanent, as (hopefully) with the corona-crises, these time-variabilities are a central issue in econometrics (Greene, 2000).

In this research, we investigate a novel two-step procedure to handle a key issue associated with forecasting after a structural change, namely the lack of sufficient data associated with the new situation. Dondelinger, Mukherjee, and Initiative (2020) propose the joint lasso as a method that allows for information sharing between groups by treating them as related problem instances and estimating the group-specific equations simultaneously. This method, therefore, performs well in small sample sizes. Our two-step procedure uses the joint lasso to increase the forecasting abilities of regression models, however, to be able to use this method, first groups have to be established. This constitutes the first step and involves cluster analysis where we use stability algorithms to determine the number of clusters in the dataset when using k-means. Each cluster will form a group in the joint lasso regression and will thus have the same coefficients. In this way, our two-step approach consists of first determining the number of clusters, 'k', by an algorithm that maximizes the robustness of the clustering (i.e. stability algorithm) and applying k-means with this value for 'k'. Secondly, forecasts are obtained by using the coefficients estimated from the joint lasso model. We use the same data for clustering and estimation of the coefficients to get as much information out of the data as possible to investigate if this enhances the forecasting ability. Thus, we investigate the following research question:

Does our two-step procedure, consisting of clustering and applying the joint lasso method, increase forecasting performance?

In their paper, Dondelinger et al. (2020) mention that they rely on expert knowledge and clinical trials to determine groups, which at times do not lead to clear groupings. This possibly explains the mixed results and why their method does not always outperform other methods such as a pooled regression. By applying cluster analysis we try to extract as much information and patterns from the data. This can lead to better results and be of help in a situation where the theory has not been established or in a social science such as economics; in which consensus sometimes is hard to find, in which the theory is questionable (e.g. rationality) or empirical results do not (fully) support the theory (e.g. international trade theories).

The choice for the clustering method and determining the number of clusters is not a trivial one. In cluster analysis, a major issue is assessing the quality of the outcome. How well the proposed partition fits the data is difficult to assess, as different cluster algorithms give different partitions (none of which have proven to be the 'best') and there is the issue of the number of clusters in which the data should be split (Arbelaitz, Gurrutxaga, Muguerza, PéRez, & Perona, 2013). There has been a lot of research on the quality of cluster analyses and algorithms to select a certain partition 'k' (Milligan & Cooper, 1985; Dubes, 1987; Bezdek, Li, Attikiouzel, & Windham, 1997; Brun et al., 2007; Arbelaitz et al., 2013). They show that there is no one method that always outperforms the rest, it depends on the data and to which set the selection approach belong. For example, selection procedures may focus on within-cluster dispersion relative to an expected distribution, minimizing the distortion (average distance per dimension) or on a measure of cluster stability (Fu & Perry, 2020). Most research, comparing the performance of different algorithms, happens when a new selection algorithm is proposed. This possibly creates a bias in the data selection and the choice of rivalling selection procedures to compare with, such that their proposed method seems more attractive. Arbelaitz et al. (2013) without introducing a new method, perform comparisons to test multiple methods, which suggest which value for 'k' to take when performing, for example, k-means. Yet, they completely leave out algorithms focused on cluster stability, therefore we propose doing their approach on algorithms that use stability as a metric.

By performing a similar study on several algorithms which try to minimize cluster instability and create stable clusters as defined by Wang (2010), researchers can better assess which algorithm to apply if a stability based validation is the appropriate goal. Especially if it turns out, as in the paper by Arbelaitz et al. (2013) on their set of algorithms, which algorithm to use is dependent on the characteristics of the data. This research, can then show which kind of algorithm to apply on what kind of data. We use the results of this comparison part to determine which stability algorithm we should use for our two-step approach.

Our two-step procedure is performed on economic data. The data consists of macro-economic characteristics of the 28 European Union member states from 2000 to 2018. To our limited knowledge there are only a few instances, Vichi and Kiers (2001) is one of them, in which cluster analysis has been applied to cluster countries based upon economic data. We use the data from 2000 to 2016 to estimate the model and use 2017 and 2018 to forecast upon. Since our model is quite large this is equivalent to a situation in which observations are scarce. Our focus is on predicting the economic growth of countries. By comparing the forecasting results to those of other regressions methods, such as the pooled regression and pooled mean group regression, we can assess if the two-step method is appropriate to forecast after a structural change. In such a case one expects the influence of variables to change and one has little data on the new situation (i.e. Brexit, economic crises, pandemics, etc.). This paper is structured as follows; in the next chapter, the various methods and procedures that we use are explained. Afterwards, we discuss in more detail the datasets. In chapter 4, we discuss the results of both the cluster analysis and the two-step approach on the economic data. Lastly, we dedicate a chapter on giving our concluding thoughts, areas of improvement and possible interesting new paths to explore.

2. Methodology

We describe the procedure and methods used in research. We start by going over the evaluation of the stability algorithms for clustering, afterwards, we discuss the procedure applied to the economic data of European Union member states. This, to investigate if clustering in combination with the so-called joint lasso regression can improve forecasting performance compared to pooled regressions.

2.1 Cluster analysis

In cluster analysis, one tries to obtain a set of disjoint clusters or groups based on the features and (hidden) patterns of the data. The objective is to make sure that the observations in one group are similar while those between different partitions are different (Madigan, 2002). Different methods exist to achieve clusters of data, they can be distanced or non-distanced based. In this paper, we focus on k-means, a distance-based method, to cluster our data (Alsabti, Ranka, & Singh, 1997). K-means works by finding the solution to the objective function J:

$$\arg\min_{s} J = \arg\min_{s} \sum_{i=1}^{k} \sum_{x_j \in S_i} \parallel x_j - \mu_i \parallel^2, \tag{2.1}$$

where x_j belongs to the set of observations $(x_1, x_2, ..., x_n)$, S_i belongs to the set of clusters $(S_1, S_2, ..., S_k)$ with $k \leq n$ and μ_i is the center chosen for cluster i. The objective function is minimized for given k, thus the number of clusters k is fixed and needs to be determined either by the investigator's (prior) knowledge or by an algorithm (e.g. a stability algorithm in our case). Equation 2.1 finds k points (cluster center) and assigns each observation to the closest point of these k points such that the total squared euclidean distance between the observations and their cluster center is minimized. See algorithm 1 for a description of the algorithm that finds a solution for J.

Finding the minimum value of J is dependent on the initialization values, it is common to do several replications, each with different values for the starting centers and then choose the clustering that gives the lowest objective value. The solution the algorithm provides is dependent on the initialization values because it can get stuck in a local minimum instead of the global minimum that we look for. Wang (2010) suggests using 20 random starts, for our simulated data we use 30 different restarts and for our economic data 50. To further reduce the possibility to get stuck in a local optimum instead of finding the global optimum. The difference (30 versus 50) is due to the many datasets and algorithms to be tested in our simulation and thus the computation time would get too large.

As stated, the number of clusters (k) needs to be specified beforehand. Our cluster analysis focuses on evaluating the performance of stability based algorithms to determine the value of 'k'. With this, we want to gain insight in the different algorithms and be able to justify the choice of a particular stability based algorithm in combination with k-means to partition our economic data.

We evaluate five different stability algorithms, which are mentioned and discussed at the end of this section. We draw random samples from the dataset using a chosen method (e.g. k-means), the clustering is called more stable if the partitions change less from one sample to another than of another partitioning (Wang, 2010). Thus, it is a measure of cluster robustness produced by a method over the randomness in a sample. A robust cluster should only contain observations that always appear together regardless of the clustering analysis.

To gain the aforementioned insight we apply the stability algorithms in combination with k-means to cluster simulated data (detailed description of the data can be found in the data chapter). The stability algorithm suggests a value for 'k' in k-means, to evaluate the suggestion by each stability algorithm a metric has to be chosen. Normally, the correct value for 'k', the correct number of clusters, is compared with the value proposed by the algorithm and it is considered a success if the algorithm chooses this correct 'k' (Wang, 2010; Fu & Perry, 2020; Fang & Wang, 2012; Tibshirani & Walther, 2005). Thus, it is considered a success if the number of clusters into which the data needs to be clustered as proposed by the stability algorithm matches the number of correct number of clusters. The advantage of using simulated data is that beforehand it is known what the actual clustering should be and thus also what the correct value for 'k' is. However, the limitation of this measure of success is that the correct number of clusters does not always align with the 'best' partition (Figure 2.1). Which is the partition that matches the correct partition the most (following the construction of the simulation) (Arbelaitz et al., 2013). Therefore, Arbelaitz et al. (2013) proposes a partition similarity measure, the one we use is the Adjusted Rand. And so the stability algorithm has correctly partitioned the data in our approach when it chooses the value 'k' with the greatest Adjusted Rand. The Rand index measures the extent of agreement between two clusterings, where an agreement is seen as two observations being assigned to the same cluster or different clusters in both clusterings. The Adjusted Rand index corrects the original Rand index for chance (Hubert & Arabie, 1985):

$$\frac{Rand - Expected\ Rand}{Maximum\ Rand - Expected\ Rand}, \tag{2.2}$$

where Rand stands for the Rand index and is calculated as:

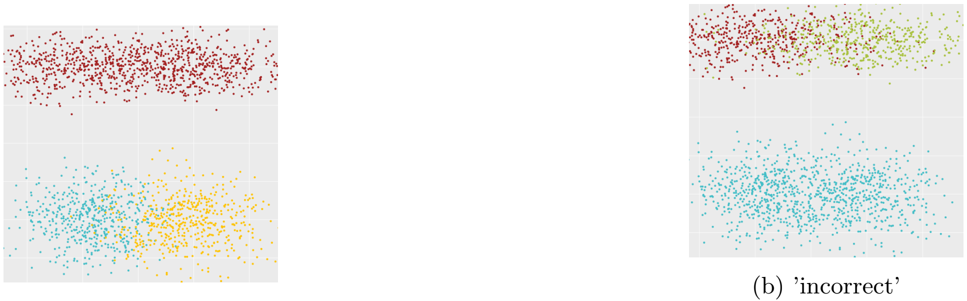
$$RI = \frac{X_{ss} + X_{dd}}{X_{ss} + X_{dd} + X_{sd} + X_{ds}}, \tag{2.3}$$

here there are two partitions and $X_{ss}, X_{dd}, X_{sd}, X_{ds}$ are the number of pairs of a set that are in the same subsets in both partitions, different subsets in both partitions, in the same subset in the first partition but not in the second and in different subsets in the first partition but in the same subset for the second partition, respectively.

2.1.1 Stability

Before we delve into the five stability clustering algorithms that we compare, it is first necessary to more precisely define clustering stability. Given a set of observations $\mathbf{X} = \{x_1, x_2, ..., x_n\}$ residing in a p-dimensional space, $\mathbf{X} \in \mathbb{R}^p$, a cluster algorithm, $\Psi(\mathbf{X}, k)$, creates a mapping $\psi : \mathbf{X} \mapsto \{1, 2, ..., k\}$ with $k \geq 2$. The distance between two clusters $\psi_1(\mathbf{X})$ and $\psi_2(\mathbf{X})$ can be calculated with:

$$d(\psi_1(\mathbf{X}), \psi_2(\mathbf{X})) = \mathbb{P}(\psi_1(X) = \psi_1(Y), \psi_2(X) \neq \psi_2(Y)) + \mathbb{P}(\psi_1(X) \neq \psi_1(Y), \psi_2(X) = \psi_2(Y)) \tag{2.4}$$



(a) 'correct'
This is the original: two overlapping circular clusters and one elongated cluster.

(b) 'incorrect'
Same cluster figure partitioned differently, due to the overlapping being considered as one cluster

Figure 2.1: Two different partitions into three clusters
(a), is seen as correct as the data is simulated by two spheres (blue and yellow) which overlap and one ellipse (red). It, thus, consists of three bodies (e.g. clusters). (b) applies a cluster algorithm to divide the data into three clusters. However it splits the ellipse and combines the overlapping spheres. Many observations are now taken together that should not be together or are split while you want them to be seen as the same. Splitting into two cluster (ellipse as one cluster and overlapping sphere as the other) would cause observations to be taken together that you might not want but at least observations that do belong together will be taken together.

where X and Y are independently drawn from **X**, hence this distance can be seen as the probability of disagreement between the two clusters. Using this definition of distance the instability can be calculated as:

$$s(\Psi, k, n) = \mathbb{E}[d\{\Psi(Z^n; k), \Psi(Z^{n*}; k)\}], \tag{2.5}$$

where $\Psi(\cdot, k)$ is applied to two independent samples Z^n and Z^n* of size n from **X** and the expectation is taken with respect to these samples.

This metric for stability is small for a stable clustering ($s(\Psi, k, n)$ lies between 0 and 1). An incorrect number for the amount of clusters leads to an unstable clustering. With a value for 'k' larger than the correct 'k', the 'true' clusters split into smaller ones that change between samples. For example, one of the 'true' clusters is forcefully split into two clusters, while there is no pattern that determines which of the two new clusters an observation belongs. In this case, an observation will randomly be assigned to one of the new clusters. A similar issue holds for a 'k' smaller than the correct 'k' since true clusters get clumped together and this most likely also changes from one sample to another. Even in asymptotically large samples where convergence happens to a stable clustering, the rate of conversion is a tell since clustering algorithms with the correct value for 'k' converges faster as shown by Wang (2010).

2.1.2 Prediction strength algorithm

One of the first algorithms associated with cluster stability is called the Prediction Strength algorithm (Ps) by Tibshirani and Walther (2005). The method uses the following principle: first, divide the data into a training set and a test set to be divided into 'k' clusters; second apply the cluster method to the training set (e.g. k-means: $\Psi(Z_{training}; k)$); then apply the same method to the test set ($\Psi(Z_{test}; k)$), lastly the established cluster centers from $\Psi(Z_{training}; k)$ are used to assign the test data to the training clusters (an observation is assigned to the cluster which center is most near).

Now define an $n \times n$ matrix $(C[\Psi(Z_{training}; k), Z_{test}])$, where n is the amount of observations in Z_{test}. Entry ii' is 1 if observation i and observation i' are in the same cluster in both the prediction clustering as well as the test clustering. Thus, for each pair of observation in the test set we determine if the training clustering puts them in the same cluster if the test cluster did so as well. Putting this together Tibshirani and Walther (2005) gives us the following metric:

$$Ps(k) = \min_{1 \leq j \leq k} \frac{1}{n_j(n_j - 1)} \sum_{i \neq i' \in A_j} C[\Psi(Z_{training}; k), Z_{test}]_{ii'}, \tag{2.6}$$

with k being the number of clusters, A_j being cluster j and n_j being the amount of observations inside cluster A_j. Thus, the prediction strength is the minimum proportion of observation pairs inside a test cluster that are also assigned to the same cluster by the training cluster centers over each test cluster. From experiments Tibshirani and Walther (2005) suggests taking the largest $k*$ over a set of 'k' values such that Ps is above the 0.8-0.9 threshold of a scale from 0 to 1.

The final thing to be mentioned is how the training and test set should be chosen from the data. The method chosen for this is cross-validation, more specifically two-fold cross-validation to avoid bias and overfitting on the training set. R-fold cross-validation entails that the data is split into r equally sized proportions, of which r-1 proportions (or blocks) are used as training set and the remaining r'th block is used as test test. Every proportion is once used as test set, while the remaining are used as training set, doing this ensures that the results are not biased due to the chosen block. Of all the combinations of training/test blocks the results are gathered and combined and will count as Ps for that particular value of 'k'.

2.1.3 Cross-Validation algorithms

The two Cross-Validation algorithms by Wang (2010) are called Cross-Validation with voting (CV_v) and Cross-Validation with averaging (CV_a) (Wang, 2010). Instead of using r-fold cross-validation they are based on a leave-many-out cross-validation method (which in this case is a set of splittings with the same splitting ratio) and it works with an estimated version of equation 2.5. The data is split into three sets: two equally sized training sets and one test set. Then the two training sets are used to construct two clusterings via $\Psi(X_{t1}; k)$ and $\Psi(X_{t2}; k)$ (k-means in our case). These two clusterings are used to predict the test set and then the similarity distance (equation 2.5) between these two predictions is measured.

Method for CV_v (Wang, 2010):

1. Permutate the data $\mathbf{X} = \{x_1, ..., x_n\}$ into $\mathbf{Z}^{*c} = \{x_1^{*c}, ..., x_n^{*c}\}$

2. Split $\mathbf{Z}^{*c}$ into three parts with m, m and $n - 2m$ observations, respectively (we set $m = n/3$): $\mathbf{Z}_1^{*c} = \{x_1^{*c}, ..., x_m^{*c}\}$, $\mathbf{Z}_2^{*c} = \{x_{m+1}^{*c}, ..., x_{2m}^{*c}\}$ and $\mathbf{Z}_3^{*c} = \{x_{2m+1}^{*c}, ..., x_n^{*c}\}$

3. $V_{ij}^{*c}(\Psi, k, \mathbf{Z}_1^{*c}, \mathbf{Z}_2^{*c}) = I[I[\psi_1^{*c}(x_i^{*c}) = \psi_1^{*c}(x_j^{*c})] + I[\psi_2^{*c}(x_i^{*c}) = \psi_2^{*c}(x_j^{*c})] = 1]$
   here $\psi_1^{*c} = \Psi(\mathbf{Z}_1^{*c}; k)$ and $\psi_2^{*c} = \Psi(\mathbf{Z}_2^{*c}; k)$. This equation gives 1 if the two clustering do not agree on whether x_i^{*c} and x_j^{*c} should be in the same cluster (i.e. instability).

4. Now the estimation of equation 2.5 based on this c'th permutation becomes:

$$\hat{s}^{*c}(\Psi, k, m) = \sum_{j=i+1}^{n} \sum_{i=2m+1}^{j} V_{ij}^{*c}(\Psi, k, \mathbf{Z}_1^{*c}, \mathbf{Z}_2^{*c}) \qquad k = 2, ..., K$$

5. Compute

$$\hat{k}^{*c} = \underset{2 \leq k \leq K}{\arg\min}\, \hat{s}^{*c}(\Psi, k, m)$$

6. Repeat steps 1-5 for $c = 1, ..., C$ and compute $\hat{k}$ as the mode of $\{\hat{k}^{*1}, ..., \hat{k}^{*C}\}$.

Method for CV_a (Wang, 2010):

1. Step 1-4 are the same as in CV_v
2. Repeat steps 1-4 for $c = 1, ..., C$ and compute

$$\hat{s}^{(}\Psi, k, m) = \frac{\sum_{c=1}^{C} \hat{s}^{*c}(\Psi, k, m)}{C}$$

3. Compute

$$\hat{k}^{*c} = \underset{2 \leq k \leq K}{\arg\min}\, \hat{s}(\Psi, k, m)$$

.

Thus, with voting the value for 'k' is determined by the k that most of the time gives the least amount of instability, while for averaging it is determined by the k that over all permutations gives the lowest average instability. In both cases when there is a tie in step 5 or 6 of CV_v or in step 6 of CV_a the largest value of 'k' is chosen as tie-breaker. For our research we set the number of permutations (C) equal to 25.

2.1.4 Bootstrap validation algorithm

The method proposed by Wang (2010) was supposed to result in better cluster partitions (i.e. more stable) and be computationally faster than previous suggested cross-validation algorithms by Wang (2010). However, CV_v and CV_a split the data into three sets, two training sets and one test set. The use of only one-third of the dataset to train k-means causes inefficiencies, thus a new variant was proposed that relies on bootstrapping (Fang & Wang, 2012). Bootstrapping relies on drawing with replacement from the sample to create more simulated samples without knowledge of the underlying distribution. This way the training set is of the same size as of the original dataset. Thus, instead of taking permutations and splitting the data, observations are randomly drawn (with replacement) from the original dataset until there are two additional datasets with the same length as the original dataset.

Bootstrap Validation algorithm (Bv):

1. Generate B independent bootstrap sample-pairs $(\mathbf{X}_{b1}, \mathbf{X}_{b2})$ with $b = (1, ...B)$. Each sample consists of n observations drawn with replacement from the original dataset of size n.

2. Create the clusterings $\Psi(\mathbf{X}_{b1};k)$ and $\Psi(\mathbf{X}_{b1};k)$ for all $b = (1, ...B)$

3. For each bootstrap sample-pair calculate:

$$\hat{s}_b\left(\Psi(\mathbf{X}_{b1};k), \Psi(\mathbf{X}_{b2};k)\right) =$$

$$\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\left|I\left\{\Psi_{X_{b1},k}\left(x_i\right) = \Psi_{X_{b1},k}\left(x_j\right)\right\} - I\left\{\Psi_{\bar{X}_{b2},k}\left(x_i\right) = \Psi_{\bar{X}_{b2},k}\left(x_j\right)\right\}\right|$$

4. Compute the cluster instability as

$$\hat{S}_B(\Psi, k, n) = \frac{1}{B}\sum_{b=1}^{B}\hat{s}_b\left(\Psi(\mathbf{X}_{b1};k), \Psi(\mathbf{X}_{b2};k)\right)$$

5. Compute

$$\hat{k} = \underset{2\leq k\leq K}{\arg\min}\,\hat{s}(\Psi, k, n)$$

Fang and Wang (2012) propose to increase the set of values 'k', (k = 1,...,K), if the algorithm in step 5 suggests to take 'K' as the value for $\hat{k}$.

2.1.5 Gabriel cross-validation algorithm

The Gabriel cross-validation algorithm (Gc) is based on a form of cross-validation in which both the columns and rows are permuted (Fu & Perry, 2020). In the stability algorithms, clusters are considered useful if in multiple independent samples from the same population the clustering arises. The Gc uses a different approach, but is still inspired by the stability algorithms and thus we gather it under the stability algorithms. The stability comes from using both the dimensions of an observation and the cross-validation of the observations themselves to create meaningful clusters that manifest themselves in independent samples.

Method for Gc:

1. Divide the sample matrix into a

$$\mathcal{X} = \begin{bmatrix} X_{train} & Y_{train} \\ X_{test} & Y_{test} \end{bmatrix}$$

$$\text{where } X_{train} \in \mathbb{R}^{n\times p}, Y_{train} \in \mathbb{R}^{n\times q}, X_{test} \in \mathbb{R}^{m\times p}, \text{ and } Y_{test} \in \mathbb{R}^{m\times q}$$

   Using a $K \times L$ fold with $K = 5$ and $L = 2$

2. Cluster the rows of Y_{train} with k-means yielding 'k' cluster centers

3. Use the rows of X_{train} as predictors and the cluster labels of the rows of Y_{train} as corresponding labels. Use these pairs to fit a model that can predict the label of the rows of X_{train}. This can be done by computing the mean value of X for each class; we assign an observation to class 'i' if that class has the closest mean.

4. Apply this model to the rows of X_{test}, thus for every row you obtain a label. Then assign these labels to the rows of Y_{test}.

5. Each row of Y_{test} is now related to a cluster center $\mu_{\hat{Y}i}$ where i is the row of Y_{test}. Calculate the metric Gc(k) as:

$$Gc(k) = \frac{1}{m} \sum_{i=n+1}^{n+m} (Y_i - \mu_{\hat{Y}i})^2$$

The average Gc(k) over the folds will be calculated and the value of 'k' corresponds to the one minimizing the average value. In the case of a tie, choose the smallest 'k'.

Thus, to summarize our approach: we apply k-means to simulated data with 'k' ranging from 2 to 10. Then we let the 5 stability algorithms determine which value for 'k' they propose. An algorithm will be counted as a success if it chooses the 'k' that maximizes the Adjusted Rand. The simulated data consists of 640 different datasets, which contain 64 combinations of six different characteristics and each combination has 10 replications. The exact details and explanations of this data can be found in the data section. Based on the results the algorithm which has the most successes is chosen as algorithm to determine the amount of clusters for the economic data.

2.2 Regressions & Forecasting

This section explains the methods and modelling strategy for the economic data. The goal of this section is to gauge the effect clustering has (via k-means and the stability algorithm) on the performance of forecasting via the joint lasso regression (Dondelinger et al., 2020) when there is a limited amount of observations. We compare the forecasting performance of this joint lasso regression with the traditional methods used on this kind of dataset. We first explain the methods used and how they are implemented before we delve into the procedure used.

2.2.1 Pooled regression

The data that we use consists of European Union member states in the period 2000-2017. This means it consists of 28 countries (cross-sectional data, denoted by subscript 'i') and 18 years (time serial data, denoted by subscript 't') of which the years 2016 and 2017 will be left out and be forecasted upon. Data with both a cross-sectional and time serial component are called panel data. In our panel data, we use growth in GDP per capita as the dependent variable ($\Delta \ln(y)_{i,t}$). Where Δ stands for the first difference and y stands for GDP per capita. The independent variables are GDP per capita in the previous year ($\ln(y)_{i,t-1}$), capital accumulation ($\ln(Kap)_{i,t}$), human capital ($\ln(H)_{i,t}$), population growth ($Pop_{i,t}$), annual inflation percentage ($Fl_{i,t}$), general government revenue ($\ln(Govrev)_{i,t}$), general government consumption expenditure ($\ln(Govexp)_{i,t}$), government capital formation ($\ln(Govcap)_{i,t}$), $R\&D$ expenditure ($\ln(RD)_{i,t}$), trade exposure ($Txp_{i,t}$), and a time trend (T). For more details on how these variables are actually defined and measured we refer to the data chapter.

One way to deal with panel data is to ignore the time dimension and view all observations as part of one big cross-section. In the style of Bassanini and Scarpetta (2002) we achieve the following model:

$$\begin{aligned}\Delta\ln(y)_{i,t} = {} & \beta_0 + \beta_1\Delta\ln(Kap)_{i,t} + \beta_2\Delta\ln(H)_{i,t} + \beta_3\Delta Pop_{i,t} + \beta_4\Delta Fl_{i,t} \\ & + \beta_5\Delta\ln(Govrev)_{i,t} + \beta_6\Delta\ln(Govexp)_{i,t} + \beta_7\Delta\ln(Govcap)_{i,t} \\ & + \beta_8\Delta\ln(RD)_{i,t} + \beta_9\Delta Txp_{i,t} + \theta_1\ln(y)_{i,t-1} + \theta_2\ln(Kap)_{i,t} \\ & + \theta_3\ln(H)_{i,t} + \theta_4 Pop_{i,t} + \theta_5 Fl_{i,t} + \theta_6\ln(Govrev)_{i,t} + \theta_7\ln(Govexp)_{i,t} \\ & + +\theta_8\ln(Govcap)_{i,t} + \theta_9\ln(RD)_{i,t} + \theta_{10}Txp_{i,t} + \theta_{11}T + \epsilon_{i,t},\end{aligned} \tag{2.7}$$

the Δ sign in front of variables means that the first difference was taken. In this model the usual OLS properties hold, the estimators are unbiased and consistent. The advantage of using this model is that in the case of limited observations, which we try to investigate, all the observations are 'pooled' together to estimate the parameters. In other words, these observations have the same coefficient. However, the downside is that all the coefficients are the same for all countries and across time. This is not very realistic, thus (Bassanini & Scarpetta, 2002) proposes the pooled mean group estimation (*PMG*).

2.2.2 Pooled mean group estimation

PMG is a regression model that allows for the short-run coefficients to be different per country while imposing the long-run coefficients to be equal for each country. This model thus fits economic theory which tells that there is a common (conditional) steady-state to which economies converge in the long-run. Another advantage of the model is the gained efficiency in small sample size since outliers have limited influence on the countries coefficients in contrast to a model in which all coefficients are country-specific (Bassanini & Scarpetta, 2002). This leads to the following equation:

$$\begin{aligned}\Delta\ln(y)_{i,t} \quad = \quad & \left.\begin{aligned}& \beta_0 + \beta_1\Delta\ln(Kap)_{i,t} + \beta_2\Delta\ln(H)_{i,t} + \beta_3\Delta Pop_{i,t} \\ & +\beta_4\Delta Fl_{i,t} + \beta_5\Delta\ln(Govrev)_{i,t} + \beta_6\Delta\ln(Govexp)_{i,t} + \beta_7\Delta\ln(Govcap)_{i,t} \\ & +\beta_8\Delta\ln(RD)_{i,t} + \beta_9\Delta Txp_{i,t}\end{aligned}\right\}\text{short-run} \\ & \left.\begin{aligned}& +\theta_1\ln(y)_{i,t-1} + \theta_2\ln(Kap)_{i,t} + \theta_3\ln(H)_{i,t} + \theta_4 Pop_{i,t} + \theta_5 Fl_{i,t} \\ & +\theta_6\ln(Govrev)_{i,t} + \theta_7\ln(Govexp)_{i,t} + \theta_8\ln(Govcap)_{i,t} \\ & +\theta_9\ln(RD)_{i,t} + \theta_{10}Txp_{i,t} + \theta_{11}T + \epsilon_{i,t}.\end{aligned}\right\}\text{long-run}\end{aligned} \tag{2.8}$$

The PMG allows for short-run coefficients, intercepts and the error variances to differ across the countries. It could also allow for a convergence parameter to differ across countries as in the paper of Bassanini and Scarpetta (2002). We, however, choose the convergence parameter to be equal across the countries considering that the European Union member states are more similar and according to economic theory should converge quicker than the OECD countries. The faster and similar convergence can be attributed to an intensive intra-trade, an overarching governmental body, a common currency for most countries (even in the case of countries with a different currency there is still a peg or tie to the Euro) and a high level of (economic) integration. The common convergence parameter also allows for easier implementation of the lasso and joint lasso.

2.2.3 Lasso and joint lasso

The lasso regression is a method that introduces a bit of bias into the estimation of the parameters in order to reduce the overfitting and enhance the forecasting ability (Tibshirani, 1996). In machine learning this is called the bias versus variance trade-off, because you fit the training data less well (higher bias) in order to reduce the error between the test data and the predicted values for the test data (lower variance). Another benefit of using a lasso term is that it allows for variable selection because it is able to put the value of coefficients to zero if their effect is small. The closely related ridge regression (another regularization technique) lets such coefficients asymptotically shrink to zero but they will never actually get to zero. The lasso form is:

$$\hat{B} = \arg \min_{\beta \in \mathbb{R}^p} \left\{ \frac{1}{N} \|y - X\beta\|_2^2 + \lambda \|\beta\|_1 \right\} \tag{2.9}$$

where $\|\cdot\|_q$ denotes the l_q-norm [1] and λ is a tuning parameter which follows from cross-validation. The larger the λ the larger the bias, will be and thus the poorer the training sample fit. Furthermore, if λ becomes bigger the parameters, which do not add (much) to the prediction of the dependent variable go to zero. The downside of the lasso regression (i.e. including a lasso term) is that obtaining the standard errors and thus determining the statistical significance of the coefficients is still an unresolved issue (Tibshirani, Hoefling, Wang, & Witten, 2011).

The lasso forms the basis for the joint lasso, this technique allows groups of observations to be pooled together to increase the predictive ability and the efficiency to estimate the coefficients, whilst at the same time introducing similarity instead of equality between the coefficients of groups (Dondelinger et al., 2020). By jointly estimating the groups this way, information between groups can be shared. This approach is beneficial for problems in which the sample size per group is small, a high number of coefficients have to be estimated and/or one expects group-specific coefficients to be similar but not identical. In our case, we can view countries that are clustered together as one group and since they are all EU member states we expect the coefficients do not differ greatly between groups. Thus the introduction of a penalty to obtain similarity between groups makes sense. For example, the effect a growth in population for Poland will not be the same as for the Netherlands on the growth of the economy but one does not expect a huge difference, especially not considering the free movement of labour in the EU. The joint lasso is of the form:

$$\hat{B} = \underset{B=|\beta_1 \cdots \beta_k|}{\arg\min} \sum_{k=1}^{K} \left\{ \frac{1}{n_k} \|y_k - X_k\beta_k\|_2^2 + \lambda \|\beta_k\|_1 + \gamma \sum_{k'>k} \tau_{k,k'} \|\beta_k - \beta_{k'}\|_2^2 \right\} \tag{2.10}$$

,

in which $\tau_{k,k'}$, λ and γ are called tuning parameters. As can be seen, λ and γ are considered to be equal across the groups whilst τ differs and can be changed to determine the degree of similarity between certain subgroups. The λ and γ are determined by cross-validation, whilst $\tau_{k,k'}$ is set at unity as done by Dondelinger et al. (2020). The index 'k' refers to the different clusters in our setting.

2.2.4 Procedure & implementation

To test if the joint lasso can improve the forecasting performance on this economic dataset, we first use k-means in combination with the 'best' stability method chosen from the previous section to get

[1] $\|(x_1, x_2, \ldots, x_n)\|_q = (|x_1|^q + |x_2|^q + \ldots + |x_n|^q)^{1/q}$

meaningful (in our case stable) groupings. The clusters proposed by the stability algorithm forms the K groupings in the joint lasso regression. The forecasting performance is measured by the mean standard prediction error (*MSPE*) over the period 2017-2018 and compared with the forecasting performance of the pooled regression (equation 2.7), *PMG* (equation 2.8) and the joint lasso for K equals two. The reason behind the pooled regression (no groupings) is that Dondelinger et al. (2020) show that their method has mixed results and does not always beat the pooled regression. While we choose *MSPE* because it fits economic theory and it is tailored for this dataset, also if the joint lasso terms are added then it is the same as seeing every country as a grouping. Moreover, by choosing K=2 we evaluate if the joint lasso improves forecasting performance but that the groupings chosen by the stability algorithm are not useful. The choice for K=2 is because this is the simplest partition (ignoring the trivial K=1) and results, most likely (due to using mostly economic parameters) in a division between rich and poor countries which is an intuitive partition. Lastly, we would like to stress that besides the clustering into two clusters (K=2) and the number of clusters proposed by the chosen stability algorithm, we also automatically include no groupings (K=1) and all countries belonging to a separate group (K=28) by using the pooled regression and *PMG*, respectively.

The implementation is based on the optimization method of Dondelinger et al. (2020), however, to be applicable to our work it has to be augmented. In our research, some coefficients do not change per country (long-run coefficients) and thus only part of the coefficients need to obtain the joint lasso penalty (the third part of equation 2.10). We start by first neglecting the long-run part such that we get an equation that is similar to the one of Dondelinger et al. (2020). If we define X_{sdiag} as a block-diagonal $n \times pK$ matrix with X_{sk} along the diagonals we get the short-run equation (equation 2.8). Here X_{sk} is the 'kth' group of short-run observations, n stands for the total number of observations, p for the number of short-run coefficients and K for the number of groups. Thus, we obtain $y_s - X_{sdiag} b_s$ with:

$$X_{\text{sdiag}} = \begin{pmatrix} X_{s1} & & \\ & \ddots & \\ & & X_{sK} \end{pmatrix} \quad b_{\text{s}} = \begin{pmatrix} b_{s1} \\ \vdots \\ b_{sK} \end{pmatrix} \quad y_{\text{s}} = \begin{pmatrix} y_1 \\ \vdots \\ y_{K}. \end{pmatrix} \tag{2.11}$$

We define the l_2 penalty in such a way that we can move it into the squared term. The Γ matrix is a $pK(K-1)/2 \times pk$ matrix containing the pair-wise constraints belonging to a $pK(K-1)/2 \times 1$ vector of zeros ($\overrightarrow{0}$). Γ consists of p-row blocks $\Gamma_{k,k'}, k, k' \in [1, K], k < k'$ that entail the constraint between two coefficient vectors β_{sk} and $\beta_{sk'}$. $\Gamma_{k,k'}$ is filled with[2]:

$$\Gamma_{k,k'}(l,m) = \begin{cases} \gamma\tau_{k,k'} & \text{if } l = -p(k-1) + m \\ -\gamma\tau_{k,k'} & \text{if } l = -p\,(k'-1) + m \\ 0 & \text{otherwise}\ . \end{cases} \tag{2.12}$$

We now obtain: $\hat{b}_s^{aug} = \underset{b_s^{aug}}{\arg\min} \left\| y_s^{aug} - X_{sdiag}^{aug} b_s^{aug} \right\|_2^2 + \lambda \left\| b_s^{aug} \right\|_1$, with

$$X_{\text{sdiag}}^{\text{aug}} = \begin{pmatrix} X_{\text{sdiag}} \\ \Gamma \end{pmatrix} \quad y_{\text{s}}^{\text{aug}} = \begin{pmatrix} y_{\text{s}} \\ \overrightarrow{0} \end{pmatrix}\ . \tag{2.13}$$

[2] There seems to a mistake in the way Dondelinger et al. (2020) defines $\Gamma_{k,k'}$. Fortunately, they made their package available for us to check and it seems that the matrix should indeed be filled as we propose. Thus, their package fills it the same way as mentioned here and not as defined in their paper.

Lastly, we incorporate the long-run variables without the l_2 penalty, but they can have a lasso term. To do this we introduce the $n \times m$ matrix X_l in which m are the long-run coefficients and the rows are in the same order as the rows of X_{sdiag}. Furthermore, we need the b_l $m \times 1$ vector of long-run coefficients. Now we can use the glmnet package with:

$$X_{\text{final}} = \begin{pmatrix} X_l & X_{sdiag} \\ 0 & \Gamma \end{pmatrix} \quad b_{\text{s}}^{final} = \begin{pmatrix} b_l \\ b_s \end{pmatrix} \quad y_{\text{s}}^{\text{aug}} = \begin{pmatrix} y_{\text{s}} \\ \vec{0} \end{pmatrix}, \tag{2.14}$$

for it is now a classic lasso problem.

3. Data

In this chapter, we describe the data used for our experimental setups. First, the simulated datasets which are used to evaluate the different stability algorithms are covered. Afterwards, the economic data are specified, these are used to cluster the countries using k-means with the help of the stability algorithm. As previously mentioned, the stability algorithm to be used is determined by evaluating the results from the simulated dataset.

3.1 Simulated data

A synthetic dataset is chosen since it has several advantages over real data: characteristics can easily be controlled by the researcher, the correct partition is known and independent of the knowledge of the researcher. Of equal importance is the fact that many of the real datasets used for evaluation of cluster analysis are designed for supervised learning, and thus are not always adapted for unsupervised techniques such as clustering. This leads to problems such as identifying the correct partition (Arbelaitz et al., 2013).

The simulated data follows the design of Arbelaitz et al. (2013), however, some changes are made (an extra characteristic added being the largest), all of which are discussed. The characteristics are: the number of clusters (K), number of explanatory features of a datapoint (*Exp*), degree of cluster overlap (*Ov*), ratio of the density of the first cluster with respect to the other clusters (*Den*), level of noise (*Lon*) and extra noise-dimensions of a datapoint (*Dim*). *Lon* represent observations that contain cluster related variables, however, the observations do not belong to any cluster (e.g. the variables are measured wrongly). Whereas *Dim* represents variables that are not cluster related.

In the original setup, K and *Exp* have three possible states and the rest have two, however, in our setup, all characteristics have two possible states (Table 3.1). Since all possible combinations of the characteristics will be used, the addition of the characteristic *Dim* quickly leads to a combinatorial explosion if some characteristics have three possible states. As all possible configurations will be randomly created 10 times, there will be 640 datasets.

With these six characteristics, the common aspects and problems of datasets and clustering are examined. Especially, the amount of cluster overlap is a problem for clustering analyses and can

Table 3.1: Characteristics and possible states

Characteristic	Possible States
n_{min}	100
K	2, 4
Exp	2, 8
Ov	1.5 (strict), 5 (bounded)
Den	1, 4
Lon	0, 0.1
Dim	0, 2

lead to incorrect partitions (most common is that the overlapping clusters are counted as one or are separated incorrectly). By setting strict *Ov* we make sure that the overlap distance is strictly adhered meaning every cluster overlaps with at least one other cluster. While using a bounded *Ov* ensures there is a maximum allowed overlap. The level of *Den* allows for one cluster to have significantly more observations than the rest, due to the approximate cluster volume to be the same but the densities being different. This difference can cause trouble when clustering with k-means (Raykov, Boukouvalas, Baig, & Little, 2016). *Lon* determines if the simulated data should contain any noise which in this case refers to data points which do not belong to any cluster and may represent errors induced while gathering or processing the data. *Dim* is added as a characteristic and determines if the data is captured in a subspace of the total amount of dimensions (i.e. variables are included that do not contain any information or are not cluster related). This characterizes a common phenomenon in econometrics, namely that insignificant explanatory variables are included. Lastly, by allowing for different values of K and *Exp*, we examine if the mutual relation between these two characteristics matters (smaller, equal or greater). Thus, with all possible configurations, we can examine which stability algorithm combined with k-means generally leads to the best outcome and if there are subsets in which a certain stability algorithm achieves superior results. This allows for adjusting the combination based on the characteristics of the dataset to be used. For example, the noise in social sciences is generally larger than in the natural sciences.

The procedure to create the clusters follows Arbelaitz et al. (2013) which we now outline. First, we define a sampling window that is used for all the datasets: a hypercube contained in the coordinates [0,0,0,0,...,0] and [50,50,50,50,...,50]. Within this hypercube we define a reduced sampling window ([3,3,3,3,...,3] and [47,47,47,47,...,47]) and in this reduced sampling window, all the cluster centers are drawn with a uniform distribution. The data points belonging to the same cluster are drawn from a multivariate normal distribution with its respective cluster center as mean and the identity matrix as a covariance matrix.

The first cluster, with center c_1, will have $n_{min} * den$ data points and creates a density asymmetry if $den \neq 1$. Since all the remaining clusters will have n_{min} points, while all clusters have approximately the same volume. Afterwards for the other $K - 1$ clusters, a center c_i is drawn, this cluster center should adhere to the following property: $\parallel c_i - c_j \parallel \geq 2 * Ov \quad \forall c_i \neq c_j$, else a new center should be drawn. In the case of a strict overlap, on top of the former restriction, another restriction should hold, namely, a random cluster center that already exists should be chosen randomly and the distance to this center should be $2 * Ov$. After all the cluster have been created, the noise is created with a uniform distribution inside the sampling window and the number of points being $Lon * n_{min} * (Den + K - 1)$.

Then if $Dim \neq 0$, for every point, an extra dimensions is created (thus the hypercube space is expanded by the number of extra dimensions). These extra dimensions are drawn from the exponential distribution, with $\lambda = 10$ If the coordinate of the new dimension is outside the reduced sampling window then a new coordinate has to be drawn. Finally, if any point of any cluster is outside the sampling window a new point has to be drawn from the multivariate normal distribution belonging to that cluster.

3.2 Economic data

The dataset which is used for the regressions and forecasting analysis, after first being clustered, is now discussed in detail. As the model is, at its core, inspired by Bassanini and Scarpetta (2002), consequently so are the variables. This means that the variables are a combination of the determinants of the Solow growth model and policy-related variables. The data are gathered of the 28 European Union member states (United Kingdom was still part of the EU during the time period under investigation) from 2000 to 2018. There are differences between the data used by Bassanini and Scarpetta (2002) and the ones in this paper (the different time period and countries is a trivially obvious one). The differences are due to: 1. availability of better proxies for the variables, 2. lack of the data for all of the EU member states and 3. a selection had to be made, for Bassanini and Scarpetta (2002) also do not use all the variables at once. Rather they chose subsets at a time, thus avoiding multicollinearity issues. The selection was made based upon their most significant findings and economic theory (Burda & Wyplosz, 2013). Now follows a discussion of the variables and the most paramount differences.

3.2.1 Solow growth model variables

In the model of Solow, economic growth as the dependent variable is linked to three main factors: capital accumulation, population size and technological advancement. In our model, the dependent variable, economic growth, is proxied by the growth in real GDP per capita. The relation with capital accumulation follows from one of Kaldor's stylized facts which links output per hour to capital per hour (Burda & Wyplosz, 2013). This lead to the neoclassical theory that the expansion of production capacity increases output. In our study, the ratio of real private physical capital formation to real GDP (i.e. private investment share) acts as the propensity to accumulate fixed capital. From this same neoclassical theory, it follows that a steady-state transition happens due to the law of diminishing returns kicking in. One of the factors allowing for a sustained long-run growth in both the capital stock and output is the growth in the employed labour force. We measure this with the percentage growth in population between the ages of 15 and 64. The other ingredient allowing for this long-run growth is the technological progress, on top of that, it is the factor that allows for the permanent growth in per capita output. We decompose technological progress into an increase in knowledge and innovations, the former is proxied by the average number of years of schooling of the population age 24-64 (human capital) while the latter is estimated by gross expenditure on R&D as a percentage of GDP.

It is worth noting that some of the proxies are rather crude and critique on the accounting of these variables have been made. For instance when it comes to calculating the GDP and growth in GDP, a rise in these does not automatically mean a richer country or even an increase in output and some strange phenomenon can occur due to the mix of globalization,(intangible) resource allocation and accounting rules as is seen by the rise of Ireland's GDP by 26% in 2015 (OECD, 2016). Another example of a crude and narrow proxy is that of human capital since it does not take into account other dimensions of human development nor the quality of the education. However, the way it is obtained is less crude than that of Bassanini and Scarpetta (2002). For they approximate the number of years of education by first dividing every countries education system into three levels, use the cumulative years to obtain the education level and multiply this by the ratio of the population that falls into one of these three levels. While our research directly takes mean years of schooling as input. For details on the source of the variables see appendix B.

3.2.2 Policy variables

The Solow growth model advocates convergence of countries in terms of wealth and growth, however, the evidence for this unconditional convergence is lacking. Instead, the model has been adjusted that supports conditional convergence (Bassanini & Scarpetta, 2002). By controlling for policy variables and including initial conditions, conditional convergence can be achieved. We propose the following variables for the conditions: GDP per capita in the previous year, a country-specific constant and trade exposure (*Txp*). The trade exposure is translated into a weighted average of export exposure and import penetration adjusted for country size. First, the export exposure (*Ex*) is measured by the export to GDP and import penetration (*Mp*) by import to consumption penetration (domestic production minus exports plus imports). Then, the trade exposure is measured as $Ex+(1-Ex)*Mp$ (Bassanini & Scarpetta, 2002). Lastly, this trade exposure is regressed on the population size of the countries and the residuals are then used as *Txp*. This way a measure independent of country size is obtained.

General government revenue to GDP (*Govrev*), the ratio of general government final consumption expenditure to GDP (*Govexp*) and government capital formation (*Govcap*, proxied by government real fixed capital accumulation to GDP) are measures that relate to government size and financing. In economics, it is known that the size of government matters since the government can provide public goods, taxes and subsidies can distort the functioning of the market and government loans and investments can cause the crowding-out effect. Furthermore, positive external effects such a spillover can be under the social optimum due to individual agents having a utility function different from that of society as a whole. The government can then step in and provide incentives to achieve this social optimal level (in case of negative external effects the reverse will hold). These factors thus influence the market and economic growth. Naturally, the way it is financed (tax versus non-tax revenue or even direct or indirect tax receipts) also matters, however due to the lack of data for all the countries and for all the years this could not be included. Measures of financial development in the form of market capitalization and credit deposits by banks to the private sector as a percentage of GDP are included by Bassanini and Scarpetta (2002), however Leahy, Schich, Wehinger, Pelgrin, and Thorgeirsson (2001) conclude that these measures are too crude, the effect too small and that they mostly work indirectly via capital investments (which are already included) on the growth of a country. Thus we leave them out.

The final variable to be included is the annual percentage of the GDP deflator as a measure for inflation (*Fl*). The level and stability of inflation affect many economic processes, if it has a low level and it is stable it will reduce economic uncertainty and raise the efficiency of the price mechanisms. Also, due to the strong tie with interest rates it has an effect on the willingness of companies to take risks and invest in long-run investment decisions.

Bassanini and Scarpetta (2002) in their paper take the natural logarithm of each variable, this is however, not possible for some variables (trade exposure, inflation and population growth) are not always positive. One can argue that in the countries that they observe and for their time-span they are always positive, however trade exposure can never be always positive for every country (it is easy to show that some residuals have to be negative). Thus, either they made a mistake, took the absolute value without mentioning it or used complex numbers. In our case, we will take the natural logarithm for every variable except the above three mentioned. As mentioned earlier, we investigate the time period 2000 to 2018 but a few things are noteworthy. Firstly, as we take the first difference for the short-run part of the equations (see equation 2.8) we have one-time observation less. Secondly, we take 2000-2016 as the hold-out or training set and 2017-2018 as the test to evaluate the forecasting performance of the regressions. Since our method is a two-step procedure to increase the forecasting performance (first clustering and then applying the joint lasso method) we will also only use 2000-2015 as the data for clustering. Lastly, when it comes to preprocessing the data for cluster analysis there is no consensus on the procedure. Milligan and Cooper (1988) advocate normalization before performing a cluster analysis. Therefore, we perform cluster analysis on the normalized data combined with principal component analysis. Milligan and Cooper (1988) proposes the following normalization method,

$$Z = \frac{X - Min(X)}{Max(X) - Min(X)} \tag{3.1}$$

For the normalized data, we let the chosen stability algorithm choose the 'k' in k-means for ten replications each and then make a decision which cluster partition (which 'k' in k-means) to use based on which 'k' value is recommended most often.

4. Results

We first discuss the results of the simulation study and then proceed to the results of the two-step procedure.

4.1 Simulation results

Figure 4.1 shows the total number of correct proposals (according to the rand Index) for 'k' in k-means by each algorithm as a ratio of the total number of simulations. As mentioned earlier, since we have all combinations of six characteristics, each characteristic has two states and each combination is replicated ten times, we get a total number of 640 simulations. Clearly, the two methods designed by Wang (2010) together with the method designed by Fang and Wang (2012) perform significantly better. These three methods (Bv, CV_a, CV_v) outperforming the Ps -method is logical in the sense that this is one of the first stability algorithms and proposing a new algorithm only makes sense if it contains an advantage. The advantage here is clearly its effectiveness (Wang, 2010). A major surprise to us is the relatively bad performance of Gc, not only has it been introduced very recently. But in the paper Fu and Perry (2020) show very good results for this method, hence we considered it as a possible candidate to cluster our economic data and thus we included it in our evaluation.

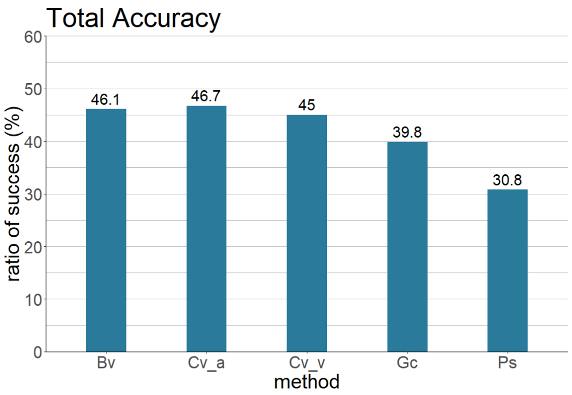
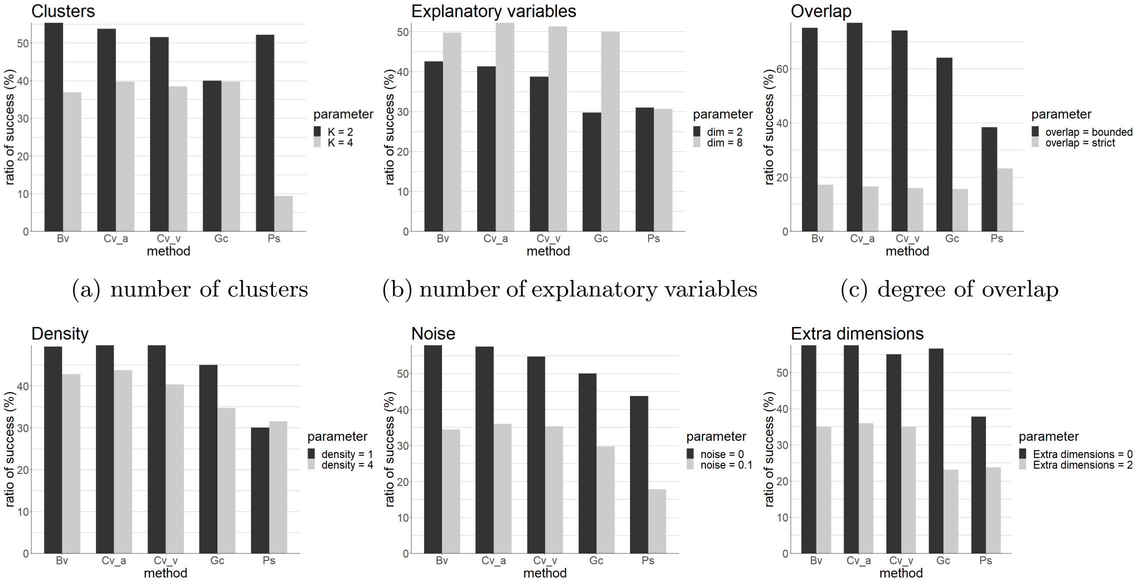


Figure 4.1: Overall results where the percentage of success rate is highlighted

To make sure we did not make a mistake in our programming/implementation, we redid the procedure using the R package of the paper. We got similar results and only a small improvement when using their correlation correction version (these are also the results we publish here). Further investigation shows that the Gc does perform well with the simple simulations. For instance when there is limited to no overlap or when the amount of clusters is small relative to the amount of explanatory dimensions. The results quickly deteriorate when the datasets become more complex, more so than the other algorithms. For example, when two characteristics are combined (noise and strict overlap) it rarely correctly picks the correct partition. While we do use the Rand Index to measure a success and not the number of simulated clusters as Fu and Perry (2020) does, which does drop the rate of success (see Appendix C), it does not explain the underperformance. When we count a success as finding the number of clusters corresponding to the simulated number of clusters, then still it underperforms by about 30% on average. One of the reasons is that the algorithm uses a more complex relationship to determine clusters. It looks for patterns and relations between the different explanatory variables and not just between the different observations within one explanatory variable. Since in our simulation we use the identity matrix for the covariance structure, there is no such relation between the variables. Another possible explanation is that Fu and Perry (2020) proposes to break ties by choosing the smallest 'k'. While it is difficult to retrieve the intermediate steps with the R package, we see that in some instances the largest 'k' proposed by the algorithm was correct. CV_a and CV_v propose selecting the largest number of clusters in case of a tie and this procedure in many ties meant selecting the correct partition.

Of course, the results for CV_a and CV_v are not even close to the 100 % success rate reported in Wang (2010). However, that is solely due to our more complex simulated examples. The examples used by Wang (2010), especially regarding the distance-based simulations, are quite simple and more of a toy example. When we focus solely on the same kind of dataset (no overlap, no noise, no density difference, the number of explanatory variables equal or higher than the number of clusters and only the possibility of extra noise-dimensions is included), then we also get a perfect result. According to Figure 4.1, CV_a has the best overall score and similar to the best performing algorithms in Arbelaitz et al. (2013). While a one-to-one comparison is not possible due to our slightly different simulations it does give an indication. Namely, CV_a is on par and could even be better than their best-tested method.



(a) number of clusters (b) number of explanatory variables (c) degree of overlap

(d) density (dis)similarity (e) level of noise (f) number of noise dimensions

Figure 4.2: Success rate broken down by characteristics

In Figure 4.2 the success rate is broken down into subsets. For each characteristic, we display what happens if only the value for that characteristic is changed while we keep the other characteristics fixed. From, Figure 4.2 we clearly see that all algorithms have trouble when there is strict overlap (Figure 4.2c). The difference between the two states of this characteristic is also the largest for all algorithms except for Ps which has the largest gap in success rate when the number of clusters is varied. Across all characteristics and broken down into their states, we can see that CV_a has the highest number of successes in almost all the subsets. The cases when it is not the 'best' is usually when it is beaten by Bv (Figure 4.2b, dim = 2), but in such cases, it is still the second 'best'. Only when there are four clusters (Figure 4.2a) or when the overlap is strict (Figure 4.2c) is neither Bv nor CV_a on top. Lastly, we want to mention that the prediction: CV_a will outperform Bv when the sample gets larger, by Fang and Wang (2012) seems to hold. In their simple simulations Bv slightly outperforms CV_a, which they suspect comes from the small number of observations per cluster.

4.2 Two step procedure

As CV_a performs best overall and in the majority of subsets, we choose this stability algorithm to determine the number of clusters. The performance of the algorithms in the categories concerning a high number of clusters, a large number of explanatory variables, strict overlap, a difference in density and extra dimensions were especially looked at (Figure 4.2). We expect that these features are in our economic data, for example, the countries in the EU have to adhere to strict rules before even considered to enter the EU, thus they are already similar and the policies from the member states and the EU try to close this (economic and political) distance even further. In other words, we suspect that countries in different clusters still share a lot of similarities. Likewise, we expect clusters to differ in the number of countries due to countries widely differing in the number of years they are a member. Some countries like Belgium are more integrated and dependent on other countries then newer members or member that have not adopted the Euro. While we take extra dimension into extra account since we do not know if all variables are cluster related (i.e. will help in finding clusters). Finally, we looked at the high number of clusters because we are already including the situation with only two clusters.

CV_a outperforms the other algorithms in all of these categories except for the strict overlap. Here Ps outperforms the other algorithms by quite a bit, however since the Ps-algorithm performs poorly in the other situations we choose the CV_a-algorithm. The values for 'k' range from 2 to 9 and CV_a then recommends 9 as the value for 'k'. When CV_a recommends the largest value for 'k' that you allow for, Fang and Wang (2012) suggest increasing the range of 'k'. However, this is not possible since the observations (28 countries) are randomly assigned to three different samples of equal size by the algorithm. Thus, 'k' cannot have a value greater than 9. This leads us to use Bv as the stability algorithm. Bv can compute the stability for values of 'k' up to the number of observations and it is the second-best algorithm overall (Figure 4.1). We increase the values 'k' can assume from 9 to 20, Table 4.1 shows the results. Since the algorithm suggests 'k' = 15 the majority of the time, we cluster the countries over 15 groups.

Table 4.1: Ten replications of Bv over the macro-economic data

Value for 'k'	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Number of times suggested	0	0	0	0	0	0	0	0	0	0	0	0	2	7	1	0	0	0	0

Thus, we apply k-means to our economic data with 'k' = 15 as part of our two-step procedure and 'k' = 2 for comparison. Partitioning the normalized data in 15 cluster leads to the groups in Table 4.2. While we do not mean to analyze the clustering, we do note some strange groups. Such as Cyprus and Spain being grouped or Malta, Portugal and Slovenia in one group. On the other hand, there are groups that make sense especially historically and geographically which naturally lead to similar economies and economic policies. For example, Belgium and Germany or Latvia, Lithuania and Poland or Bulgaria and Romania. Lastly, Austria and Finland or Ireland, Netherlands and Sweden while geographically far apart, in terms of the EU terms, they can be justified as their governments on many occasions band together on EU level signalling their similar interests and policies (also in the case of the latter group they are often criticized for being tax havens).

When we apply k-means with 'k' = 2 to the same data we achieve the split in Table 4.3. Interestingly there seems an East-West divide instead of the infamous North-South divide in the European Union when it comes to their economies and governmental policies (Appendix D). Also, Spain and Portugal do not belong to this richer and more developed first class. This makes sense since both countries were hit hard by the financial crises and seem to struggle more still by the after-affects than countries like Ireland.

Table 4.4 shows the results of the forecasts. We notice that going from pooled regression to the PMG (i.e. country-specific short-run coefficients) decreases the forecasting performance. This is in line with our suspicion that there are not enough observations per country to estimate accurately. When we split the countries into two groups and incorporate the joint lasso terms we get a similar forecasting performance as the pooled regression. However, when we apply the

Table 4.2: EU member states into fifteen class memberships

First class	Austria, Finland	Ninth class	Ireland, Netherlands, Sweden
Second class	Latvia, Lithuania, Poland	Tenth class	Italy
Third class	Estonia, Slovakia	Eleventh class	France, United Kingdom
Fourth class	Luxembourg	Twelfth class	Croatia, Hungary
Fifth class	Cyprus, Spain	Thirteenth class	Belgium, Germany
Sixth class	Bulgaria, Romania	Fourteenth class	Greece
Seventh class	Malta, Portugal, Slovenia	Fifteenth class	Czech Republic
Eight class	Denmark		

Table 4.3: EU member states into two class memberships

First class	Austria, Belgium, Denmark, Estonia, Finland, France, Germany, Ireland, Italy, Luxembourg, Netherlands, Sweden, United Kingdom
Second class	Bulgaria, Croatia, Cyprus, Czech Republic, Greece, Hungary, Latvia, Lithuania, Malta, Poland, Portugal, Slovenia, Slovakia, Spain

two-step procedure and thus increase the number of groups to 15 we do get an improvement in the forecasting performance. Thus, our two-step procedure performs best in terms of forecasting.

Table 4.4: Forecast results

	Pooled regression	PMG	Joint lasso (two groups)	Two-step procedure
Number of clusters	1	28	2	15
Lasso term (λ)	-	-	$1.46 * 10^{-4}$	$1.86 * 10^{-5}$
Joint lasso penalty (γ)	-	-	0.1	2.0
MSPE	0.0161	0.0233	0.0161	0.0121
Improvement wrt pooled regression (%)	0	-44.5	0.12	24.9
Standardized RMSPE	1.25	1.50	1.25	1.08

The RMSPE is the root of the MSPE as such the scale is equal to that of the dependent variable. We divide this by the standard error of the dependent variable (in the forecasting period) to obtain a standardized version.

5. Conclusion & Discussion

Our goal was to increase the forecasting performance of economic models when there is little data. For example, the data needed to forecast is only recently being measured or collected. Or a disrupting event has happened which is likely to change the estimators and/or the model. In which case the desire is to not use the data from before the event or at least limit the need for it. Our work is inspired by Dondelinger et al. (2020) in which a high-dimensional regression is applied to group structured data. We used their joint lasso regression to increase the forecasting performance of a model used to determine the economic growth of countries.

As the method requires to group the data beforehand, we had to establish a group structure. We decided to use k-means in combination with a stability algorithm to cluster the countries. To decide upon the stability algorithm to use, we first evaluated five methods and then made our decision based on the results. No algorithm outperformed the other algorithms in every possible combination of the six characteristics. However, the cross-validation with averaging algorithm performed best in most of our simulated cases. This algorithm was used as an initial attempt to cluster the data. The algorithm could only cluster the data into a maximum of 9 clusters due to the limited sample size. As it suggested this maximum number of clusters there were probably more than 9 clusters. This forced us, to move onto the bootstrap validation algorithm, which performed second-best in most of our simulated sets.

Seven out of ten times, the bootstrap validation algorithm suggested partitioning the countries in 15 clusters. These groupings incorporated with the economic growth model and the joint lasso method were put against three other methods. Namely, a pooled regression of the economic growth model, a pooled mean grouped regression of the economic growth model and against the same economic growth model in combination with the joint lasso method but with two groupings. These two groupings were made by applying k-means with 'k = 2'. The values of γ and λ in the joint lasso were set by cross validation. We compared the predictive performance of these four methods and found that the pooled mean group estimator performed the worst. The pooled regression and the joint lasso on two groups performed a lot better. The difference in the mean squared prediction error was too small, between the pooled regression and the joint lasso on two groups, to declare one outperforming the other in terms of forecasting performance. Our two-step procedure, however, did outperform all the other methods and performed the best in terms of forecasting.

This does not mean that our two-step procedure is guaranteed to work. It needs to be tested further on different datasets, to investigate in which cases it does perform well. And to make sure it does not only outperform in this particular research. In their paper, Dondelinger et al. (2020), get mixed results and do not always outperform the pooled regression. Thus further emphasizing the need to investigate if the problem is finding the 'correct' partition or if the joint lasso itself is situational in its performance. Furthermore, it is interesting to further research if the two-step procedure works better with other cluster selection algorithms (which determine the number of clusters) or with other clustering methods then k-means. We have used the stability algorithms but perhaps the stability property is not the best method for dividing the data into groups for the joint lasso. Lastly, we set the $\tau_{k,k'}$ to unity in our research, which determines the similarity estimators should have between groups. However, a better approach could be to create a weighted version based on the distance between the cluster centers of the groups. This not only allows for the amount of similarity to differ between different groups, but it also allows the similarity to be stronger the closer the groups are according to the clustering method.

References

Alsabti, K., Ranka, S., & Singh, V. (1997). An efficient k-means clustering algorithm.

Arbelaitz, O., Gurrutxaga, I., Muguerza, J., PéRez, J. M., & Perona, I. (2013). An extensive comparative study of cluster validity indices. *Pattern Recognition*, *46*(1), 243–256.

Bassanini, A., & Scarpetta, S. (2002). The driving forces of economic growth. *OECD Economic studies*, *2001*(2), 9–56.

Bezdek, J. C., Li, W., Attikiouzel, Y., & Windham, M. (1997). A geometric approach to cluster validity for normal mixtures. *Soft Computing*, *1*(4), 166–179.

Brun, M., Sima, C., Hua, J., Lowey, J., Carroll, B., Suh, E., & Dougherty, E. R. (2007). Model-based evaluation of clustering validation measures. *Pattern recognition*, *40*(3), 807–824.

Burda, M., & Wyplosz, C. (2013). *Macroeconomics: a european text.* Oxford university press.

Dondelinger, F., Mukherjee, S., & Initiative, A. D. N. (2020). The joint lasso: high-dimensional regression for group structured data. *Biostatistics*, *21*(2), 219–235.

Dubes, R. C. (1987). How many clusters are best?-an experiment. *Pattern Recognition*, *20*(6), 645–663.

Fang, Y., & Wang, J. (2012). Selection of the number of clusters via the bootstrap method. *Computational Statistics & Data Analysis*, *56*(3), 468–477.

Fu, W., & Perry, P. O. (2020). Estimating the number of clusters using cross-validation. *Journal of Computational and Graphical Statistics*, *29*(1), 162–173.

Greene, W. H. (2000). Econometric analysis 4th edition. *International edition, New Jersey: Prentice Hall*, 201–215.

Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of classification*, *2*(1), 193–218.

Leahy, M., Schich, S., Wehinger, G., Pelgrin, F., & Thorgeirsson, T. (2001). Contributions of financial systems to growth in oecd countries.

Madigan, D. (2002). *Descriptive modeling.* Departement of Statistics, Rutgers University.

Milligan, G. W., & Cooper, M. C. (1985). An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, *50*(2), 159–179.

Milligan, G. W., & Cooper, M. C. (1988). A study of standardization of variables in cluster analysis. *Journal of classification*, *5*(2), 181–204.

OECD. (2016). *Irish gdp up by 26.3% in 2015?* Retrieved from `https://www.oecd.org/sdd/na/Irish-GDP-up-in-2015-OECD.pdf`

Raykov, Y. P., Boukouvalas, A., Baig, F., & Little, M. A. (2016). What to do when k-means clustering fails: a simple yet principled alternative algorithm. *PloS one*, *11*(9).

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, *58*(1), 267–288.

Tibshirani, R., Hoefling, G. N., Wang, P., & Witten, D. (2011). The lasso: some novel algorithms and applications.

Tibshirani, R., & Walther, G. (2005). Cluster validation by prediction strength. *Journal of Computational and Graphical Statistics*, *14*(3), 511–528.

Vichi, M., & Kiers, H. A. (2001). Factorial k-means analysis for two-way data. *Computational Statistics & Data Analysis*, *37*(1), 49–64.

Wang, J. (2010). Consistent selection of the number of clusters via crossvalidation. *Biometrika*, *97*(4), 893–904.

Appendices

A. K-means algorithm

```
Algorithm 1: K-means
  Data: A set of n observations $x_j$ and integer k
  Result: k number of cluster centers $\mu_i$ and which observation belongs to which cluster
  Initialize k centers $(\mu_1, \mu_2, ..., \mu_k)$ such that $\mu_i = x_j$ for $i \in k$ and $j \in n$
  while $J$ does not change significantly do
      for ( each vector $x_j$ ) {
          Assign $x_j$ to the nearest cluster center $\mu_c$
          (i.e. $\| x_j - \mu_c \|^2 \leq \| x_j - \mu_i \|^2$ for $i \in k$)
      }
      for ( each Cluster with center $\mu_i$ ) {
          calculate the new center $\mu_i^*$ of each cluster by calculating the centroid of all $x_i$
           belonging to the current cluster
      }
      compute the new $J = \sum_{i=1}^{k} \sum_{x_j \in S_i} \| x_j - \mu_i^* \|^2$
  end
```

B. Economic data sources

Variable	Source	Location
Real GDP per capita (y)	Eurostat	https://ec.europa.eu/eurostat/web/products-datasets/-/sdg_08_10
Capital accumulation (Kap)	Eurostat	https://ec.europa.eu/eurostat/web/products-datasets/product?code=sdg_08_11
Human capital (H)	United Nations	http://www.hdr.undp.org/en/data
Population growth (Pop)	World Bank	https://data.worldbank.org/indicator/SP.POP.1564.TO?end=2018&locations=EU&most_recent_value_desc=false&start=2000
GDP deflator (Fl)	World Bank	https://data.worldbank.org/indicator/NY.GDP.DEFL.KD.ZG.AD?end=2018&start=2001
Government revenue ($Govrev$)	Eurostat	https://ec.europa.eu/eurostat/web/products-datasets/-/gov_10a_main
Government expenditure ($Govexp$)	Eurostat	https://ec.europa.eu/eurostat/web/products-datasets/-/gov_10a_main
Government capital formation ($Govcap$)	Eurostat	https://data.worldbank.org/indicator/NE.CON.GOVT.ZS?end=2018&start=2000
R&D (RD)	Eurostat	https://ec.europa.eu/eurostat/web/products-datasets/product?code=tipsst10
Exports of goods and services	World Bank	https://data.worldbank.org/indicator/NE.EXP.GNFS.ZS?end=2018&start=2000
Imports of goods and services	World Bank	https://data.worldbank.org/indicator/NE.IMP.GNFS.ZS

C. Number of clusters as measure of success

A success is defined as the stability algorithm suggesting the value for 'k' to be the number of clusters corresponding to the simulated number of clusters. Instead of the value for 'k' that corresponds to the largest Rand index.

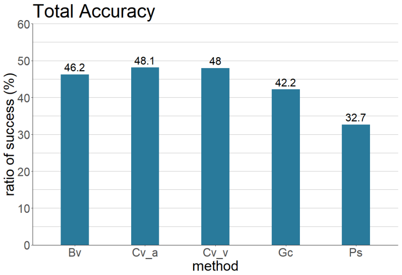
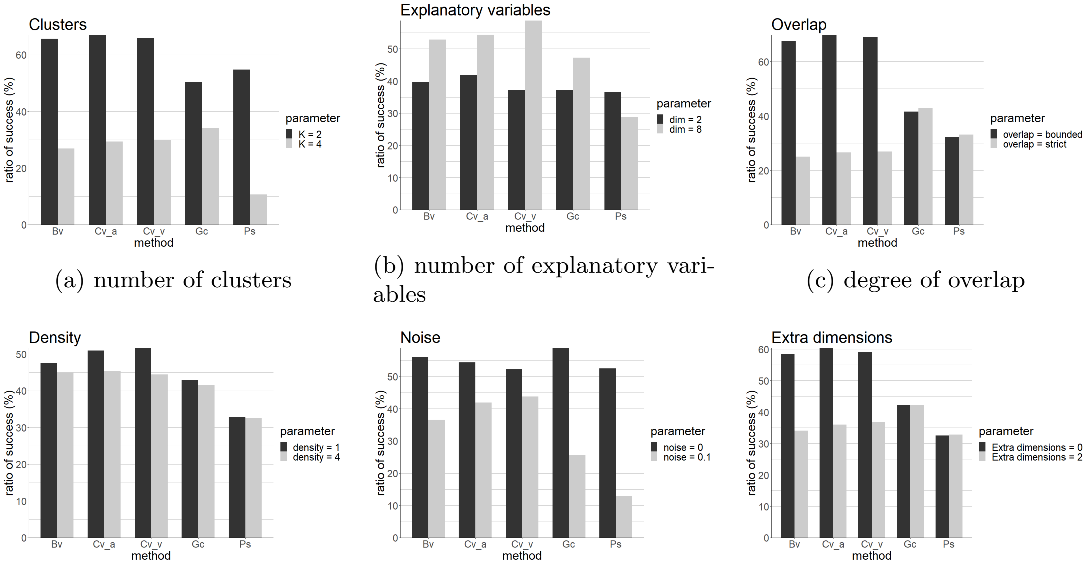


Figure A1: Overall results where the percentage of success rate is highlighted



(a) number of clusters

(b) number of explanatory variables

(c) degree of overlap

(d) density (dis)similarity

(e) level of noise

(f) number of noise dimensions

Figure A2: Success rate broken down by characteristics

D. Map of the cluster memberships

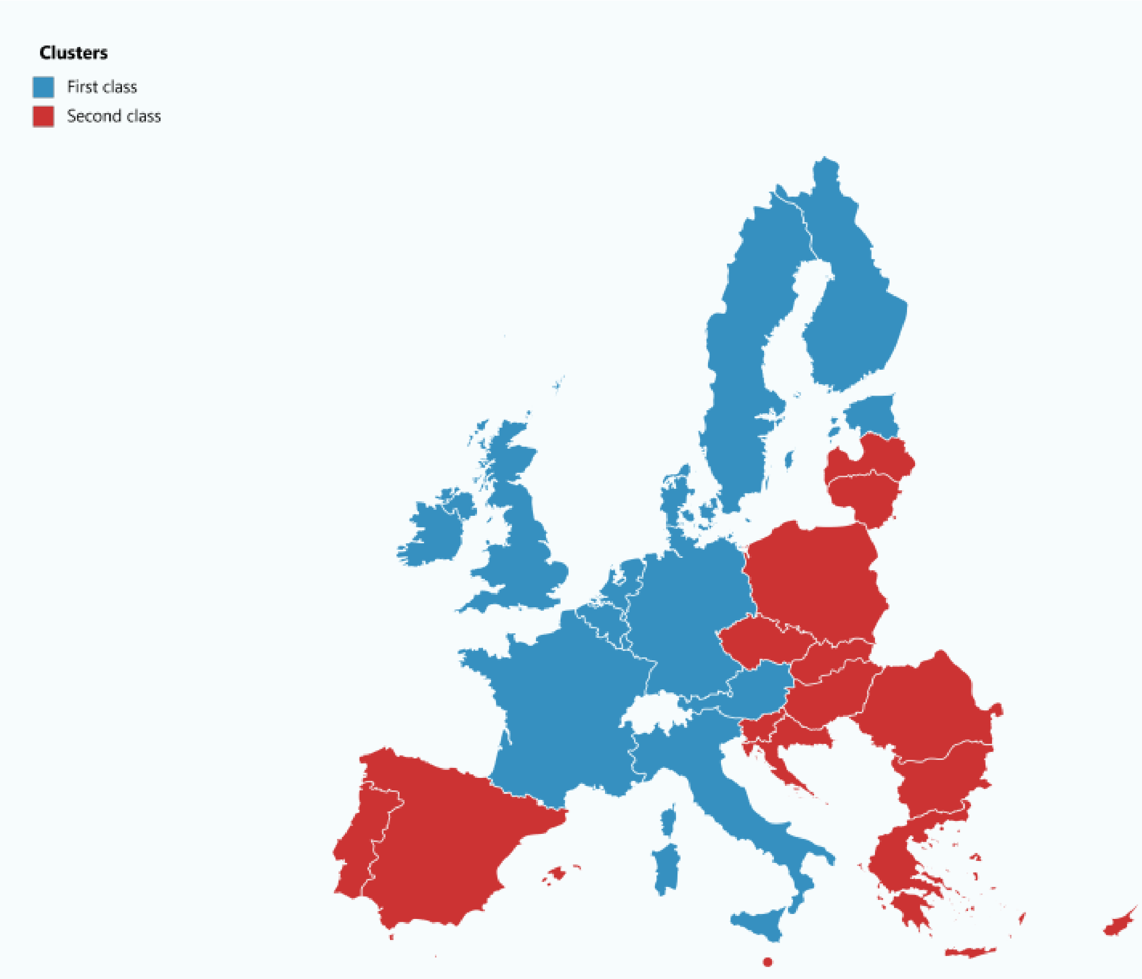


Figure A1: EU member states divided into two clusters.

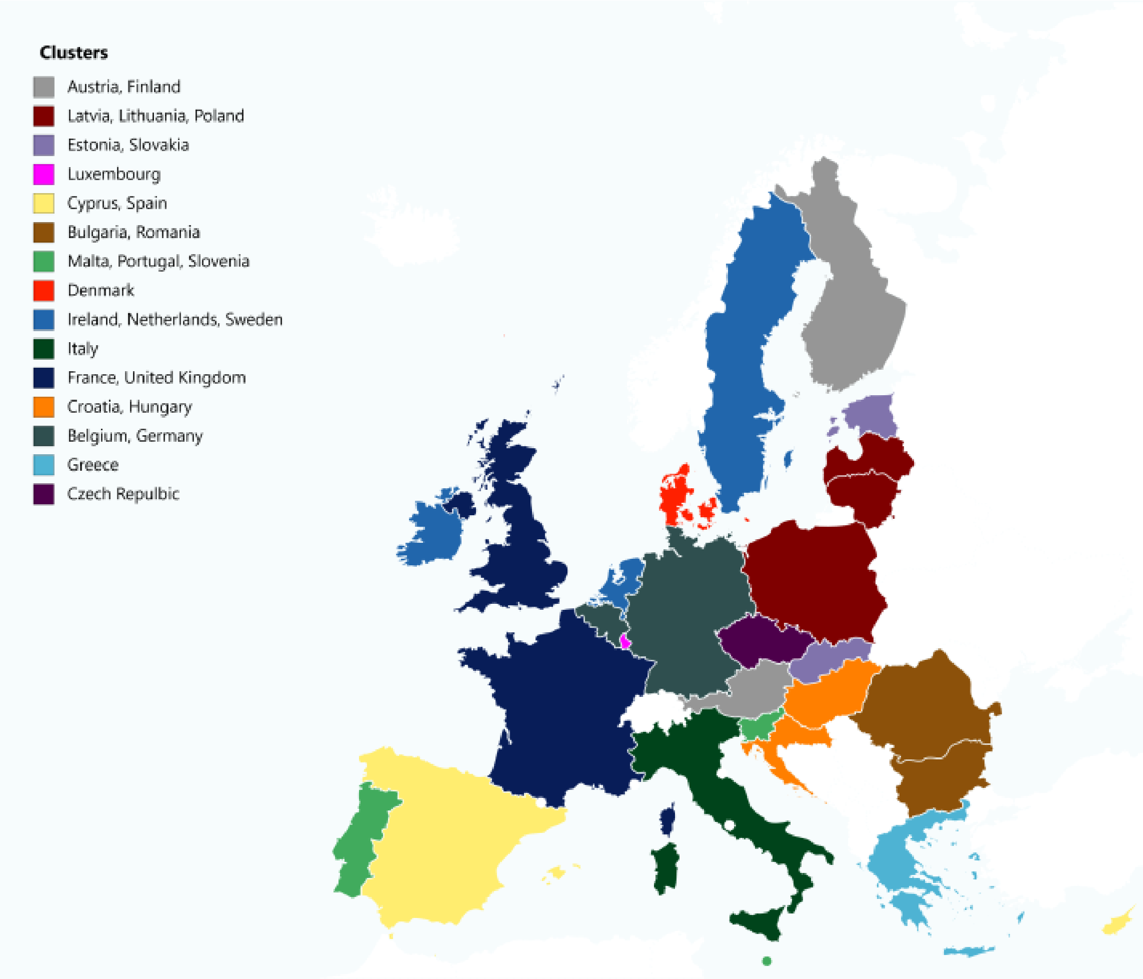


Figure A2: EU member states divided into fifteen clusters.