João, Igor Custodio; Lucas, Andre; Schaumburg, Julia

Working Paper

Clustering Dynamics and Persistence for Financial Multivariate Panel Data

Tinbergen Institute Discussion Paper, No. TI 2021-040/III

Provided in Cooperation with:
Tinbergen Institute, Amsterdam and Rotterdam

Suggested Citation: João, Igor Custodio; Lucas, Andre; Schaumburg, Julia (2021) : Clustering Dynamics and Persistence for Financial Multivariate Panel Data, Tinbergen Institute Discussion Paper, No. TI 2021-040/III, Tinbergen Institute, Amsterdam and Rotterdam

This Version is available at:
https://hdl.handle.net/10419/237773

TI 2021-040/III
Tinbergen Institute Discussion Paper

Clustering Dynamics and Persistence for Financial Multivariate Panel Data

Igor Custodio João[1]
Andre Lucas[1]
Julia Schaumburg[1]

[1] Vrije Universiteit Amsterdam and Tinbergen Institute, The Netherlands

Clustering Dynamics and Persistence for Financial Multivariate Panel Data*

Igor Custodio João[a] Andre Lucas[a] Julia Schaumburg[a]

[a]Vrije Universiteit Amsterdam and Tinbergen Institute, The Netherlands

May 6, 2021

*Email addresses: i.custodiojoao@vu.nl (Custodio João), a.lucas@vu.nl (Lucas), j.schaumburg@vu.nl (Schaumburg). Address correspondence to Igor Custodio João, Vrije Universiteit Amsterdam, De Boelelaan 1105, 1081 HV Amsterdam, The Netherlands, also reachable by email or by phone at +31 06 38 14 62 55.

Abstract

We introduce a new method for dynamic clustering of panel data with dynamics for cluster location and shape, cluster composition, and for the number of clusters. Whereas current techniques typically result in (economically) too many switches, our method results in economically more meaningful dynamic clustering patterns. It does so by extending standard cross-sectional clustering techniques using shrinkage towards previous cluster means. In this way, the different cross-sections in the panel are tied together, substantially reducing short-lived switches of units between clusters (flickering) and the birth and death of incidental, economically less meaningful clusters. In a Monte Carlo simulation, we study how to set the penalty parameter in a data-driven way. A systemic risk surveillance example for business model classification in the global insurance industry illustrates how the new method works empirically.

Key words: dynamic clustering, shrinkage, cluster membership persistence, Silhouette index, insurance.
JEL classification: G22, C33, C38.

Introduction

We propose a new method to cluster multivariate panel data in a dynamic yet stable and economically meaningful way. Building on established cross-sectional clustering methods such as k-mean clustering, we provide a straightforward and intuitive algorithm to link adjacent cross-sections by introducing persistence of assignments via a penalty parameter that can be chosen in a data-driven way. This results in clusters that are dynamic and time-varying in size, location, number, and composition. As our algorithm ties the different cross-sections and their clusters together, changes happen gradually over time and cluster switches become more persistent, both of which are important features in many economic and financial applications; see also our empirical application to business model identification in the global insurance industry.

Many approaches in econometrics thus far for clustering panel data with dynamic cluster numbers and cluster composition rely on either repeated cross-sectional clustering, hierarchical clustering, or on the clustering of whole time-series; see, for example, Oliveira and Gama (2012) for an application of sequential cross-sectional clustering, Ayadi et al. (2016) for hierarchical clustering with a related application to ours, and Bonhomme and

Manresa (2015) for a method using clustering of whole time-series. A variety of model-based methods to cluster panel data are surveyed in Frühwirth-Schnatter (2011), but none of these incorporate the dynamics of cluster composition, i.e., potential switches of cluster membership over time. All of these methods have substantial drawbacks for typical economic and financial applications. Repeated cross-sectional clustering typically generates clusters that are unstable over time, as the obtained structure at one point in time has no bearing on the next cross-section. Additionally, as cluster labels are arbitrary, it is unclear how groups can be tracked over time. Assignment instabilities are also likely to occur when the panel is treated as one large cross-section, to which a hierarchical clustering algorithm is applied. Clustering whole time series occupies the other end of the spectrum: it does not allow for changing compositions of clusters, as a unit remains in a cluster for all times. By contrast, in many economic applications we expect at least some units to possibly switch their cluster identity over longer periods of time. This is particularly true if the sample period covers periods of financial or economic crisis. Still, if units switch cluster identity, we expect them to typically do so gradually and persistently. For instance, in the context of business models we hardly ever see that units cross from group A to B one period, only to return from B to A the next period, and so on.

The new method we introduce in this paper extends the repeated cross-sectional clustering of for instance Oliveira and Gama (2012) by adding time-dependency to the cluster assignments. This induces stability, while still allowing for switches in cluster membership as well as changes in the number of clusters. More specifically, we penalize cluster switches by modifying how we measure distances from the cluster means. These distances include the distances of the current unit to its cluster mean in the previous cross-section, weighted by a penalty parameter. To set the penalty parameter, we propose a modified version of the silhouette index, a widely used cluster validation index introduced by Rousseeuw (1987). To enable tracking of the dynamic clusters over time, we build on algorithmic ideas from the literature to identify clusters by maximizing the overlap in cluster membership; see for instance Kalnis et al. (2005) and Oliveira and Gama (2010).

Our work also relates to the somewhat more distant literature on segmenting audio

recordings, see Fox et al. (2011). A typical finding there is that hidden Markov models produce over-segmentation, that is, too frequent jumping between states, a feature which we call flickering here. In that literature, the problem is solved by introducing a parameter for self transitioning and imposing a prior distribution over it while proceeding in Bayesian fashion. Our approach is different in that we reduce the dynamic problem to a collection of static ones, and introduce a stickiness (or self-transitioning) hyper-parameter chosen by well-known cluster validation criteria. Our work is also somewhat related to Catania (2021) and Lucas et al. (2020). Both of these use a dynamic mixture modeling approach, the former in an i.i.d. set-up, and the latter using a Hidden Markov Model (HMM) allowing for changes in cluster membership. Both of them are therefor subject to the over-segmentation problem signalled earlier. Lucas et al. (2020) addresses this by going to higher-order HMM dynamics, impeding cluster identity reversals too quickly after an initial switch. Our methodology departs from this approach in at least two important ways. First, we adopt a more standard, non-parametric approach to the clustering problem without leaning on the explicit parametric mixture model distribution assumptions in Catania (2021) and Lucas et al. (2020). This allows for an easy generalization of our approach to different clustering algorithms. Second, our penalty parameter determining the stickiness in cluster membership is determined in a data driven way, whereas the stickiness in Lucas et al. (2020) appears to be set exogenously.

Our application to business model classification in the global insurance industry over a longer, turbulent period illustrates the need for a compromise between allowing for time-variation and ensuring cluster composition stability. The clusters in our application can be interpreted as a business model classification and can be used for systemic risk surveillance; see Ayadi et al. (2016) for a related application in the banking industry. Though we would like to allow companies to change business model over a longer, possibly stressed time-span, we would typically see such changes as rather persistent once engaged in, as they relate to high-level strategic choices of the companies involved. Ayadi et al. (2021) apply Ward's (1963) hierarchical clustering algorithm to a set of bank characteristics, treating the panel as one cross-section, and investigate the changing composition of each

cluster over time. As their method puts no restriction on the number of cluster switches a bank can endure, erratic switching of business models may be found. Furthermore, in their approach, there is no way to incorporate dynamics in cluster parameters, that may occur due to market developments or crises. In contrast, Lucas et al. (2019) use a generalized autoregressive score finite mixture model with moving clusters, but without changes in cluster composition. Our approach extends both of these by allowing for persistent switching via penalization of erratic switches or so called flickering. As we show, this results in much more stable, though still dynamic clusters.

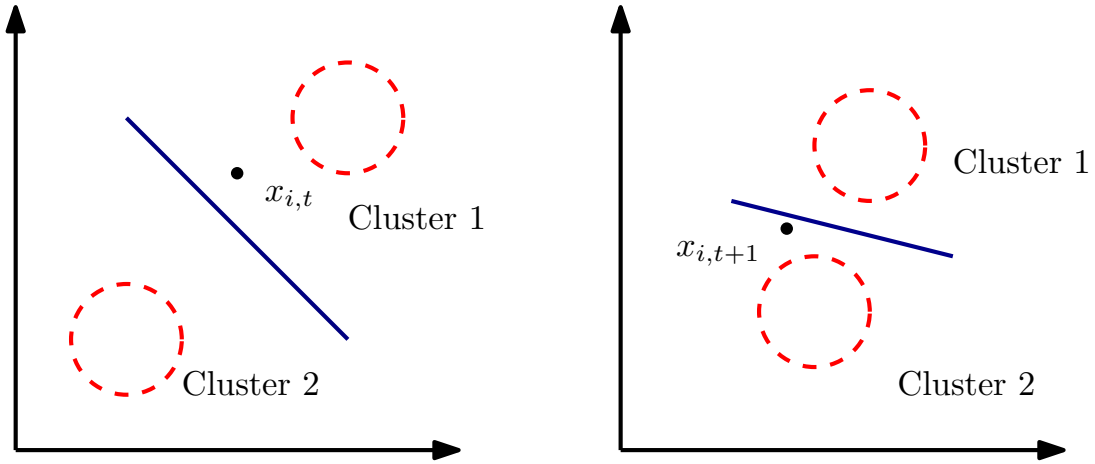
The remainder of this paper is set up as follows. In Section 1 we introduce the methodology and discuss how to set the penalization parameter. Section 2 studies the method in a controlled setting and shows that the method reduces overall misclassification rates compared to competing methods. Section 3 discusses the empirical application. Section 4 concludes.

1 Methodology

In this section, we first introduce our robust clustering methodology. Next, we explain how we link cluster identities over time, which is a crucial step in our method. Finally, we provide data-driven ways to select the shrinkage penalty parameter in our approach.

1.1 Penalized cross-sectional clustering

Consider a panel of multivariate financial data, with $x_{i,t}$ denoting a *vector* of observed characteristics for unit $i = 1, \ldots, N$ at time $t = 1, \ldots, T$. Our goal is to assign each unit i to a peer group of similar units at each point in time t. An example of such a situation is the monitoring of business models in the financial industry by a regulator; see Section 3. In the realistic setting of changing market conditions, technological advances, and shifts in regulatory requirements, we expect that some firms may move to a different group or business model at some point in time. However, switching from group A to group B at one point in time, only to switch back to A in the following period, is unrealistic in many



(a) time t (b) time $t+1$

Figure 1: k-means clustering at two consecutive times. The red circles represent the location of the cluster centers. The blue line separates the clusters and is halfway between both cluster centers.

situations that involve long-term strategies, such as the choice of a business model. A suitable clustering method should therefore mitigate excessive cluster switches.

To illustrate this, consider an example with two features in $x_{i,t}$. This is visualized in in Figure 1. Assume we cluster each cross-section t separately by the k-means algorithm into two clusters. Units are assigned to the cluster with the closest cluster center. This divides the space in two regions. If an observation $x_{i,t}$ at time t is close to the border that separates the clusters, as in the left-hand panel, even a small disturbance to its position might shift it to the other cluster. A second switch might then occur if it suffers another small and roughly opposite disturbance in the next period, and so on. We would observe significant short-lived cluster switches or 'flickering' caused by little actual movement. Such flickering might not be economically meaningful, and therefore undesirable.

The model presented here takes the cross-section at time t and combines it with the $t-1$ cluster assignments to produce sticky assignments at time t. The model is based on a first-pass clustering using one of the many available standard algorithms, followed by a penalized re-evaluation of the assignments. We assume that the number of cluster switches is sufficiently small from one cross-section to the next to allow us to track cluster movements through time by means of the identity of the cluster members. To appreciate this, note that when stepping from time $t-1$ to t, the labels in the cross-sectional cluster

assignments may have no clear correspondence. For instance, the labels A and B at time $t-1$ might have been switched around at time t. Alternatively, we might have two clusters A and B at time $t-1$, whereas we have three candidate clusters A, B, and C at time t. To solve the arbitrary labeling of clusters across cross-sections, we propose a mapping procedure based on the maximum overlap between cluster membership. The procedure is explained in detail in Section 1.2 and provides the distance of $x_{i,t}$ to the *current* candidate location of unit i's *previous* cluster. For instance, if unit i belonged to cluster A at $t-1$, we will not only consider unit i's distance to cluster B at time t, but we will also consider its distance to the *current* (time t) position of cluster A.

To introduce the formal algorithm, we first need some further notation. Let $h_{i,t}$ denote the cluster assignment of unit i at time t, such that $h_t = (h_{1,t}, \ldots, h_{N,t})'$ denotes the $N \times 1$ vector of all cluster assignments for cross-section t. We now start at time $t = 1$ with a standard cross-sectional clustering algorithm and cluster selection criterion to obtain the number of clusters K_t and the cluster identities h_t at $t = 1$. Next we move to the next time period $t = 2$ and run a clustering algorithm to obtain a *candidate* set of cluster assignments $\tilde{h}_t$. Using the mapping methodology M of Section 1.2, we relabel the cluster identities in $\tilde{h}_t$ to $\tilde{h}'_t = M(h_{t-1}, \tilde{h}_t)$, such that the identities in h_{t-1} and $\tilde{h}'_t$ are comparable, i.e., when say 90% of the units in cluster A at time t appear in what is called cluster Z at time $t+1$, then we re-label cluster Z to be called A. Based on $\tilde{h}'_t$ we can compute the *current*, candidate location of each of the *previous* clusters in h_{t-1}, except for the clusters that were discontinued. As an example, if we would estimate the location by the mean, the current location of unit i's previous cluster can be estimated by $c(h_{i,t-1}, h'_t) = (\#P_i)^{-1} \sum_{j \in P_i} x_{j,t}$, where $P_i = \{j \mid \tilde{h}'_{j,t} = h_{i,t-1}\}$ and $\#P_i$ denotes the number of elements in P_i. If the number of elements in P_i is positive, we now shrink the observation $x_{i,t}$ towards the current location of its previous cluster. We do so by defining $\tilde{x}_{i,t} = (1-\varepsilon) \cdot x_{i,t} + \varepsilon \cdot c(h_{i,t-1}, \tilde{h}'_t)$, where ε is a fixed penalty parameter in the unit interval, and $c(k, h)$ denotes the cluster location or center vector for cluster identity k using assignment vector h. The effect can be seen in Figure 2.

Using the shrunk observations, we now run a second pass of the cluster assignments

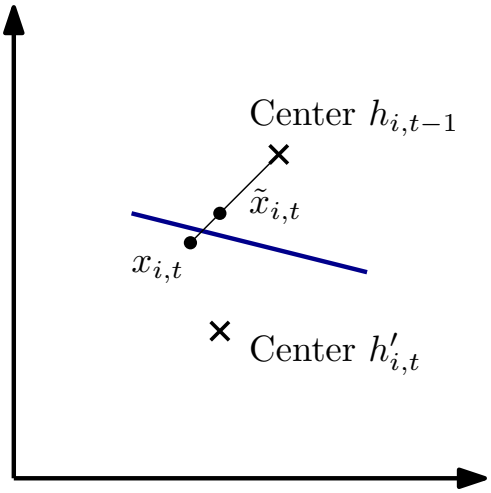


Figure 2: We can interpret $\tilde{x}_{i,t}$ as an artificial position lying an ε fraction of the way from the current position $x_{i,t}$ and the center of its last cluster $h_{i,t-1}$

as

$$h_{i,t} = \mathbb{1}_{i,t} \cdot \tilde{h}'_{i,t} + (1 - \mathbb{1}_{i,t}) \cdot h_{i,t-1}, \tag{1}$$

$$\mathbb{1}_{i,t} = \begin{cases} 1 & \text{if} \quad \#P_i = 0 \quad \text{or } d\big(\tilde{x}_{i,t}, c(h'_{i,t}, \tilde{h}'_t)\big) < d\big(\tilde{x}_{i,t}, c(h_{i,t-1}, \tilde{h}'_t)\big) \\ 0 & \text{else.} \end{cases} \tag{2}$$

where d is a distance measure. In words: if the shrunk observation $\tilde{x}_{i,t}$ is closer to the new candidate cluster, or if the old cluster is discontinued, the unit switches to the new cluster. Otherwise, the cluster remains in the old cluster. [1] The shrinkage of the observation towards the current location of the previous cluster ensures that cluster switches become less likely. If ε equals zero, there is no shrinkage and units can switch cluster identity freely from one cross-section to the next. The steps are repeated for all cross-sections $t, t+1, \ldots, T$, including a step to determine the number of clusters in each cross-section. The complete algorithm is summarized in Algorithm 1.

It is important to note here that we have been silent thus far about which clustering algorithm is used, which distance measure d, and which measure of cluster centroid c. This means that the current shrinkage technique can be applied in a wide variety of settings. Any cross-sectional clustering algorithm that produces a distance measure can be adapted in the above way to feature stickiness. For example, in graph-based algorithms such as in

Algorithm 1: Dynamic clustering with shrinkage

input : The data, the maximum number of clusters K per cross-section, a shrinkage parameter ε.

output: T vectors of assignments h_t.

for $t \in [T]$ **:**

  for $k \in \{2, 3, \ldots, K\}$ **:**

    Run clustering algorithm; obtain candidate cluster assignments $\tilde{h}_t$

    if $t > 1$ **then**

      $\tilde{h}'_t \leftarrow M(h_{t-1}, \tilde{h}_t)$

      compute new locations of old clusters: $c(h_{i,t-1}, \tilde{h}'_t)$

      shrink observations to old clusters: $\tilde{x}_{i,t}$

      $h^{(k)} \leftarrow$ re-assign shrunk observations to clusters

    else

      $h^{(k)} = \tilde{h}_1$

    $s_t^k \leftarrow$ compute silhouette index for this cross-section based on current cluster assignments h_t

  $K_t \leftarrow \arg\max_i s_t^i$; select number of clusters in cross-section t

  $h_t \leftarrow h^{(K_t)}$; store final assignments for cross-section t

Zahn (1971) or Grundmann et al. (2010), we can shrink the weight of edges connecting points that belonged to the same cluster at $t-1$.

To select the number of clusters in each cross-section in Algorithm 1, we use the silhouette statistic of Rousseeuw (1987), which peaks at the optimal number of clusters. The silhouette of point i at time t is given by

$$s(x_{i,t}) = \frac{b(x_{i,t}) - a(x_{i,t})}{\max\{a(x_{i,t}), b(x_{i,t})\}},$$

where $a(x_{i,t})$ is the average distance from point i to other points in its own cluster, and $b(x_{i,t})$ is the average distance from point i to the points in the nearest other cluster. That is, if i belongs to cluster A, and $d(x_{i,t}, C)$ is the average distance from i to the points in some cluster C, we can write

$$a(x_{i,t}) = d(x_{i,t}, A)$$

and

$$b(x_{i,t}) = \min_{B \neq A} d(x_{i,t}, B).$$

The average for a cross-section at t is then $n^{-1}\sum_{i=1}^{n} s(x_{i,t})$. Following Rousseeuw (1987) we set $s(x_{i,t}) = 0$ if A only contains unit i. Intuitively, the average silhouette measures how tight the clusters are (when $a(x_{i,t})$ is low) and how separate they are from each other (when $b(x_{i,t})$ is high). This makes it a useful measure of fit, which we adapt in Section 1.3 to obtain a data-driven way to select the shrinkage parameter ε .

1.2 Mapping

Standard cross-sectional clustering algorithms produce arbitrary labels without any connection to labels assigned in previous cross-sections. These impedes the identification of the current location of an observation's previous cluster. In order to identify an observation's previous cluster, we need to find a correspondence between the labels at $t-1$ and new candidate labels at t. To do so, we leverage on the time-dimension of the units by looking at the overlap of every two clusters at consecutive times, as in Kalnis et al. (2005). To illustrate this, consider a setting where at time t the cross-sectional algorithm produces clusters A and B, while at time $t+1$ it produces the clusters labeled C and D. If all units that belong to cluster A at time t also belong to cluster C at time $t+1$, and all units that belong to cluster B at time t belong to cluster D at $t+1$, then the most natural correspondence is to assign the same label to A and C, as well as to B and D. Following Oliveira and Gama (2010), we refer to this procedure as *mapping*.

To generalize this idea to the less obvious case where there are switches, we form a contingency matrix where the element in row i and column j represents how many points were assigned to cluster i at time t and to cluster j at time $t+1$. We can then formalize the idea of maximizing overlap between clusters at different times as maximizing the trace of this matrix with respect to the ordering of the columns. For example, if both periods have two clusters and the maximum is attained when the contingency matrix is formed with the column corresponding to cluster 2 on the left and the one for cluster 1 on the right, then cluster 2 at $t+1$ maps to cluster 1 at t, and so on. This is the case for the

matrix below where we reorder the columns to achieve the maximum overlap:

$$\text{tr}\begin{pmatrix} 3 & 2 \\ 5 & 1 \end{pmatrix} = 4 \;\rightarrow\; \text{tr}\begin{pmatrix} 2 & 3 \\ 1 & 5 \end{pmatrix} = 7.$$

This problem can also be written as:

$$\max_{P} \text{tr}(C_t P), \tag{3}$$

where C_t is the contingency matrix from time t to $t+1$, and P is a permutation matrix (where the elements are either zero or one and each column has exactly one non-zero element). In the example above the optimal P would be

$$P^* = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

For a small number of clusters at $t+1$, this problem can be easily solved by exhaustive search. However, this is not always true. For 10 clusters we would need to look at $10! = 3628800$ cases. Fortunately an efficient algorithm has been developed for this problem, called the Hungarian algorithm (Kuhn, 1955).

An extension of this method to the situation where the number of clusters increases or decreases from t to $t+1$ can be easily defined. In this case, the contingency matrix becomes rectangular so it is not possible to take its trace. A solution is to maximize the trace of the largest square matrix inside it by switching the columns of the rectangular matrix. That is, the extra clusters' overlap will not go into the objective function. We can still formulate the problem as in equation (3) if we augment the C_t with a matrix of zeroes such that the resulting matrix is square, i.e.,

$$C_t^* = \begin{pmatrix} C_t \\ 0_{m-n\times m} \end{pmatrix}, \qquad C_t^* = \begin{pmatrix} C_t & \mathbf{O}_{n\times n-m} \end{pmatrix}, \tag{4}$$

for the case $n < m$ and $n > m$, respectively, where $0_{a\times b}$ is the matrix of zeroes of dimension $a \times b$. The problem can then again be written as (3), with C_t^* taking the place of C_t. This formulation is equivalent to stating that the extra clusters exist in both time steps, but have no members in one of them, when building the contingency matrix. Therefore this extended problem can still be solved by the Hungarian algorithm.

The solution P can be interpreted as follows. Cluster i at $t+1$ maps to cluster j at t if $P_{i,j} = 1$, where $P_{i,j}$ is the (i,j)th element of P^*. Put differently, if we are interested in finding the $t+1$ counterpart of cluster j at t, we need to look at the row index of the element in column j of P equal to 1. Similarly, the counterpart of $t+1$ cluster i at time t is found by the column index of the ith row of P holding the number 1.

1.3 Selection of the shrinkage parameter

In this section, we propose a modification of the silhouette statistic to set the shrinkage parameter ε. We benchmark this statistic against the cross-validation approach of Fu and Perry (2020), which we also briefly introduce in this section, but which turns out to work less satisfactorily in the simulations.

Our aim is to reduce misclassification of observations to clusters. However, as clustering is an unsupervised learning technique, such misclassification can only be measured in a controlled simulation setting. For real data, misclassification cannot be measured, as the true cluster labels are unknown.

As our main method for selecting ε we use the Gini-weighted silhouette index. As discussed earlier, the silhouette index by itself is used to select the appropriate number of clusters in each cross-section t in Algorithm 1. We aggregate it to a measure of fit for the entire panel by multiplying each time-t average silhouette by one minus the Gini coefficient, using the formula given by David (1968), re-scaled by N:

$$G_t = \frac{\sum_{i=1}^{K_t}(2i - K_t - 1)\cdot \#P_{i:K_t}}{K_t \cdot N}, \qquad GWS = \sum_{t=1}^{T}(1 - G_t)\cdot s_t,$$

where $\#P_{i:K_t}$ is the number of units in the i-th smallest cluster, and s_t denotes the

silhouette index of cross-section t. David (1968) shows that this is equivalent to the mean absolute difference

$$G_t = \frac{\sum_{i=1}^{K_t} \sum_{j=1}^{K_t} \left| \#P_i \; - \; \#P_j \right|}{2K_t \cdot N}.$$

The latter expression clearly shows that G_t equals zero when the clusters have homogeneous sizes, and increases to $1 - K_t^{-1}$ as inequality increases.

The GWS emphasizes clustering outcomes with a more even division of observations over clusters per cross-section as opposed to a few large clusters and a large number of isolated small clusters. This is in line with our objective to reduce flickering: we want to discourage the short-lived birth and death of small, isolated clusters from one cross-section to the next. Note that the GWS statistic is easy to compute and a direct by-product of Algorithm 1.

To benchmark the Gini weighted silhouette GWS, we also compute the cross-validation statistic for clustering of Fu and Perry (2020). In their paper, Fu and Perry use cross-validation to determine the optimal number of clusters in a cross-sectional clustering problem. We instead use their approach to set the shrinkage parameter ε.

This cross-validation technique of Fu and Perry (2020) consists in splitting the dataset both along units and variables, such that we have 4 groups in each "fold" (or subsample) of the entire dataset. The variables are split into artificial "predictor" and "response" variables, while the units are split into training and testing datasets. By applying our clustering method to the training response variables, cluster assignments are produced that we treat as observed, similar as in a standard cross-validation procedure for classification problems. Next, the training predictor variables are used to fit cluster means that best predict the assignments. This results in the best classifier in the training sample. Finally, this classifier is evaluated on the testing dataset and results in a cross-validation error. The cross-validation error is computed for a range of values for the penalization parameter ε, and used to determine the optimal ε. This procedure mimics the approach in Fu and Perry (2020), but than for ε rather than for the number of clusters as done in the original paper.

2 Simulations

In this section, we investigate the ability of our method to assign units to their respective clusters at each time point. Since we are in a situation of unsupervised learning, misclassification rates are not observed when analyzing real data. The current simulation setting is therefore particularly useful, because here we observe the true assignments and therefore can assess the misclassification performance of our data-driven shrinkage parameter selection approach and compare it to benchmark methods from the literature.

All simulations are based on a number of clusters drawn from six-dimensional Gaussian distributions.[1] The centers are drawn randomly from the vertices of a six-dimensional unit hypercube. The cluster covariance matrices are set equal to the identity matrix in the benchmark simulation setting. Unit variances in a unit cube imply that there is a large probability that a draw will be closer to another than to its own cluster center, thus creating substantial overlaps of the point clouds and substantial misclassification risk. In a second set-up we limit the overlap by setting the variances to $1/2$ for each component. At each time point, observations are drawn from their current cluster distribution. Units switch clusters at each t with probability p, where we vary p from 0 to 0.25 across different designs. In all settings we use $T = 20$ time points, $N = 120$ units, both in line with the empirical data in Section 3, and 100 simulations runs. As our first-pass cross-sectional clustering algorithm we choose a simple k-means, though as stated before our method can accommodate different cross-sectional clustering methods, distance definitions, and ways to measure cluster centriods.

The baseline simulation results are shown in Figure 3. We see that at low levels of switching in the DGP, i.e. $p \in \{0, 0.01, 0.1\}$, our method with positive ε improves on the unconstrained case ($\varepsilon = 0$). Moreover, there are clearly optimal values for ε in the misclassification plot (upper left panel). Setting ε to the these optimal values leads to reductions of misclassification errors from 16% down to 16% for both $p = 0$ and $p = 0.01$. Without knowing the true classifications, it is still striking that the Gini-

[1] We choose 6 dimensions so as to allow at least three variables in each fold for the cross-validation.

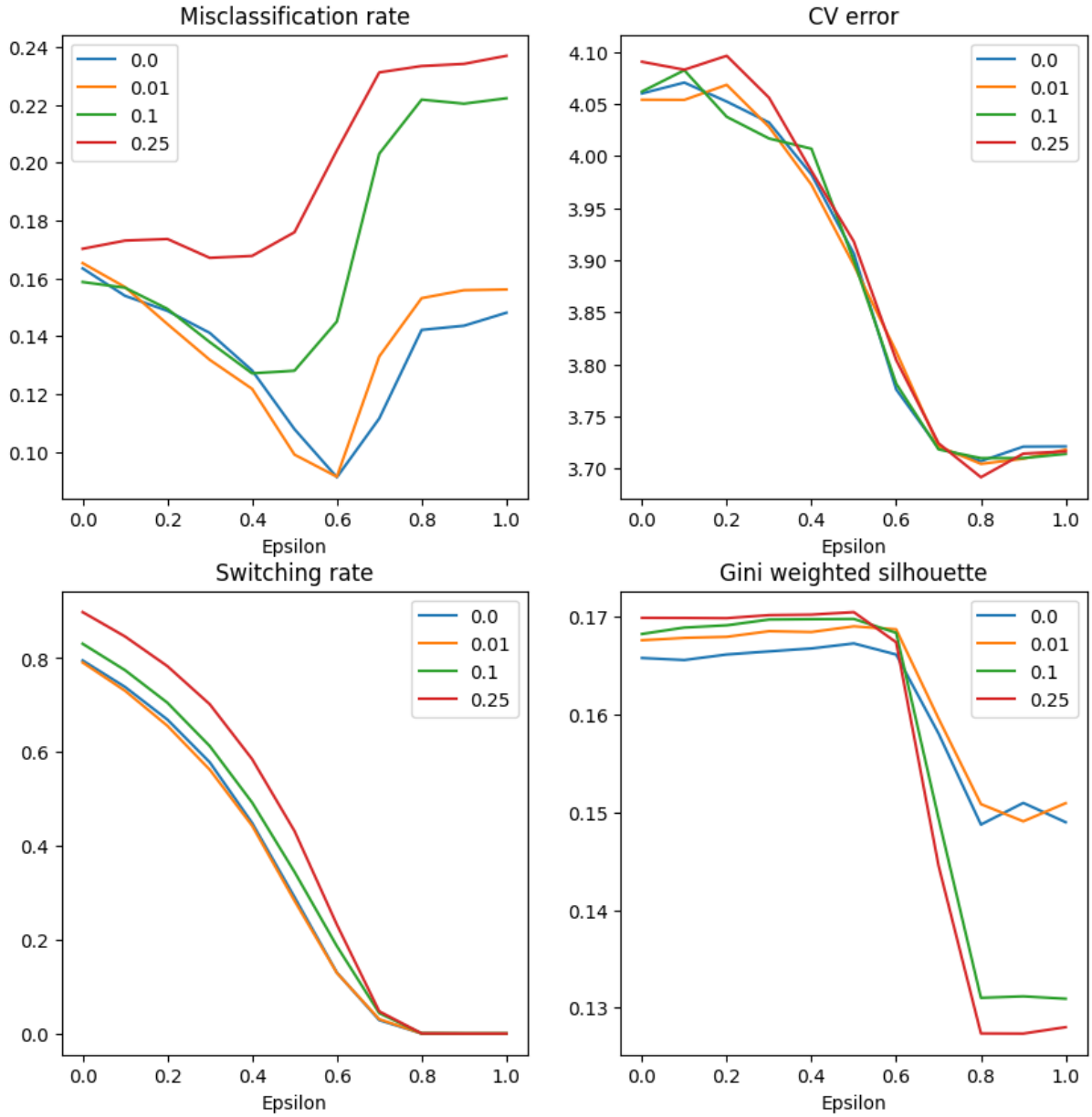


Figure 3: Simulation results for four values of p. Baseline setting.

weighted silhouette index peaks at about the optimal ε (lower right panel), while the cross-validation (CV) error flattens out around the same point (upper right panel). The plot of switching rate (lower left panel) shows exactly what the model achieves: drastic reduction in the number of cluster switches. Our validation strategy seems to become less accurate only when we increase p beyond what we expect from a setting with persistent assignments such as business model classification, see e.g. $p = 0.25$. We do not expect this to be a major problem as our model is primarily intended for dynamic settings with substantial cluster membership persistence.

It is important to note that the baseline case presents a challenging clustering problem: the variances are high compared to the distances between the cluster means, such that standard cross-sectional methods might produce large errors. In a setting with lower variances such as in Figure 4, we find much smaller misclassification errors. Still, by increasing ε away from 0 reaches we decrease the misclassification rate by about 67% for $p = 0.00$ and 0.01. The maximum of the Gini-weighted silhouette index in these cases again points clearly to the minimal misclassification rate. For higher p, there is no clear maximum between 0 and 0.5 for the Gini weighted silhouette, which is in accordance with the relative flatness of the misclassification curve for these values of ε. We conclude that there is little change in the fit over this region. When setting $\varepsilon > 0.5$ we see the misclassification rates for $p = 0.1, 0.25$ picking up, something we see mirrored in the weighted silhouette for $\varepsilon > 0.6$.

We also see in Figures 3 and 4 that the cross-validation error approach to select ε works less well. Cross-valiation errors appear lowest for high values of ε, i.e., with too much persistence in cluster membership. If ε were set based on this criterion, misclassification rates would be higher than for the weighted silhouette approach. We therefore do not use the cross-validation approach in our empirical work in Section 3.

To see the effect of choosing the number of clusters, we extend the previous simulation set-ups by also letting the algorithm choose the number of clusters K_t in each cross-section from 2 to 4, whereas the true number of clusters is always 2. The results are in Figures 5 and 6 for the unit variance and the 1/2 variance case, respectively.

Particularly the case of an unknown number of clusters combined with large cluster overlap poses a challenge for any method. This is seen in Figure 5. Misclassification rates are high throughout, and only for $p = 0.00, 0.01$ show a clear dependence on the shrinkage parameter ε. Particularly in those two cases, the reduction in misclassification is substantial at more than 15 percentage points when the optimal penalty parameter is chosen. We see that the Gini-weighted silhouette points to values of ε between 0.3 and 0.6, where the sharpest declines in the weighted silhouettes occur. These values appear slightly below the optimal values for misclassification at around $\varepsilon = 0.6$. The CV error,

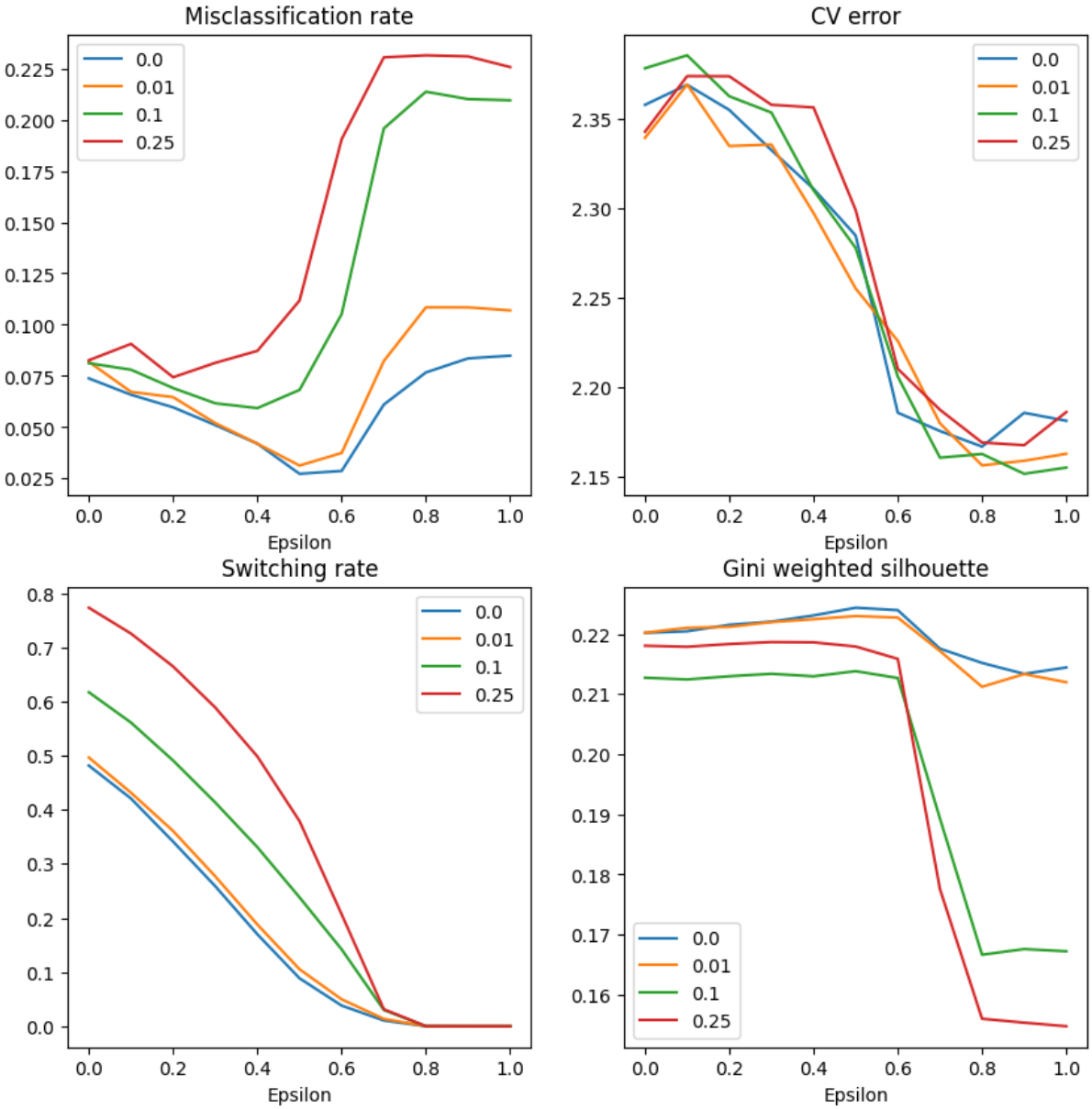


Figure 4: Simulation results for four values of p. Half-variance setting.

by contrast, seems to flatten out around too high value of around $\varepsilon = 0.8$.

As mentioned, the setting with unit variance and cluster centers on the unit hypercube vertices is extremely challenging. If we scale back the challenge somewhat by considering variances of 0.5, we obtain the results in Figure 6. This figure recovers our earlier finding, particularly for low true levels of switching the Gini-weighted silhouette decreases sharply after the optimal value of ε from a misclassification perspective, allowing us to cut misclassification rates in those settings by 50% in a data-driven way. By contrast, the cross-validation based approach again results in much to high persistence levels of cluster

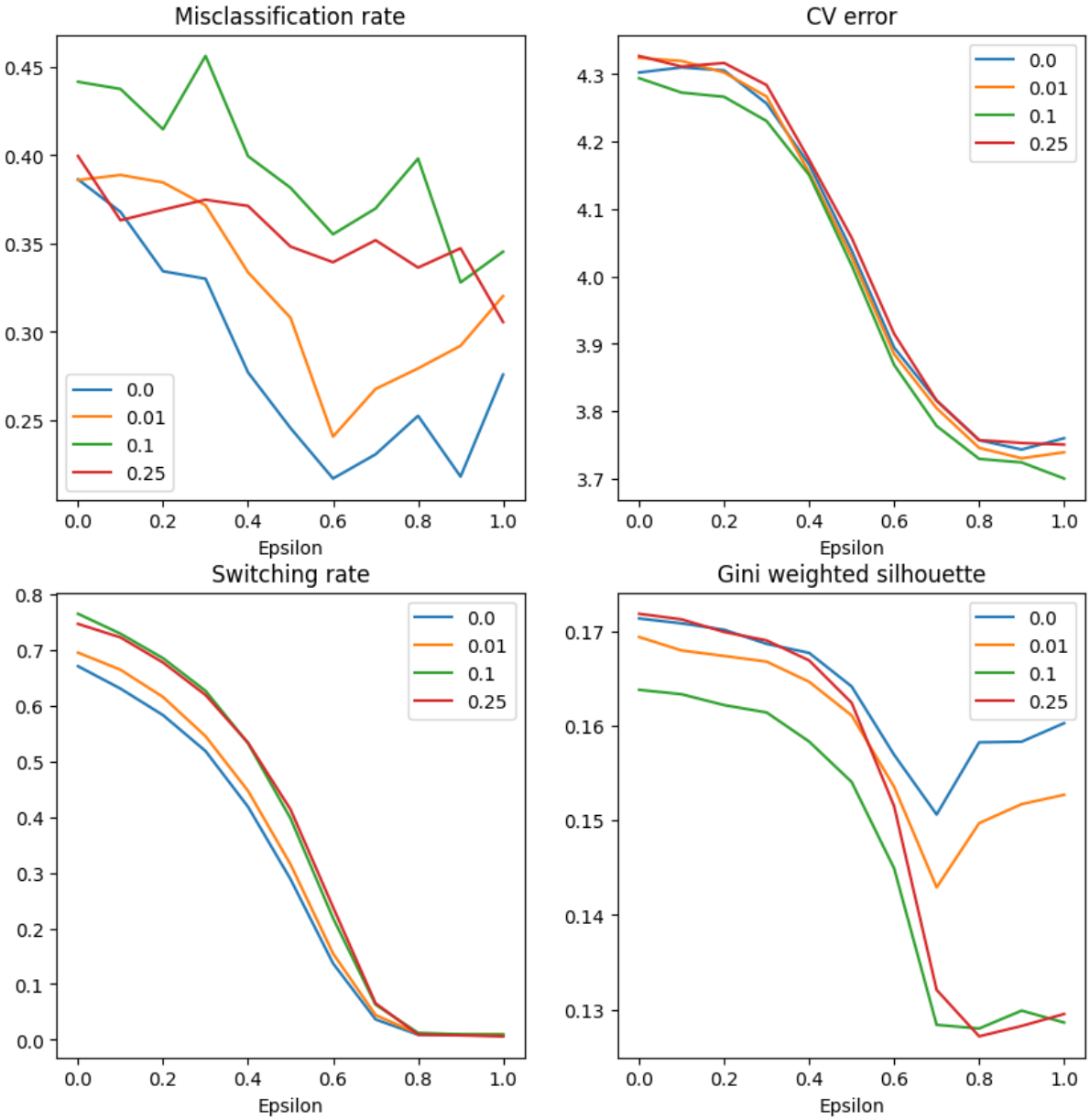


Figure 5: Simulation results for four values of p. Benchmark setting. The number of clusters k may vary between 2 to 4, while the true number is always 2.

membership, and therefore may miss important aspects in the dynamics of the data.

To benchmark our clustering approach, we run three clustering approaches based on Ward's hierarchical clustering. The first approach (Ward plain) clusters each cross-section separately, and links the labels through the mapping step as in Section 1.2. The pooled Ward benchmark takes all observations of units in time as separate units, resulting in $N \times T$ units in total, and treats them as a single cross-section in the clustering step. Finally, time-aggregated Ward uses each observation t of the same unit i as different

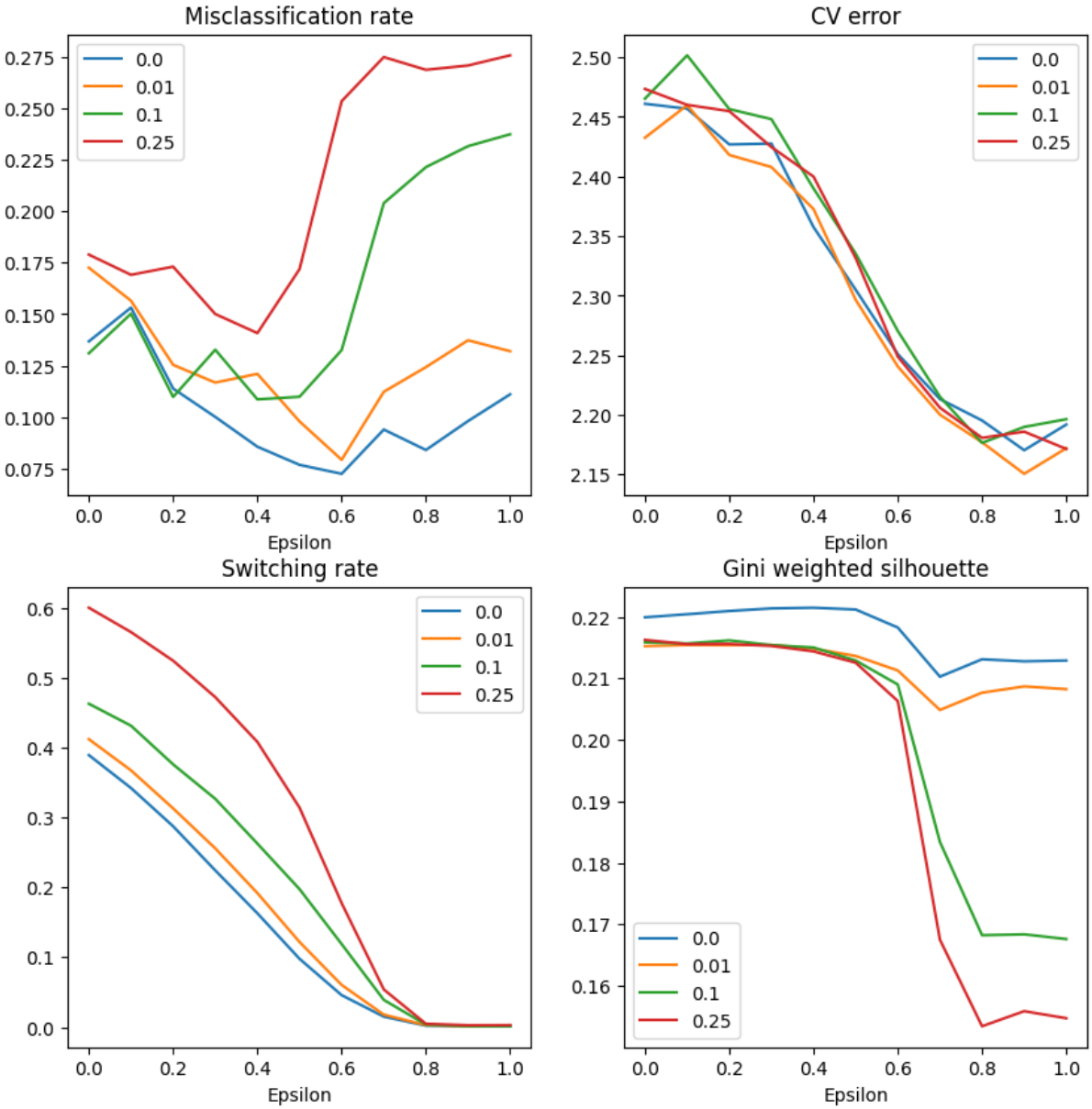


Figure 6: Simulation results for four values of p. Half-variance setting. The number of clusters k may vary between 2 to 4, while the true number is always 2.

variable. This results in N units observed in $P \times T$ dimensions. This approach does not allow for switches and effectively clusters the whole time series of a unit. All benchmarks are implemented in a simulation set-up with the correct number (2) of clusters. The results are presented in Table 1 and can be compared to Figures 3 and 4.

		Misclassification		Switches	
		Baseline	Half-var.	Baseline	Half-var.
Model	p				
Ward	0.0	0.182	0.101	0.391	0.285
plain	0.01	0.182	0.108	0.392	0.305
	0.1	0.183	0.119	0.408	0.348
	0.25	0.208	0.133	0.431	0.393
Ward	0.0	0.182	0.117	0.368	0.267
pooled	0.01	0.170	0.128	0.368	0.284
	0.1	0.176	0.122	0.386	0.330
	0.25	0.185	0.121	0.423	0.379
Ward	0.0	0.019	0.001	0.000	0.000
time-aggregated	0.01	0.045	0.043	0.000	0.000
	0.1	0.212	0.195	0.000	0.000
	0.25	0.284	0.281	0.000	0.000

Table 1: Misclassification rates and switches for the benchmark models.

All benchmarks approaches produce larger misclassification errors than our new penalized clustering approach. Only Ward's time-aggregated approach for small p appears to fare slightly better, but at the cost of not allowing any switches at all. As a consequence, it produces disproportionately large errors as p increases, exceeding the penalized clustering approach of this paper. The difference in misclassification rates are substantial enough that a large set of sub-optimal choices of ε still beat the benchmarks. For instance, in the baseline design with p set at 0 and 0.01, any choice of ε produce lower errors than either the Ward or the Ward pooled benchmark. At $p = 0.1$ the missclassification rate in our approach is only higher when $\varepsilon > 0.6$. This suggests that in such settings, the penalties added to cluster swithches in our approach only have a mild effect if cluster swithches happen more often.

3 Empirical application

In this section we apply the clustering methodology described above to a global dataset of insurance company balance sheet items. The data are taken from Compustat. To the extent of our knowledge, this is the first clustering study using insurance firms' balance sheet data to identify business models, where most previous studies use banking data instead. We first describe the dataset. Next, we apply the algorithms and the validation and diagnostics methods described earlier in this paper, and evaluate the composition of the resulting clusters.

Following the choice of variables in Lucas et al. (2019), we choose variables that broadly reflect the split between insurance and investment income, and are widely available across the sample. The variables chosen are: Total Assets, Total Revenue, Insurance Income, Insurance Premiums, Investment Income, Investment Assets, Cash, Benefits and Claims, and Reinsurance Assets. These variables have been used before to study the insurance industry. Biener et al. (2017) also use total assets, invested assets, and premium income, among other variables. They study the production efficiency of reinsurance firms. Colquitt and Hoyt (1997) use premium income, the ratio of reinsurance ceded to premiums, and

total assets as a scaling, among other variables. Their focus is on the determinants of hedging behavior by life insurers.

The firms and time span are chosen in order to balance the length and width of the final sample. The data are observed at quarterly frequency. Each variable is standardized by the Total Assets to make them comparable across firms of different sizes (except Total Assets itself), demeaned, and standardized by the standard deviation. Remaining missing values are interpolated. The resulting dataset covers 49 firms from 2005 to 2017 (52 quarters). See Table A.1 in the appendix for a list of the companies included.

We apply Algorithm 1, allowing the number of clusters to range from 2 to 10 clusters. Again we choose k-means clustering as a first-pass clustering algorithm. From the diagnostics displayed in Figure 7 we make a first decision on the value of the shrinkage parameter ε. This is further corroborated in the analysis below. We clearly see a trade-off between the fit as measured by the Gini-weighted silhouette statistic, and the number of switches. A first slight decrease in fit of the model as measured by the Gini-weighted silhouette starts setting in around $\varepsilon > 0.45$, and a second, sharper one around $\varepsilon > 0.65$. The flatness of the silhouette curve before this point indicates that we can increase the stickiness parameter with little consequence to the cross-sectional clustering fit. It is also important to note that at $\varepsilon = 0.6$, less than one third of the unconstrained switches are left. In fact, the switching rate drops quickly between $\varepsilon = 0.3$ to 0.6 without a serious effect on fit. The flexibility in terms of switches that is left in the model seems to be important: beyond $\varepsilon = 0.65$ the fit drops much more sharply We conclude that at $\varepsilon \approx 0.6$, we have eliminated most of the flickering, while the economically important switches remain.

The pattern and density of cluster switches resulting from the choice of $\varepsilon = 0.6$ corroborates the above decision. Figure 8 illustrates how successful the method is in sifting out flickering, i.e. economically implausibly frequent switches. The figure shows each switch in our dataset for two different values of ε. The colored squares represent switches of unit $i = 1, \ldots, N$ on the vertical axis at each time t on the horizontal axis. The color of a square indicates the cluster number unit i switches into at time t: common colors indicate that many observations switch into the same existing or new cluster.

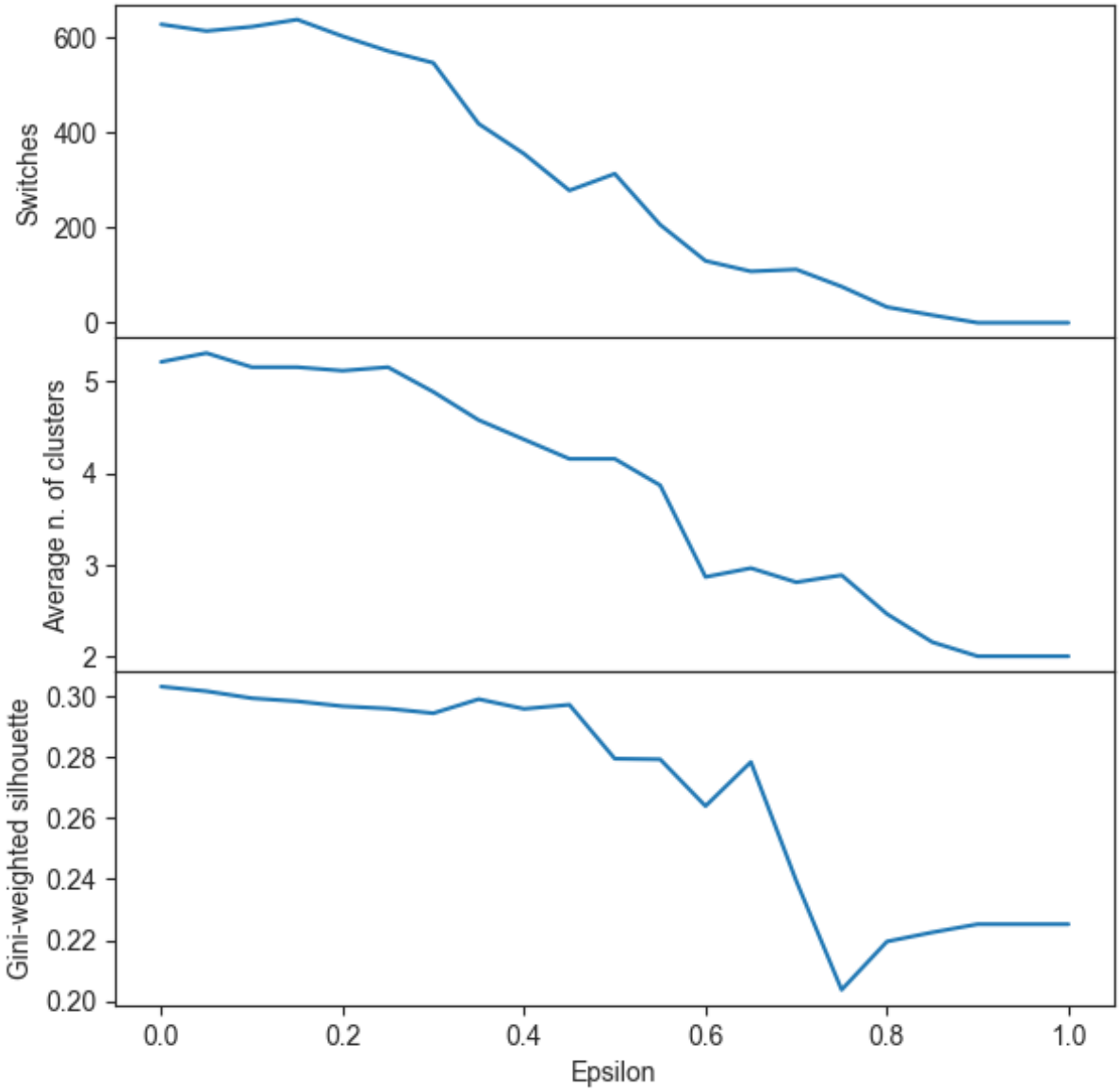
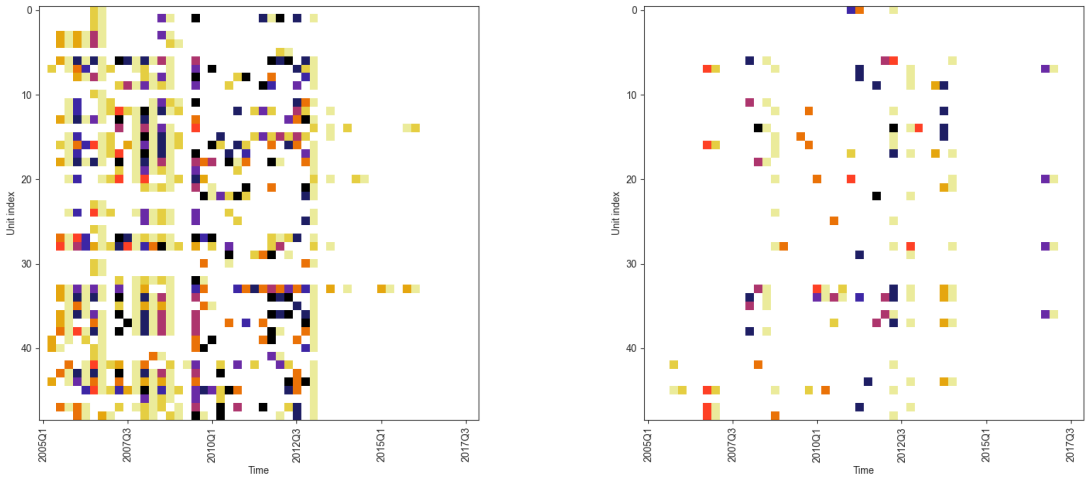


Figure 7: Diagnostics of the clustering for several values of ε: total number of cluster switches; average number of clusters across time; and average Gini-weighted silhouette.

At low levels of stickiness, a plaid-like pattern appears: vertically aligned switches caused by mergers and splits of clusters occur frequently. Also, many units show flickering behavior: at a given vertical level of the graph long sequences of boxes of different, sometimes alternating colors occur, indicating unit i switches into a specific cluster at time t, but back to its former cluster at time $t+1$. Note that we do not even display the most drastic version of these plots: the left-hand plot is for $\varepsilon = 0.2$ rather than for the unconstrained switching case $\varepsilon = 0$.

By increasing ε to 0.6 we see that the number of switches is considerably scaled back.



(a) $\varepsilon = 0.2$ (b) $\varepsilon = 0.6$

Figure 8: Heatmap of switches for values of ε. Each colored square represents a switch of the row unit at the column quarter. Colors represent the destination cluster.

As mentioned before, this is achieved without substantially deteriorating the model's fit as measured by the Gini-weighted silhouette. Still, the right-hand figure still shows a non-negligible number of switches. Partly, this may be real business model switches, and partly this may be companies that are hard to clasify to any of the neighboring clusters. In any case, we see that we have largely succeeded in obtaining dense rows of alternating colored boxes, which would be symptomatic for flickering. Increasing ε further would result in even fewer switches, but as Figure 7 shows this would entail a sharp drop in model fit. We conclude that choosing ε beyond 0.6 would be too constraining for the current dataset.

We also assess our choice of ε by looking at dynamics of the clusters. Figure 9 shows the results for two values of ε. Again, we refrain from using the worst case of $\varepsilon = 0$ and start at $\varepsilon = 0.4$ (left), and $\varepsilon = 0.6$ (right). Each cluster at each time is represented by a point. Transitions from one cluster to the next are represented by colored lines. The color of each line indicates the fraction of the cluster's firms at time t that transitioned into another cluster at time $t + 1$. Fractions below 40% are omitted. Flickering can be identified here as a noise in the form of transitions back and forth across clusters. These are clearly present in the left figure.

Increasing the penalty parameter to $\varepsilon = 0.6$ without substantially affecting the model's

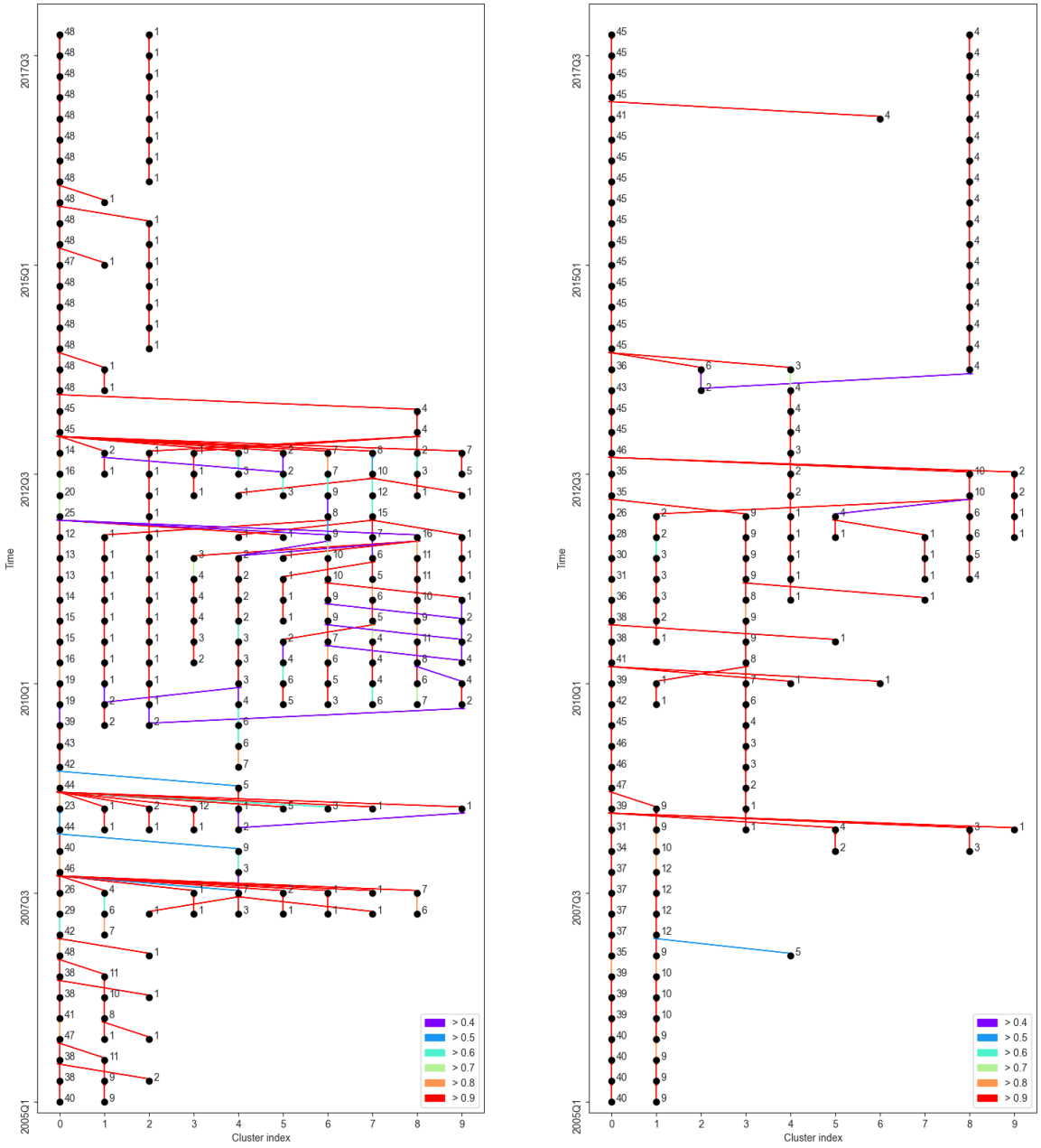


Figure 9: Cluster transition dynamics at $\varepsilon = 0.4$ (left) and $\varepsilon = 0.6$ (right). Points represent clusters identities (horizontal axis) that exist at a specific moment in time t (vertical axis). Numbers indicate the number of firms in a cluster. Lines are colored with the empirical frequency of transitions from this cluster to the next cluster. Transition frequencies below 40% are not indicated.

fit results in a much more interpretable picture, where clusters persists over much longer periods, and can therefore much better reflect the dynamics of the industry over this longer and turbulent time-span. We see one large persistent cluster across all time points comprising most units at any given time. Alongside, a secondary cluster can be seen, which changes composition across time. This indicates the existence of concurrent business

models that are clearly separated from the main body of companies. The clusters do not appear to align with standard industry classifications such as the NAICS or the GIC. Particularly in the aftermath of the great financial crisis and during the time of the European sovereign debt crisis, we see the number of clusters fanning out, implying that insurance firm behavior becomes more dissimilar in the cross-section. After this period, the industry seems to return to is former equilibrium state of a large main cluster, flanked by a secondary cluster. The homogeneity in the industry, however, appears to have somewhat increased: the size of the main cluster is has grown compared to the secondary cluster, indicating that in terms of the measured financials the insurance has become somewhat more homogeneous since the financial and sovereign debt crisis and in the onset of the new Solvency II regulatory framework at the end of the sample. Again we remark that many of these industry dynamics are lost if the shrinkage parameter is set in a non-data-driven way at too high levels such as $\varepsilon = 0.8$.

4 Conclusion

In this paper, we propose a new approach to clustering in a panel setting, allowing for dynamics in the cluster location, cluster composition and number of clusters, while ensuring stability and persistence of assignments via a penalty parameter. To implement the approach, we provide a simple method to map cluster labels from one time point to the next. The method is widely applicable, as it may be used to extend any cross-sectional clustering algorithm that produces a distance measure, including, for instance, k-means, k-medians, or hierarchical clustering. We show how the penalty parameter can be chosen in a data-driven way with a simple weighted version of the well-known silhouette index, and illustrate the good performance of the method in a controlled simulation setting. An application to business models in the global insurance sector underlines the usefulness of our method in balancing flexibility, i.e. allowing for cluster transitions, against penalizing excessive back-and-forth switching between clusters in economic settings.

Notes

[1] Of course, the procedure of candidate clustering, mapping, shrinking, and reassignment, can be iterated if so desired.

Figure legends

Figure 1: k-means clustering at two consecutive times. The red circles represent the location of the cluster centers. The blue line separates the clusters and is halfway between both cluster centers.

Figure 2: We can interpret $\tilde{x}_{i,t}$ as an artificial position lying an ε fraction of the way from the current position $x_{i,t}$ and the center of its last cluster $h_{i,t-1}$

Figure 3: Simulation results for four values of p. Baseline setting.

Figure 4: Simulation results for four values of p. Half-variance setting.

Figure 5: Simulation results for four values of p. Benchmark setting. The number of clusters k may vary between 2 to 4, while the true number is always 2.

Figure 6: Simulation results for four values of p. Half-variance setting. The number of clusters k may vary between 2 to 4, while the true number is always 2.

Figure 7: Diagnostics of the clustering for several values of ε: total number of cluster switches; average number of clusters across time; and average Gini-weighted silhouette.

Figure 8: Heatmap of switches for values of ε. Each colored square represents a switch of the row unit at the column quarter. Colors represent the destination cluster.

Figure 9: Cluster transition dynamics at $\varepsilon = 0.4$ (left) and $\varepsilon = 0.6$ (right). Points represent clusters identities (horizontal axis) that exist at a specific moment in time t (vertical axis). Numbers indicate the number of firms in a cluster. Lines are colored with the empirical frequency of transitions from this cluster to the next cluster. Transition frequencies below 40% are not indicated.

Funding

This work was supported by the Dutch National Science Foundation (NWO) [406.18.EB.011 to I.C.J. and A.L., VI.VIDI.191.169 to J.S].

References

Ayadi, R., P. Bongini, B. Casu, and D. Cucinelli (2021). Banks' Business Model Migrations in Europe: Determinants and Effects. *British Journal of Management*, 1–20.

Ayadi, R., W. P. De Groen, I. Sassi, W. Mathlouthi, H. Rey, and O. Aubry (2016). Banking Business Models Monitor 2015 Europe. *International Research Centre on Cooperative Finance*.

Biener, C., M. Eling, and R. Jia (2017). The structure of the global reinsurance market: An analysis of efficiency, scale, and scope. *Journal of Banking & Finance 77*, 213–229.

Bonhomme, S. and E. Manresa (2015). Grouped Patterns of Heterogeneity in Panel Data. *Econometrica 83*(3), 1147–1184.

Catania, L. (2021). Dynamic adaptive mixture models with an application to volatility and risk. *Journal of Financial Econometrics*, 1–34.

Colquitt, L. L. and R. E. Hoyt (1997). Determinants of Corporate Hedging Behavior: Evidence from the Life Insurance Industry. *The Journal of Risk and Insurance 64*(4), 649.

David, H. A. (1968). Gini's mean difference rediscovered. *Biometrika 55*(3), 573–575.

Fox, E. B., E. B. Sudderth, M. I. Jordan, and A. S. Willsky (2011). A sticky HDP-HMM with application to speaker diarization. *The Annals of Applied Statistics 5*(2A), 1020–1056.

Frühwirth-Schnatter, S. (2011). Panel data analysis: a survey on model-based clustering of time series. *Advances in Data Analysis and Classification 5*(4), 251–280.

Fu, W. and P. O. Perry (2020). Estimating the Number of Clusters Using Cross-Validation. *Journal of Computational and Graphical Statistics 29*(1), 162–173.

Grundmann, M., V. Kwatra, M. Han, and I. Essa (2010). Efficient hierarchical graph-based video segmentation. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, USA, pp. 2141–2148. IEEE.

Kalnis, P., N. Mamoulis, and S. Bakiras (2005). On Discovering Moving Clusters in Spatio-temporal Data. In C. Bauzer Medeiros, M. J. Egenhofer, and E. Bertino (Eds.), *Advances in Spatial and Temporal Databases*, pp. 364–381. Springer.

Kuhn, H. W. (1955). The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly 2*(1-2), 83–97.

Lucas, A., J. Schaumburg, and B. Schwaab (2019). Bank Business Models at Zero Interest Rates. *Journal of Business & Economic Statistics 37*(3), 542–555.

Lucas, A., J. Schaumburg, and B. Schwaab (2020). Dynamic clustering of multivariate panel data. *Tinbergen Institute Discussion Paper 2020-009/III*.

Oliveira, M. and J. Gama (2010). Bipartite Graphs for Monitoring Clusters Transitions. In P. R. Cohen, N. M. Adams, and M. R. Berthold (Eds.), *Advances in Intelligent Data Analysis IX*, pp. 114–124. Springer.

Oliveira, M. and J. Gama (2012). A framework to monitor clusters evolution applied to economy and finance problems. *Intelligent Data Analysis 16*(1), 93–111.

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics 20*, 53–65.

Ward, J. H. (1963). Hierarchical Grouping to Optimize an Objective Function. *Journal of the American Statistical Association 58*(301), 236–244.

Zahn, C. (1971). Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters. *IEEE Transactions on Computers C-20*(1), 68–86.

A Additional figures and tables

Table A.1: List of firms in the dataset and their cluster assignments at $\varepsilon = 0.6$ at selected quarters.

Period	2007Q3	2010Q1	2012Q3	2017Q3
Firm				
ADAMJEE INSURANCE CO LTD	1	0	8	0
ADMIRAL GROUP PLC	0	0	0	0
AGEAS SA/NV	0	0	0	0
AIA INSURANCE LANKA PLC	0	0	0	0
AL AHLEIA INSURANCE CO	0	0	0	0
AL AIN AHLIA INSURANCE CO	0	0	0	0
ALLIANZ SE	0	0	0	0
ALM BRAND A/S	0	0	0	0
ANADOLU SIGORTA	1	3	4	0
ARAB INSURANCE GROUP (BSC)	0	0	4	0
ATLAS INSURANCE LTD	0	0	8	0
BAHRAIN & KUWAIT INSURANCE	0	6	8	0
BANGKOK UNION INSURANCE PCL	0	0	8	0
CENTURY INSURANCE CO LTD	1	0	0	0
CEYLINCO INSURANCE CO LTD	1	0	0	0
CHESNARA PLC	0	0	0	0
CLAL INSURANCE ENTRPRS HLDGS	0	0	0	0
DHIPAYA INSURANCE PCL	0	0	0	0
DHOFAR INSURANCE CO	0	4	8	0
DIRECT INSURANCE - IDI INS	0	0	0	0
DOHA INSURANCE	0	0	8	0
DUBAI INSURANCE CO (P.S.C.)	0	0	0	0
ECCLESIASTICAL INSURANCE OFF	0	0	8	0
EFU GENERAL INSURANCE LTD	1	0	0	0
EFU LIFE ASSURANCE	1	3	0	0
HABIB INSURANCE	0	3	0	0

Continued on next page

Table A.1: List of firms in the dataset and their cluster assignments at $\varepsilon = 0.6$ at selected quarters.

Period	2007Q3	2010Q1	2012Q3	2017Q3
Firm				
HAREL INSURANCE INVESTMENTS	0	0	0	0
JUBILEE GENERAL INSURANCE CO	0	0	0	0
JUBILEE LIFE INSURANCE CO LT	1	3	0	0
LIPPO GENERAL INSURANCE TBK	0	0	9	0
MUNICH RE CO	0	0	0	0
MUSCAT INSURANCE CO SAOG	0	0	8	0
NAM SENG INSURANCE PCL	1	3	0	8
NAVAKIJ INSURANCE PLC	0	0	8	8
OMAN INSURANCE CO PSC	0	0	0	0
OMAN UNITED INSURANCE CO	1	1	0	0
PANIN INSURANCE TBK (PT)	0	0	9	8
QATAR INSURANCE CO	0	0	0	0
SINGAPORE REINSURANCE CORP	0	0	0	0
SOC CATTOLICA ASSICURAZIONI	0	0	0	0
STOREBRAND ASA	0	0	0	0
THAI REINSURANCE PCL	1	3	0	0
THAI SETAKIJ INSURANCE PCL	1	0	8	0
THAIVIVAT INSURANCE PCL	1	3	0	8
TOPDANMARK AS	0	0	0	0
TRYG AS	0	0	0	0
UNION ASSURANCE LTD	0	0	0	0
UNITED OVERSEAS INSURANCE	0	0	0	0
ZURICH INSURANCE GROUP AG	0	0	0	0