Automatic Detection of Cluster Structure Changes using Relative Density Self-Organizing Maps

Denny[1] **Pandu Wicaksono**[1] **Ruli Manurung**[1]

[1]Faculty of Computer Science, University of Indonesia, Indonesia
denny@cs.ui.ac.id, pandu.wicaksono91@ui.ac.id, maruli@cs.ui.ac.id

Abstract

Knowledge of clustering changes in real-life datasets is important in many contexts, such as customer attrition analysis and fraud detection. Organizations can use such knowledge of change to adapt business strategies in response to changing circumstances. Analysts should be able to relate new knowledge acquired from a newer dataset to that acquired from an earlier dataset to understand what has changed. There are two kind of clustering changes, which are: changes in clustering structure and changes in cluster memberships. The key contribution of this paper is a novel method to automatically detect structural changes in two snapshot datasets using ReDSOM. The method identifies emerging clusters, disappearing clusters, splitting clusters, merging clusters, enlarging clusters, and shrinking clusters. Evaluation using synthetic datasets demonstrates that this method can identify automatically structural cluster changes. Moreover, the changes identified in our evaluation using real-life datasets from the World Bank can be related to actual changes.

Keywords: temporal clustering, self-organizing maps, visualization.

1 Introduction

Clusters provide insights into archetypical behaviours across a population, for example from taxation records, insurance claims, customer purchases, and medical histories. A cluster is a set of similar observations of entities, but these observations are dissimilar to observations of entities in other clusters (Han et al. 2011). The process of assignment of these observations in a dataset into clusters based on similarity is called as cluster analysis (Jain et al. 1999). Clustering is an exploratory data analysis technique that aims to discover the underlying structures in data.

In order to respond to change, we need to be able to identify and understand change. One way to do so is by looking at changes of clusters in terms of their structure and memberships. This type of knowledge can help organizations develop strategies, such as in fraud detection and in customer attrition analysis. Moreover, analysts often have to understand change from two datasets acquired at two different points in time to adapt existing business strategies.

This paper presents a novel method that automatically detect changes in cluster structure from a clustering result $\mathcal{C}(\tau_1)$ obtained from dataset $\mathcal{D}(\tau_1)$ observed at time period τ_1 compared to clustering result $\mathcal{C}(\tau_2)$ obtained from dataset $\mathcal{D}(\tau_2)$, where $\tau_1 < \tau_2$. To understand what has changed, analysts need to relate new knowledge (often represented as models) acquired from a newer dataset $\mathcal{D}(\tau_2)$ to that acquired from an earlier dataset $\mathcal{D}(\tau_1)$.

In this paper, the clustering structure and the cluster assignment from a clustering result $\mathcal{C}(\tau_i)$ are defined as follow. The clustering structure of a clustering result is the shapes, densities, sizes, locations of each clusters, including similarity and distances between clusters. This structure also includes the partitioning of the data space $\mathbb{R}^d$ into Voronoi regions. On the other hand, the cluster assignments are assignments of each data vector in the dataset to a cluster. In other words, it is a partitioning of a set of data vectors into k non-overlapping and collectively exhaustive subsets.

In order to analyze clustering changes, this paper uses the ReDSOM method (Denny et al. 2010) to compare two Self-Organizing Maps $\mathcal{M}(\tau_1)$ and $\mathcal{M}(\tau_2)$ trained from two snapshot datasets $\mathcal{D}(\tau_1)$ and $\mathcal{D}(\tau_2)$. In Denny et al. (2010), changes in cluster structure are identified through visualizations by analysts. On the other hand, this paper aims to automatically detect structural cluster changes.

This paper compares SOMs since the SOMs capture the clustering structure of the underlying datasets. Changes between two related datasets can be discovered by comparing the resulting data mining models since each model captures specific characteristics of the respective dataset as in FOCUS framework (Ganti et al. 2002) and in PANDA framework (Bartolini et al. 2009). This approach is also called "contrast mining" or "change mining" (Boettcher 2011).

Most temporal clustering algorithms consider clustering of sequences of events or clustering of time series (Antunes & Oliveira 2001, Roddick et al. 2001, Roddick & Spiliopoulou 2002). The goals of this paper are different to the aims of time series clustering and clustering of sequences. Sequence clustering aims to group objects based on sequential structural characteristics. Time-series clustering, on the other hand, aims to cluster individual entities that have similar time-series patterns to discover and describe common trends in time series (Roddick & Spiliopoulou 2002). In contrast, this paper considers the clustering of observations of entities at points in time, and compares the clustering structures from *snapshot datasets* derived from *longitudinal data*. A snapshot dataset $\mathcal{D}(\tau_j)$ contains an observation of each entity $\mathcal{I}_i$ at one time period τ_j from a longitudinal data.

Entity $\mathcal{I}_i$ is defined as a *subject of interest*. For example, an entity can be a tax payer, a country, or a customer. An observation of entity $\mathcal{I}_i$ at time period τ_j is the measurements of entity $\mathcal{I}_i$ at time period τ_j based on attributes/features. These measurements are represented as data vector $\mathbf{x}_i(\tau_j)$. Longitudinal data, often referred to as *panel data*, are collection of repeated observations $\mathbf{x}_i(\tau_1), \ldots, \mathbf{x}_i(\tau_t)$ at multiple time periods $\tau_1, \ldots, \tau_t$ that track the same type of information/observation on the same set of entities $\mathcal{I}_i$ (Diggle et al. 1994). In other words, there are a number of observations associated for the same entity. An example of longitudinal data is various indicators for each country that are collected regularly by the World Bank. From these data, the snapshot datasets are the welfare condition of countries that are observed in the 1980s as one snapshot and the 1990s as the subsequent snapshot. These two observations of a country are represented as two data vectors $\mathbf{x}_i(\tau_1)$ and $\mathbf{x}_i(\tau_2)$. Analysis of change from longitudinal data leads to quite a different approach to the process of cluster analysis.

The remainder of the paper is organized as follows. The next section discusses related works in temporal cluster analysis. Sections 3 then reviews structural cluster changes detection and the relative density definition. The contribution of this paper is discussed in Section 4. Section 5 discusses our experiments on the threshold parameters used in our algorithm. The application of the algorithm with synthetic amd real-life datasets is then discussed in Section 6. Conclusions and future work are provided in Section 7.

2 Related Works

Existing methods can be differentiated based on the types of data they can handle, which are: data stream, partitioned dataset, snapshot longitudinal, univariate time series, and trajectories. Clustering snapshot datasets has not received much attention in temporal clustering. Research has focused mostly on clustering of sequences, time series clustering, data stream clustering, and trajectory clustering. The ReDSOM method clusters snapshot datasets and can contrast the clustering results between two snapshots (Denny et al. 2010). This method can also be used for multivariate time series data once transformed into snapshot datasets. However, ReDSOM does not detect structural changes automatically.

In identifying structural changes, there are a number of different approaches. MONIC (Spiliopoulou et al. 2006) and MClusT/MEC (Oliveira & Gama 2010) define clusters as set of objects. Therefore, structural changes is defined based on overlap of cluster members between two clusters of two time periods. Furthermore, MONIC uses ageing of observations to monitor evolutions of clusters which is appropriate for clustering data stream. In contrast, this paper uses snapshots datasets to analyze changes of clustering over time. MONIC+ (Ntoutsi et al. 2009) tries to generalize MONIC to include more cluster types which are clusters as geometrical objects and as distribution. When clusters are defined as geometrical objects, cluster overlap is defined by intersection of area of the two clusters. It is not clear how area is defined in MONIC+, especially for high dimensional datasets. On the other hand, this paper defines cluster overlap by the intersection of the Voronoi region of two clusters. Adomavicius & Bockstedt (2008) uses between-cluster distances to detect structural changes. Kalnis et al. (2005) uses moving cluster, which is defined based on a set of objects and spatial location. ReDSOM and Aggarwal (2005) use kernel density estimation to detect structural changes. However, the work in Aggarwal (2005) is designed for stream clustering and the concept of velocity density estimation is limited to one attribute. ReDSOM, on the other hand, is used to analyze multivariate snapshot datasets by comparing density estimation and communicate the results using visualization. Recently, Held & Kruse (2013) presented a method based on MONIC to visualize the dynamics of cluster evolution. The method were extended to detect cluster rebirth, which is a missing cluster in the previous time period that is emerged again.

The SOM-based methods to analyze cluster using multiple temporal datasets can be categorized into three approaches: chronological, temporal, and sequential cluster analysis. *Chronological cluster analysis* produces a different SOM and visualizations for each time period (Skupin & Hagelman 2005). *Temporal cluster analysis*, on the other hand, trains a single SOM with combined data from all time period (Skupin & Hagelman 2005). The limitation of this approach is its inability to detect structural clustering changes. *Sequential cluster analysis*, such as SbSOM (Fukui et al. 2008), trains a single SOM with sequential data. This paper uses chronological cluster analysis approach (Denny & Squire 2005). Given two snapshot datasets $\mathcal{D}(\tau_1)$ and $\mathcal{D}(\tau_2)$, the maps $\mathcal{M}(\tau_1)$ and $\mathcal{M}(\tau_2)$ are trained using their respective datasets. When training map $\mathcal{M}(\tau_2)$, the map $\mathcal{M}(\tau_1)$ is used as the initial map to preserve the orientation of the trained map.

3 Structural Cluster Changes Detection using ReDSOM

In ReDSOM, changes of clustering structure are identified from changes of density estimations at the same location over time. The plot of density estimation can provide useful characteristics in the data, such as skewness, multi-modality, and clustering structure. Density estimation is the construction of an estimate $\hat{f}(x)$ of an unobservable underlying density function $f(x)$ based on observed data (Silverman 1986). This estimation is also ideal to present the data to non-mathematicians, as it is fairly easy to understand (Silverman 1986). Dense regions in data space are good candidates for clusters. Conversely, very low density regions most likely contain outliers. Therefore, clusters can be found by density estimates, such as in the DENCLUE algorithm (Hinneburg & Keim 2003).

Kernel density estimation (KDE) is a non-parametric method to estimate the probability density function of the observed data at a given point (Silverman 1986). KDE is a non-parametric method, as it does not make assumption about the distribution of the observed data. This method is also known as *Parzen-Rosenblatt* window method (Parzen 1962, Rosenblatt 1956).

The approximation of density estimation can be calculated faster using prototype vectors produced by a VQ method (Hinneburg & Keim 2003, Macqueen 1967). Since SOM is also a VQ method, the prototype vectors of map $\mathcal{M}$ that is trained on dataset $\mathcal{D}$ can be used to estimate density of dataset $\mathcal{D}$. Data vector $\mathbf{x}_i \in \mathcal{D}$ can be approximated/represented using the closest prototype vector $\mathbf{m}_{b_i}$ with the accuracy measured by quantization error. In other words, the data space $\mathbb{R}^d$ is partitioned into $|\mathcal{M}|$ Voronoi regions without leaving any gaps or overlaps. A Voronoi region VR_j of a prototype vector $\mathbf{m}_j$ or cluster $C_j \in \mathcal{C}$ is

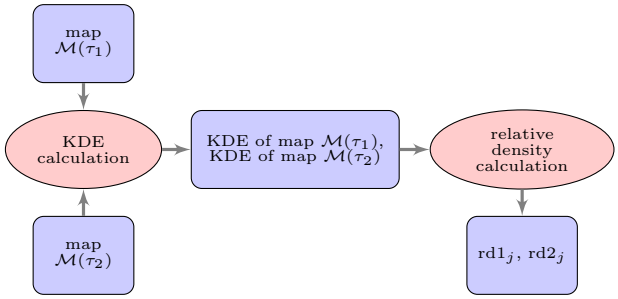


Figure 1: Relative density calculation

defined as the set of all points in $\mathbb{R}^d$ that are closest to the cluster centroid/prototype vector $\mathbf{m}_j$. Each partition contains the data vectors that are the nearest to its partition prototype vector compared to other prototype vectors on the map (Voronoi set).

This research uses two-level clustering that uses a SOM as an abstraction layer of the dataset (Vesanto & Alhoniemi 2000). The prototype vectors are clustered using a partitional clustering technique or a hierarchical agglomerative nesting (AGNES) technique to form the final clusters. The optimal clustering results are then selected based on cluster validity indexes.

When comparing maps $\mathcal{M}(\tau_1)$ and $\mathcal{M}(\tau_2)$, density estimation $\hat{f}_{h,\mathcal{M}(\tau_1)}(\mathbf{v})$ centred at the location of vector $\mathbf{v} \in \mathbb{R}^d$ on map $\mathcal{M}(\tau_1)$ might be different compared to density estimation $\hat{f}_{h,\mathcal{M}(\tau_2)}(\mathbf{v})$ at the same location on map $\mathcal{M}(\tau_2)$. When the density centred at the location of vector $\mathbf{v}$ in dataset $\mathcal{D}(\tau_2)$ is lower than in dataset $\mathcal{D}(\tau_1)$, the density estimation $\hat{f}_{h,\mathcal{M}(\tau_2)}(\mathbf{v})$ on map $\mathcal{M}(\tau_2)$ centred at the location of vector $\mathbf{v}$ is lower compared to the density estimation $\hat{f}_{h,\mathcal{M}(\tau_1)}(\mathbf{v})$ at the same location on map $\mathcal{M}(\tau_1)$, and vice-versa. Therefore, relative density $\mathrm{RD}_{\mathcal{M}(\tau_2)/\mathcal{M}(\tau_1)}(\mathbf{v})$ is defined as the log ratio of the density estimation centred at the location of vector $\mathbf{v}$ on map $\mathcal{M}(\tau_2)$ to the density estimation centred at the same location on the reference map $\mathcal{M}(\tau_1)$ (Denny et al. 2010):

$$\mathrm{RD}_{\mathcal{M}(\tau_2)/\mathcal{M}(\tau_1)}(\mathbf{v}) = log_2\left(\frac{\hat{f}_{h,\mathcal{M}(\tau_2)}(\mathbf{v})}{\hat{f}_{h,\mathcal{M}(\tau_1)}(\mathbf{v})}\right) \quad (1)$$

Let $\mathrm{rd1}_j \leftarrow \mathrm{RD}_{\mathcal{M}(\tau_2)/\mathcal{M}(\tau_1)}(\mathbf{m}_j(\tau_1))$ as a shorthand for the relative density at the location of prototype vector $\mathbf{m}_j(\tau_1)$ on map $\mathcal{M}(\tau_2)$ compared to reference map $\mathcal{M}(\tau_1)$. Similarly, let $\mathrm{rd2}_j \leftarrow \mathrm{RD}_{\mathcal{M}(\tau_2)/\mathcal{M}(\tau_1)}(\mathbf{m}_j(\tau_2))$ be the relative density at the location of prototype vector $\mathbf{m}_j(\tau_2)$ on map $\mathcal{M}(\tau_2)$ compared to reference map $\mathcal{M}(\tau_1)$. Figure 1 shows the relative density calculation.

To visualize the relative density of the locations of all prototype vectors $\mathbf{m}_j(\tau_1)$, the values of $\mathrm{rd1}_j$ are visualized on a map $\mathcal{M}(\tau_1)$ in a gradation of blue for positive values and red for negative values. ReDSOM visualization uses diverging colour scheme as this visualization has a critical mid-point which is zero (no change in density). Values of relative density over $+\delta$ are represented as dark blue, and values less than $-\delta$ are represented as dark red, where δ is a density threshold parameter. Based on experiments, the value of δ is set to 3, which means a region is considered as emerging or disappearing when its density increase by eightfold or one-eightfold, respectively.

Visualization of $\mathrm{rd1}_j$ on map $\mathcal{M}(\tau_1)$ alone cannot be used to detect emerging regions in period τ_2. Regions that emerge in dataset $\mathcal{D}(\tau_2)$ are not represented on $\mathcal{M}(\tau_1)$ because map $\mathcal{M}(\tau_1)$ only represents populated Voronoi regions of dataset $\mathcal{D}(\tau_1)$ due to the VQ property. Therefore, $\mathrm{rd2}_j$ is used to detect emerging regions because the emerging regions at period τ_2 are represented on map $\mathcal{M}(\tau_2)$. The $\mathrm{rd2}_j$ value for the Voronoi region of prototype vector $\mathbf{m}_j(\tau_2)$ would be high because the density $\hat{f}_{h,\mathcal{M}(\tau_2)}(\mathbf{m}_j(\tau_2))$ is high and the density $\hat{f}_{h,\mathcal{M}(\tau_1)}(\mathbf{m}_j(\tau_2))$ is low. In sum, values of $\mathrm{rd2}_j$ should be visualized on map $\mathcal{M}(\tau_2)$ to detect emerging clusters.

For similar reason, visualization $\mathrm{rd2}_j$ on map $\mathcal{M}(\tau_2)$ alone cannot be used to detect disappearing regions on map $\mathcal{M}(\tau_2)$. Disappearing regions are not represented by prototype vectors of map $\mathcal{M}(\tau_2)$. As a result, visualization of $\mathrm{rd1}_j$ on map $\mathcal{M}(\tau_1)$ is used to detect disappearing regions. The disappearing regions exist on map $\mathcal{M}(\tau_1)$, but no longer exist on map $\mathcal{M}(\tau_2)$.

To discover changes in clustering structure, Denny & Squire (2005) proposed cluster colour linking techniques, which is a techniques to link two clustering results from two snapshot datasets $\mathcal{D}(\tau_1)$ and $\mathcal{D}(\tau_2)$. The *cluster colour linking* technique, then, is used to visualize the first clustering result $\mathcal{CM}(\tau_1)$ on the second map $\mathcal{M}(\tau_2)$. The cluster colour of the nodes of map $\mathcal{M}(\tau_2)$ are determined by the cluster colour of their BMU (best matching unit) on map $\mathcal{M}(\tau_1)$. Given the colour of node j on map $\mathcal{M}(\tau_1)$ as $nodeCluster(j, \mathcal{M}(\tau_1))$, the colour of node j on map $\mathcal{M}(\tau_2)$ is calculated as:

$$nodeCluster\big(j, \mathcal{M}(\tau_2)\big) = nodeCluster\big(BMU\big(\mathbf{m}_j(\tau_2), \mathcal{M}(\tau_1)\big),\ \mathcal{M}(\tau_1)\big) \quad (2)$$

Since map $\mathcal{M}(\tau_2)$ follows the distribution of dataset $\mathcal{D}(\tau_2)$, the size of each cluster on map $\mathcal{M}(\tau_2)$ would follow as well.

A cluster $C_i(\tau_2) \in \mathcal{C}(\tau_2)$ is said to have emerged at time period τ_2 when the density of the cluster $C_i(\tau_2)$ occupies a well separated region that has significantly increased density in dataset $\mathcal{D}(\tau_2)$ ($\mathrm{rd2}_j \geq +\delta$) compared the region's density in the previous dataset $\mathcal{D}(\tau_1)$.

$$\frac{\big|\{\mathbf{m}_j(\tau_2) \in C_i(\tau_2) \mid \mathrm{rd2}_j \geq +\delta\}\big|}{\big|C_i(\tau_2)\big|} \geq \theta_{\text{emerging}} \quad (3)$$

A cluster $C_i(\tau_1) \in \mathcal{C}(\tau_1)$ is said to have disappeared at time period τ_2 when the density of region $\mathbf{m}_j(\tau_1) \in C_i(\tau_1)$ is significantly decreased ($\mathrm{rd1}_j \leq -\delta$) in the dataset $\mathcal{D}(\tau_2)$ compared to the previous dataset $\mathcal{D}(\tau_1)$.

$$\frac{\big|\{\mathbf{m}_j(\tau_1) \in C_i(\tau_1) \mid \mathrm{rd1}_j \leq -\delta\}\big|}{\big|C_i(\tau_1)\big|} \geq \theta_{\text{disappearing}} \quad (4)$$

Unlike a new cluster that resides in a previously unoccupied region, split clusters do not occupy a new region. A cluster split can be identified when a cluster in map $\mathcal{M}(\tau_1)$ can be separated in map $\mathcal{M}(\tau_2)$. ReDSOM visualization has to show that both split clusters in period τ_2 do not occupy new region ($0 \leq \mathrm{rd2}_j \leq \delta$). A cluster $C_i(\tau_1)$ is said to have split at time period τ_2 when the Voronoi region of cluster $C_i(\tau_1)$ is occupied by two or more well separated clusters $C_{k1}(\tau_2), \ldots, C_{kn}(\tau_2)$ in the dataset $\mathcal{D}(\tau_2)$.

Cluster merging occurs when two clusters on map $\mathcal{M}(\tau_1)$ are no longer well separated on map $\mathcal{M}(\tau_2)$. Cluster merging can be identified by cluster colour linking when two clusters on map $\mathcal{M}(\tau_1)$ are merged into one cluster outline on map $\mathcal{M}(\tau_2)$. Cluster merging is different to lost cluster where one of the clusters shrinks significantly thus having $\mathrm{rd1}_j < -\delta$. In cluster merging, the density of gap between clusters should increased in a way that can be verified using ReDSOM visualization. Clusters $C_{i1}(\tau_1), \ldots, C_{in}(\tau_1)$ are said to have merged into $C_k(\tau_2)$ at time period τ_2 when the gap between the clusters is disappear in the dataset $\mathcal{D}(\tau_2)$.

Cluster $C_i(\tau_2)$ is said to have enlarged at time period τ_2 when the part of the cluster region has significantly increased density in the dataset $\mathcal{D}(\tau_2)$.

$$\theta_{\text{overlap}} \leq \frac{\left|\left\{\mathbf{m}_j(\tau_2) \in C_i(\tau_2) \mid \mathrm{rd2}_j \geq \delta \right\}\right|}{\left|C_i(\tau_2)\right|} < \theta_{\text{emerging}} \quad (5)$$

Similarly, cluster contraction can be identified as a lost region which does not have a good separation to its neighbours. To put this another way, only a part of a cluster has disappeared. Cluster $C_i(\tau_1)$ is said to have contracted at time period τ_2 when the clusters occupies smaller region in the dataset $\mathcal{D}(\tau_2)$.

$$\theta_{\text{overlap}} \leq \frac{\left|\left\{\mathbf{m}_j(\tau_1) \in C_i(\tau_1) \mid \mathrm{rd1}_j \leq -\delta \right\}\right|}{\left|C_i(\tau_1)\right|} < \theta_{\text{disappearing}} \quad (6)$$

If a cluster does not fall into the above categories, the cluster $C_i(\tau_1)$ is evaluated wheter it is overlapped with another cluster $C_j(\tau_2)$. As above, the overlap is determined based on the Voronoi region of their prototype vectors.

4 Automatic Structural Changes Detection

Based on the ReDSOM reviewed in the previous section, this paper develops a new algorithm that detects the structural cluster changes based on the relative density measurements. The algorithm takes the clustering results, relative density measurements, and hit counts to detect structural changes as shown in Algorithm 1 and Figure 2. In general, the steps are: initializations, detecting disappearing and contracting clusters, detecting emerging and enlarging clusters, detecting merging clusters, detecting splitting clusters, and detecting overlapping clusters.

There are two ways to calculate the number of prototype vector members that disappear ($totalDarkRedRegionT1$), emerge ($totalDarkBlueRegionT2$), or overlap: weighted and unweighted calculations. Unlike the unweighted calculation where each prototype vector counts as one, in the weighted calculation, each prototype vector counts as the number of data vectors mapped to the prototype vector (hit count). Algorithm 2 shows the initialization of the algorithm.

The algorithm starts by discovering disappearing and emerging clusters (Algorithms 3 and 4). The ratio is calculated using $totalDarkRedRegionT1$ and $totalDarkBlueRegionT2$. When the ratio of the 'dark red' region in a cluster $C_i(\tau_1)$ is compared to the whole cluster $C_i(\tau_1)$ above $\theta_{\text{disappearing}}$, the cluster is considered to disappear in period τ_2. Similarly, when the ratio of the 'dark blue' region in a cluster $C_j(\tau_2)$ is compared to the whole cluster $C_j(\tau_2)$ above

Algorithm 1: Automatic structural cluster changes detection algorithm.

Input: array of cluster assignment in τ_1 CT1, array of cluster assignment in τ_2 CT2, array of cluster color linking assignment CT21, array of relative density rd1 and rd2, array of hit count in τ_1 HT1, array of hit count in τ_2 HT2
Output: List of structural changes LC for all cluster in τ_1 and τ_2

1 **initialization** (Algorithm 2)
2 **detect disappearing and contracting clusters** (Algorithm 3)
3 **detect emerging and enlarging clusters** (Algorithm 4)
4 **detect merging clusters** (Algorithm 5)
5 **detect splitting clusters** (Algorithm 6)
6 **detect overlapping clusters** (Algorithm 7)
7 **return** LC

Algorithm 2: Initialization.

```
LC ← ∅
listUnknownChangeT1 ← {C_1(τ_1), ..., C_numberClusterT1(τ_1)}
listUnknownChangeT2 ← {C_1(τ_2), ..., C_numberClusterT2(τ_2)}
contractingClusterT1 ← ∅
enlargingClusterT2 ← ∅
for j = 0 to |M| do
    if weighted then
        weightT1[j] ← HT1[j]
        weightT2[j] ← HT2[j]
    else
        weightT1[j] ← 1
        weightT2[j] ← 1
    totalClusterMemberT1[CT1[j]] += weightT1[j]
    totalClusterMemberT2[CT2[j]] += weightT2[j]
    totalClusterMemberT21[CT21[j]] += weightT2[j]
    if rd1_j ≤ −3 then
        totalDarkRedRegionT1[CT1[j]] += weightT1[j]
    if rd2_j ≥ 3 then
        totalDarkBlueRegionT2[CT2[j]] += weightT2[j]
```

θ_{emerging}, the cluster is considered to emerge in period τ_2. If a cluster has some 'dark red' or 'dark blue' region, the cluster is flagged as contracting and enlarging, respectively. This kind of cluster changes still needs to be checked further if the cluster participates in cluster splitting or cluster merging.

In the second phase, the clusters $C_i(\tau_1)$ and $C_j(\tau_2)$ are checked whether they experience splitting or merging (Algorithms 5 and 6). When two or more clusters $C_{i_1}(\tau_1), C_{i_2}(\tau_1), \ldots, C_{i_n}(\tau_1)$ from period τ_1 overlaps with the region of cluster $C_j(\tau_2)$, the clusters $C_{i_1}(\tau_1), C_{i_2}(\tau_1), \ldots, C_{i_n}(\tau_1)$ are said to merge into cluster $C_j(\tau_2)$. The overlap between cluster $C_i(\tau_1)$ and $C_j(\tau_2)$ is defined as the ratio of the Voronoi region of $C_i(\tau_1)$ which overlaps/intersects with the Voronoi region of $C_j(\tau_2)$ to the whole cluster $C_{i_1}(\tau_1)$. When this overlap is above the θ_{merging} threshold, most of

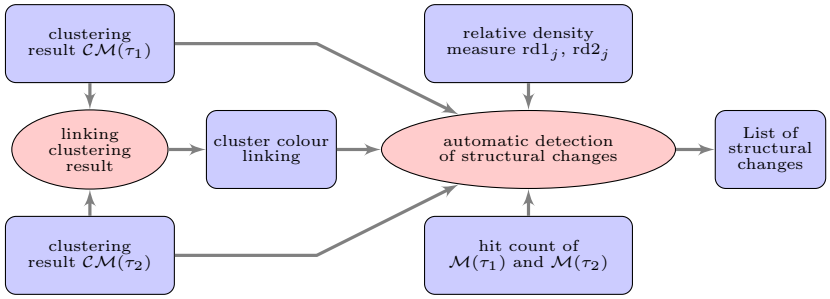


Figure 2: Automatic detection of changes in cluster structure.

Algorithm 3: Detecting disappearing and contracting cluster.

```
1 foreach C_i(τ1) ∈ listUnknownChangeT1 do
2     if totalDarkRedRegionT1[i] > 0 then
3         ratio ← totalDarkRedRegionT1[i] /
          totalClusterMemberT1[i]
4         if ratio ≥ θ_disappearing then
5             LC ← LC
              ∪{(C_i(τ1), ∅, disappearing)}
6             listUnknownChangeT1 ←
              listUnknownChangeT1 − {C_i(τ1)}
7         else
8             contractingClusterT1 ←
              contractingClusterT1 ∪{C_i(τ1)}
```

Algorithm 4: Detecting emerging and enlarging cluster.

```
1 foreach C_j(τ2) ∈ listUnknownChangeT2 do
2     if totalDarkBlueRegionT2[j] > 0 then
3         ratio ← totalDarkBlueRegionT2[j] /
          totalClusterMemberT2[j]
4         if ratio ≥ θ_emerging then
5             LC ← LC ∪{(∅, C_j(τ2), emerging)}
6             listUnknownChangeT2 ←
              listUnknownChangeT2 − {C_j(τ2)}
7         else
8             enlargingClusterT2 ←
              enlargingClusterT2 ∪{C_j(τ2)}
```

Algorithm 5: Detecting merging clusters.

```
  // Detecting overlap between all pair of
     cluster from period τ1 and τ2
1 foreach C_j(τ2) ∈ listUnknownChangeT2 do
2     listOverlapClusterT1 ← ∅
3     foreach C_i(τ1) ∈ listUnknownChangeT1 do
4         overlapCount ← 0
5         for mapUnit = 1 → |M(τ2)| do
6             if CT21[mapUnit] = i ∧
              CT2[mapUnit] = j then
                  // when the map unit of
                     M(τ2) is assigned to
                     both C_i(τ1) and C_j(τ2)
7                 overlapCount +=
                  weightT2[mapUnit]
8         ratio ← overlapCount /
          totalClusterMemberT21[i]
9         if ratio ≥ θ_merging then
10            listOverlapClusterT1 ←
              listOverlapClusterT1 ∪{C_i(τ1)}
11    if |listOverlapClusterT1| ≥ 2 then
12        foreach C_i(τ1) ∈ listOverlapClusterT1
          do
13            LC ← LC
              ∪{(C_i(τ1), C_j(τ2), merging)}
14            listUnknownChangeT1 ←
              listUnknownChangeT1 − {C_i(τ1)}
15        listUnknownChangeT2 ←
          listUnknownChangeT2 − {C_j(τ2)}
```

$C_i(\tau_1)$ is part of $C_j(\tau_2)$. Merging clusters require two or more clusters from period τ_1 that are part of cluster $C_j(\tau_2)$. The overlap is calculated and visualized using the cluster colour linking technique described earlier. Detecting cluster splitting is basically the mirror case of detecting cluster merging.

In the last phase, clusters $C_i(\tau_1)$ and $C_j(\tau_2)$ that are not yet classified are checked if their region partially overlap. If the ratio of overlap above θ_{overlap} and the cluster $C_i(\tau_1)$ was flagged as contracting or enlarging, cluster $C_i(\tau_1)$ is considered to be contracting or enlarging in period τ_2, respectively. Otherwise, the clusters with the ratio of overlap above θ_{overlap} is considered as overlap.

The complexity of the whole algorithm is $\mathcal{O}(|\mathcal{C}(\tau_1)| \cdot |\mathcal{C}(\tau_2)| \cdot |\mathcal{M}|)$, where $|\mathcal{C}(\tau_i)|$ is the number of cluster in period τ_i and $|\mathcal{M}|$ is the number of prototype vectors in map $\mathcal{M}$. The running time for Algorithm 2 is bounded by $\mathcal{O}(|\mathcal{M}|)$. The complexities of Algorithms 3 and 4 are $\mathcal{O}(|\mathcal{C}(\tau_1)|)$ and $\mathcal{O}(|\mathcal{C}(\tau_2)|)$ respectively. The running time for Algorithms 5–7 are bounded by $\mathcal{O}(|\mathcal{C}(\tau_1)| \cdot |\mathcal{C}(\tau_2)| \cdot |\mathcal{M}|)$.

5 Experiments on the Threshold Parameters

To determine the threshold parameters for each type of cluster changes, experiments on different values of these threshold parameters are performed on synthetic datasets. In these synthetic datasets, only one structural cluster change is introduced in the dataset $\mathcal{D}(\tau_2)$. In total, there are eight pairs of synthetic datasets. The threshold values used range between 0.3 and 1.0 in increments of 0.1. When the threshold value is too high, the algorithm cannot detect the changes. On the other hand, when the threshold value is too low, the algorithm might detect false positive changes. While this paper provides the guidelines for setting the value of these threshold parameter, these parameters can be tuned to suit the need of analysis.

These experiments showed that the weighted calculation allows the use of higher threshold values. For the datasets where an emerging cluster exists in the dataset $\mathcal{D}(\tau_2)$, the emerging cluster is no longer detected at $\theta_{emerging} = 0.8$ using the unweighted version. On the other hand, the weighted version can

Algorithm 6: Detecting splitting clusters.

```
1  foreach C_i(τ1) ∈ listUnknownChangeT1 do
2      listOverlapClusterT2 ← ∅
3      foreach C_j(τ2) ∈ listUnknownChangeT2 do
4          overlapCount ← 0
5          for mapUnit = 1 → |M(τ2)| do
6              if CT21[mapUnit] = i ∧
               CT2[mapUnit] = j then
7                  overlapCount +=
                   weightT2[mapUnit]
8          ratio ← overlapCount /
           totalClusterMemberT2[i]
9          if ratio ≥ θ_splitting then
10             listOverlapClusterT2 ←
               listOverlapClusterT2 ∪ {C_i(τ1)}
11     if |listOverlapClusterT2| ≥ 2 then
12         foreach C_j(τ2) ∈ listOverlapClusterT2
           do
13             LC ← LC
               ∪{(C_i(τ1), C_j(τ2), splitting)}
14             listUnknownChangeT2 ←
               listUnknownChangeT2 − {C_j(τ2)}
15         listUnknownChangeT1 ←
           listUnknownChangeT1 − {C_i(τ1)}
```

Algorithm 7: Detecting overlapping clusters.

```
1  foreach C_i(τ1) ∈ listUnknownChangeT1 do
2      foreach C_j(τ2) ∈ listUnknownChangeT2 do
3          overlapCount ← 0
4          for mapUnit = 1 → |M(τ2)| do
5              if CT21[mapUnit] = i ∧
               CT2[mapUnit] = j then
6                  overlapCount +=
                   weightT2[mapUnit]
7          ratio ← overlapCount /
           totalClusterMemberT2[j]
8          if ratio ≥ θ_overlapping then
9              if C_i(τ1) ∈ contractingClusterT1
               then
10                 LC ←
                   LC∪{C_i(τ1), C_j(τ2), contracting}
11             else if
               C_j(τ2) ∈ enlargingClusterT2 then
12                 LC ←
                   LC ∪ {C_i(τ1), C_j(τ2), enlarging}
13             else
14                 LC ← LC ∪
                   {C_i(τ1), C_j(τ2), overlapping}
15             listUnknownChangeT1 ←
               listUnknownChangeT1 − {C_i(τ1)}
16             listUnknownChangeT2 ←
               listUnknownChangeT2 − {C_j(τ2)}
17 foreach C_i(τ1) ∈ listUnknownChangeT1 do
18     LC ← LC ∪ {C_i(τ1), −, −}
19 foreach C_j(τ2) ∈ listUnknownChangeT2 do
20     LC ← LC ∪ {−, C_j(τ2), −}
```

detect the emerging cluster up to $\theta_{emerging} = 0.9$. The higher threshold value means that the weighted version is more sensitive to detect structural cluster changes. Therefore, the subsequent experiments use the weighted version with the threshold parameter 0.5.

Table 1: Threshold values used in the subsequent experiments.

Threshold name	*Threshold value*
Emerging threshold	0.5
Disappearing threshold	0.5
Merging threshold	0.5
Splitting threshold	0.5
Overlapping threshold	0.6

The experiment on the datasets where a cluster disappears in dataset $\mathcal{D}(\tau_2)$ shows that the weighted calculation can detect the changes up to $\theta_{disappearing} = 0.6$, while the unweighted version can detect up to $\theta_{disappearing} = 0.5$. On detecting merging clusters, the weighted calculation can detect the changes up to $\theta_{merging} = 0.8$, while the unweighted version can detect up to $\theta_{merging} = 0.7$. The experiment on the datasets where a cluster splits into two clusters in dataset $\mathcal{D}(\tau_2)$ shows that the weighted calculation can detect the changes up to $\theta_{splitting} = 1.0$, while the unweighted version can detect up to $\theta_{splitting} = 0.9$. Lastly, when the algorithm is evaluated on the datasets where a cluster $C_i(\tau_1)$ overlaps with a cluster $C_j(\tau_2)$, the weighted calculation can detect the changes up to $\theta_{overlapping} = 0.6$, while the unweighted version can detect up to $\theta_{splitting} = 0.7$. Therefore, the subsequent experiments use the threshold parameter shown in Table 1.

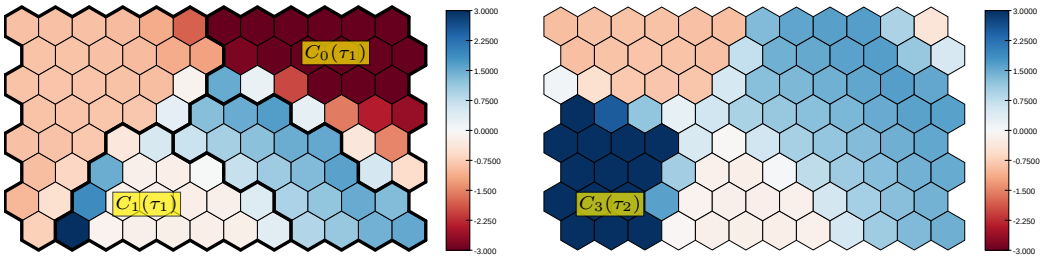
6 Evaluations on Synthetic and Real-Life Datasets

To evaluate the algorithm and the threshold values, this research uses two pairs of synthetic datasets with multiple structural changes and two pairs of real-life datasets. The first synthetic datasets 'lost-new' contains an emerging cluster and a disappearing cluster as shown in Figure 3. The second synthetic datasets contains a splitting cluster, merging clusters, expanding clusters, and contracting clusters.

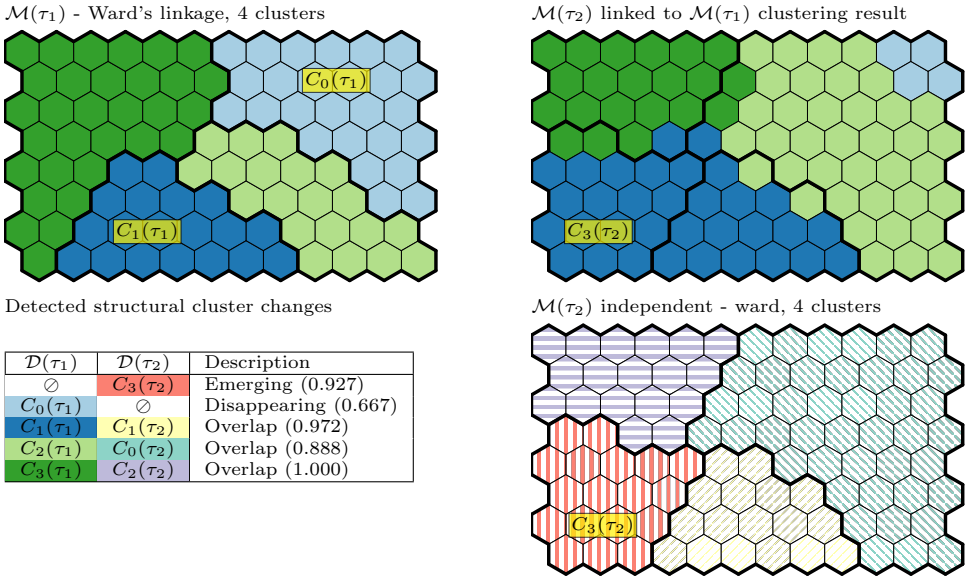
Analysis using ReDSOM and cluster colour linkage on the 'lost-new' dataset shows that there are a disappearing cluster and an emerging cluster. The scatter plot of both dataset $\mathcal{D}(\tau_1)$ (*red dots*) and $\mathcal{D}(\tau_2)$ (*blue pluses*) shown in Figure 3 indicates a lost cluster and an emerging cluster. The ReDSOM visualization shown Figure 4(a) indicates that cluster $C_0(\tau_1)$ is disappearing in the map $\mathcal{M}(\tau_2)$. Furthermore, the ReDSOM visualization of the map $\mathcal{M}(\tau_2)$ (*right*) identifies an emerging cluster $C_3(\tau_2)$. Similarly, based on analysis using cluster colour linkage on Figure 4(b), the cluster $C_0(\tau_1)$ on the map $\mathcal{M}(\tau_1)$ no longer exists in the map $\mathcal{M}(\tau_2)$. An emerging cluster $C_3(\tau_2)$ emerged on the map $\mathcal{M}(\tau_2)$.

Evaluation on the 'lost-new' dataset shows that the proposed algorithm able to identify structural cluster changes correctly. The algorithm proposed in this paper produced the table shown in Figure 4(b) (*bottom left*). The algorithm correctly identifies cluster $C_3(\tau_2)$ as an emerging cluster. Furthermore, the algorithm also identifies cluster $C_0(\tau_1)$ as a disappearing cluster. No significant structural changes detected in the other clusters in $\mathcal{M}(\tau_1)$: $C_1(\tau_1)$, $C_2(\tau_1)$, and $C_3(\tau_1)$.

The World Development Indicator (WDI) dataset (World Bank 2003) is a multi-variate temporal dataset covering 205 countries. The experiments compare the clustering structure based on 25 selected indicators that reflect different aspects of welfare,



(a) The ReDSOM visualization of the $\mathcal{M}(\tau_1)$ (*left*) identifies that cluster $C_0(\tau_1)$ is disappearing in the map $\mathcal{M}(\tau_2)$. The ReDSOM visualization of the map $\mathcal{M}(\tau_2)$ (*right*) identifies an emerging cluster $C_3(\tau_2)$.



$\mathcal{D}(\tau_1)$	$\mathcal{D}(\tau_2)$	Description
⊘	$C_3(\tau_2)$	Emerging (0.927)
$C_0(\tau_1)$	⊘	Disappearing (0.667)
$C_1(\tau_1)$	$C_1(\tau_2)$	Overlap (0.972)
$C_2(\tau_1)$	$C_0(\tau_2)$	Overlap (0.888)
$C_3(\tau_1)$	$C_2(\tau_2)$	Overlap (1.000)

(b) The clustering result of the map $\mathcal{M}(\tau_1)$ is shown using colour on both the map $\mathcal{M}(\tau_1)$ (*top left*) and the map $\mathcal{M}(\tau_2)$ (*top right*). The independent clustering result of the map $\mathcal{M}(\tau_2)$ is shown using thick outlines and patterns on the map $\mathcal{M}(\tau_2)$ (*bottom right*). The cluster $C_0(\tau_1)$ on the map $\mathcal{M}(\tau_1)$ no longer exists in the map $\mathcal{M}(\tau_2)$. An emerging cluster $C_3(\tau_2)$ emerged on the map $\mathcal{M}(\tau_2)$.

Figure 4: ReDSOM, distance matrix, and clustering result visualizations of the synthetic 'lost-new' datasets. Map $\mathcal{M}(\tau_1)$ shown on the *left hand side* and $\mathcal{M}(\tau_2)$ on the *right hand side*.

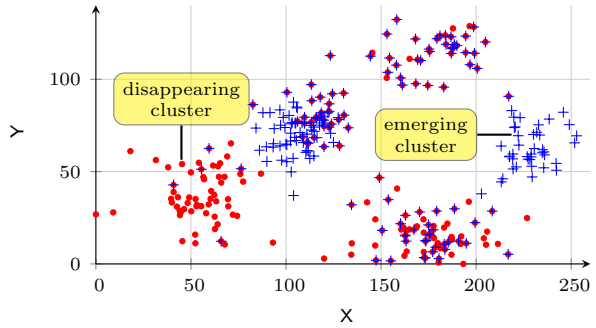


Figure 3: The scatter plot of the synthetic datasets $\mathcal{D}(\tau_1)$ (*red dots*) and $\mathcal{D}(\tau_2)$ (*blue pluses*) that contains emerging cluster and disappearing cluster.

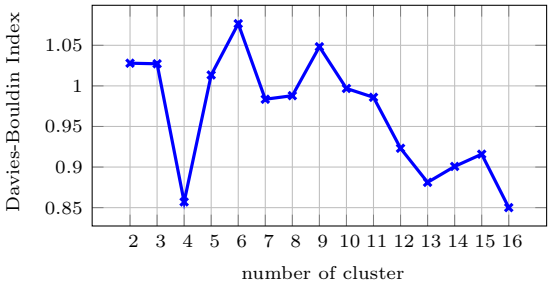
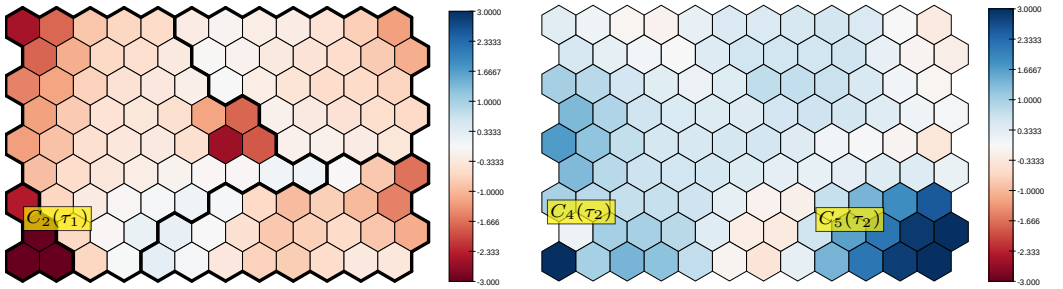


Figure 6: The plot of the Davies-Bouldin Index for k-means clustering result of the 1980s dataset.

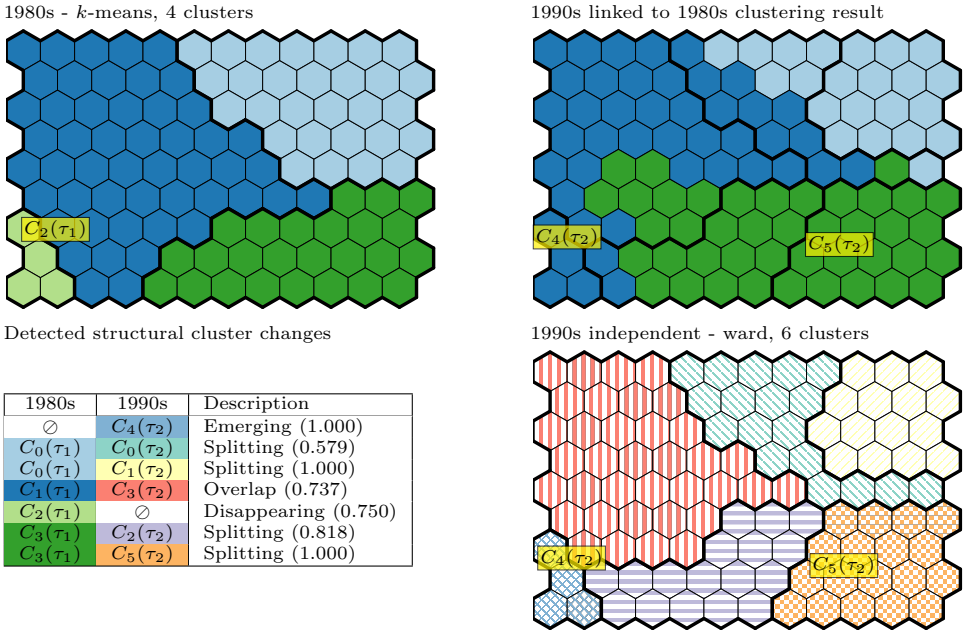
such as *population*, *life expectancy*, *mortality rate*, *immunization*, *illiteracy rate*, *education*, *television ownership*, and *inflation* (Denny & Squire 2005). The annual values are grouped into 10-years value by taking the latest value available in the period. The 1980s data is used as period τ_1 and the 1990s as period τ_2.

The analysis of cluster changes starts by selecting the clustering result of the datasets $\mathcal{D}(\tau_1)$ and $\mathcal{D}(\tau_2)$. Cluster validity indexes, such as Davies-Bouldin Index and Silhouette Index, or dendrogram tree were used to guide this selection. The Davies-Bouldin index of the k-means clustering results (Figure 6) shows that the optimal clustering result is four clusters for the 1980s. Based on the dendrogram of Ward linkage, the optimal clustering result is six clusters for the 1990s.

The table (*bottom left* in Figure 5(b)) shows the detected structural cluster changes. The cells' back-



(a) The ReDSOM visualization of the 1980s map (*left*) identifies cluster $C_2(\tau_1)$ is disappearing in the 1990s map. The ReDSOM visualization of the 1990s map (*right*) identifies a new cluster $C_4(\tau_2)$ and a new region $C_5(\tau_2)$ compared to the 1980s map.



1980s	1990s	Description
⊘	$C_4(\tau_2)$	Emerging (1.000)
$C_0(\tau_1)$	$C_0(\tau_2)$	Splitting (0.579)
$C_0(\tau_1)$	$C_1(\tau_2)$	Splitting (1.000)
$C_1(\tau_1)$	$C_3(\tau_2)$	Overlap (0.737)
$C_2(\tau_1)$	⊘	Disappearing (0.750)
$C_3(\tau_1)$	$C_2(\tau_2)$	Splitting (0.818)
$C_3(\tau_1)$	$C_5(\tau_2)$	Splitting (1.000)

(b) The clustering result of the 1980s map is shown using colour on both the 1980s map (*left*) and the 1990s map (*right*). The independent clustering result of the 1990s map is shown using thick outlines and patterns on the 1990s map (*right*). The cluster $C_2(\tau_1)$ on the 1980s map no longer exists in the 1990s map. A new cluster $C_4(\tau_2)$ emerged on the 1990s map.

Figure 5: ReDSOM, clustering result visualizations, and detected structural cluster changes of the world's welfare and poverty maps of the 1980s (*left*) and the 1990s (*right*).

ground colour refers to the colour of clusters used in the figure. The rd1_j visualization (*left* of Figure 5(a)) shows that cluster $C_2(\tau_1)$ disappears in the 1990s. This change is confirmed by the cluster colour linking *top right* visualization shown in Figure 5(b). This disappearing cluster is detected in the table. This cluster consists of four Latin American countries: Brazil, Argentina, Nicaragua, and Peru who experienced a debt crisis in the 1980s, which is known as the 'lost decade'. However, many Latin American countries undertook rapid reforms in the late 1980s and early 1990s. The algorithm also detected two clusters, $C_0(\tau_1)$ and $C_3(\tau_1)$, in the 1980s that were split in the 1990s. The cluster $C_4(\tau_2)$ emerged in the 1990s, consisting of China and India, which were characterized by high total labour forces.

7 Conclusion and Future Work

We have presented an algorithm to automatically detect structural changes between two clustering results obtained from two snapshot datasets. Based the evaluations, the algorithm can detect various structural changes. As a SOM produces a considerably smaller-sized set of prototype vectors, it allows an efficient use of two-level clustering and this automatic detection. The method presented here can be extended further to detect cluster changes from multiple snapshots.

References

Adomavicius, G. & Bockstedt, J. (2008), 'C-TREND: Temporal cluster graphs for identifying and visualizing trends in multiattribute transactional data', *IEEE Transaction on Knowledge and Data Engineering* **20**(6), 721–735.

Aggarwal, C. C. (2005), 'On change diagnosis in evolving data streams', *IEEE Transactions on Knowledge and Data Engineering* **17**, 587–600.

Antunes, C. M. & Oliveira, A. L. (2001), Temporal data mining: An overview, *in* 'KDD 2001 Workshop on Temporal Data Mining', pp. 1–13.

Bartolini, I., Ciaccia, P., Ntoutsi, I., Patella, M. & Theodoridis, Y. (2009), 'The panda framework for

comparing patterns', *Data and Knowledge Engineering* **68**(2), 244–260.

Boettcher, M. (2011), 'Contrast and change mining', *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **1**(3), 215–230.

Denny & Squire, D. M. (2005), Visualization of cluster changes by comparing self-organizing maps, *in* 'Advances in Knowledge Discovery and Data Mining, PAKDD 2005, Proceedings', Vol. 3518 of *LNCS*, Springer, pp. 410–419.

Denny, Williams, G. J. & Christen, P. (2010), 'Visualizing Temporal Cluster Changes using Relative Density Self-Organizing Maps', *KAIS* **25**(2), 281–302.

Diggle, P. J., Liang, K.-Y. & Zeger, S. L. (1994), *Analysis of Longitudinal Data*, Oxford University Press, New York.

Fukui, K.-I., Saito, K., Kimura, M. & Numao, M. (2008), Sequence-based som: Visualizing transition of dynamic clusters, *in* 'Computer and Information Technology (CIT) 2008. 8th IEEE International Conference on', pp. 47–52.

Ganti, V., Gehrke, J., Ramakrishnan, R. & Loh, W.-Y. (2002), 'A framework for measuring differences in data characteristics', *Journal of Computer and System Sciences* **64**(3), 542–578.

Han, J., Kamber, M. & Pei, J. (2011), *Data Mining: Concepts and Techniques (third edition)*, Morgan Kaufmann.

Held, P. & Kruse, R. (2013), Analysis and visualization of dynamic clusterings, *in* 'HICSS', IEEE, pp. 1385–1393.

Hinneburg, A. & Keim, D. A. (2003), 'A general approach to clustering in large databases with noise', *KAIS* **5**, 387–415.

Jain, A. K., Murty, M. N. & Flynn, P. J. (1999), 'Data clustering: a review', *ACM Computing Survey* **31**(3), 264–323.

Kalnis, P., Mamoulis, N. & Bakiras, S. (2005), On discovering moving clusters in spatio-temporal data., *in* 'SSTD 2005, Proceedings', Vol. 3633 of *LNCS*, Springer, pp. 364–381.

Macqueen, J. B. (1967), Some methods of classification and analysis of multivariate observations, *in* 'Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability', University of California Press, pp. 281–297.

Ntoutsi, I., Spiliopoulou, M. & Theodoridis, Y. (2009), 'Tracing cluster transitions for different cluster types', *Control and Cybernetics* **38**(1), 239–259.

Oliveira, M. & Gama, J. a. (2010), Bipartite graphs for monitoring clusters transitions, *in* 'Advances in Intelligent Data Analysis IX', Vol. 6065 of *LNCS*, Springer Berlin / Heidelberg, pp. 114–124.

Parzen, E. (1962), 'On estimation of a probability density function and mode', *The Annals of Mathematical Statistics* **33**(3), 1065–1076.

Roddick, J. F., Hornsby, K. & Spiliopoulou, M. (2001), An updated bibliography of temporal, spatial, and spatio-temporal data mining research, *in* 'Temporal, Spatial, and Spatio-Temporal Data Mining', Vol. 2007 of *LNCS*, pp. 147–163.

Roddick, J. F. & Spiliopoulou, M. (2002), 'A survey of temporal knowledge discovery paradigms and methods', *IEEE TKDE* **14**(4), 750–767.

Rosenblatt, M. (1956), 'Remarks on some nonparametric estimates of a density function', *Annals of Mathematical Statistics* **27**(3), 832–837.

Silverman, B. (1986), *Density estimation for statistics and data analysis*, Chapman and Hall Ltd, London, New York.

Skupin, A. & Hagelman, R. (2005), 'Visualizing demographic trajectories with Self-Organizing Maps', *GeoInformatica* **9**, 159–179.

Spiliopoulou, M., Ntoutsi, I., Theodoridis, Y. & Schult, R. (2006), Monic: modeling and monitoring cluster transitions, *in* 'KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining', ACM Press, New York, NY, USA, pp. 706–711.

Vesanto, J. & Alhoniemi, E. (2000), 'Clustering of the Self-Organizing Map', *IEEE Transactions on Neural Networks* **11**(3), 586–600.

World Bank (2003), *World Development Indicators 2003*, The World Bank, Washington DC.