Control and Cybernetics

vol. **38** (2009) No. 1

Tracing cluster transitions for different cluster types*†

by

Irene Ntoutsi[1], Myra Spiliopoulou[2] and Yannis Theodoridis[1]

[1]Department of Informatics, University of Piraeus, Greece

[2]Faculty of Computer Science, University of Magdeburg, Germany

e-mail: {ntoutsi, ytheod}@unipi.gr, myra@iti.cs.uni-magdeburg.de

Abstract: Clustering algorithms detect groups of similar population members, like customers, news or genes. In many clustering applications the observed population evolves and changes over time, subject to internal and external factors. Detecting and understanding changes is important for decision support. In this work, we present the MONIC$^+$ framework for cluster-type-specific transition modeling and detection. MONIC$^+$ encompasses a typification of clusters and cluster-type-specific transition indicators, by exploiting cluster topology and cluster statistics for the transition detection process. Our experiments on both synthetic and real datasets demonstrate the usefulness and applicability of our framework.

Keywords: dynamic environments, change detection, cluster transitions, transition indicators, cluster-type-specific indicators.

1. Introduction

For many clustering applications, clusters should not be observed as static objects, since the underlying datasets undergo changes over time, e.g., customers and their buying preferences, scientific publications and their topics or viruses and their resistance to medicaments. Research on spatiotemporal clustering, incremental clustering and stream clustering addresses the problem by *adapting* clusters to changing datasets. However, the *tracing and understanding* of the changes themselves is of no less importance for effective decision support. This latter direction is the subject of this work.

One difficulty with detecting cluster changes results from the fact that there exist a variety of cluster types due to different clustering methods that have been proposed in the literature, e.g., hierarchical, partitioning, density based methods. To overcome this problem, in our previous work (Spiliopoulou et

*A short version of this work is constituted by Spiliopoulou, Ntoutsi and Theodoridis (2007)

†Submitted: June 2008; Accepted: October 2008

al., 2006), we proposed the MONIC framework that models clusters as sets of objects. MONIC is independent of the clustering algorithm since it relies on the contents of the underlying data stream. However, due to its generality, MONIC does not exploit, during the transition detection process, the particular features of the different cluster types. In this work, we extend MONIC into MONIC$^+$ that also covers the special characteristics associated with the different cluster types, thus allowing us to capture cluster-type-specific transitions.

After discussing related work in Section 2, we introduce in Section 3 a typification of clusters. In the same section, we specify the notion of cluster overlap and cluster match so as to detect consecutiveness between clusters discovered at different timepoints over an accumulating data stream. Section 4 contains our cluster transition detection method and heuristics for different cluster types. In Section 5, we present our first experimental results, whereas Section 6 concludes our study.

2. Related work

Research relevant to our work can be categorized into methods for cluster change detection and methods for spatiotemporal clustering. Also relevant to our work are methods on the specific subject of topic evolution.

2.1. Cluster change detection

The FOCUS change detection framework (Ganti, Gehrke and Ramakrishnan, 1999) compares two datasets and computes a deviation measure between them, based on the data mining models they induce. Clusters compose a special case of models: they are modeled as non-overlapping regions described through a set of attributes (structure component) and corresponding to a set of raw data (measure component). However, the emphasis in this work is on comparing datasets, not in understanding how a cluster has evolved inside a new clustering.

The PANDA framework (Bartolini et al., 2004) proposes methods for comparing simple and complex patterns defined over raw data and over other patterns, respectively. In PANDA terminology, a cluster is a simple pattern, whereas a clustering is a complex pattern. PANDA, though, concentrates on the generic and efficient realization of comparisons between patterns, rather than on detection and interpretation of cluster transitions.

The MONIC framework (Spiliopoulou et al., 2006) categorizes and traces changes upon accumulating datasets by studying changes in the clusterings derived from these datasets. MONIC treats clusters as sets of objects, thus it is independent of the clustering algorithm. Due to its generality, however, MONIC does not exploit the specific characteristics of each cluster type, e.g., topology. In this work, we extend MONIC by exploiting the special features of each cluster type and thus, we derive appropriate transition indicators for specific cluster types.

Meila (2002) provides an overview of the related work on comparing different clustering results produced from the same dataset under different mining parameters, e.g., different algorithms or different parameters. The comparison process relies on criteria based on i) counting pairs, ii) cluster matching, and iii) variation of information. The counting pairs criteria are based on counting the pairs of points on which the two clusterings agree (i.e., place them in the same cluster). The cluster matching criteria are based on finding for each cluster of the first clustering its best match in the second clustering, where the best match is evaluated based on the number of common points between the two clusters. Recently, Zhou, Li and Zha (2005) proposed a new measure in this category that also considers the distance between cluster centroids, in order to yield more intuitive results. The variation of information criteria measure the amount of information that is lost or gained when moving from one clustering to the other. Note, however, that all these methods refer to the comparison of (different) clusterings extracted from the same dataset and thus cannot be applied in the general case of different datasets (Ntoutsi, Pelekis and Theodoridis, 2007).

2.2. Spatiotemporal clustering

Aggarwal (2005) models clusters through kernel functions and their changes as kernel density changes at each spatial location of the trajectory. The emphasis is on computing change velocity and finding the locations with the highest velocity–the epicenters. Three different types of change are considered in this work: i) data coagulation that corresponds to connected regions in the data, which have velocity density larger than a user defined threshold, ii) data dissolution that correspond to connected regions, whose velocity density is smaller than a user defined threshold and iii) data shift to other locations. Yang, Parthasarathy and Mehta (2005) detect formation and dissipation events for clusters of spatial scientific data. Their framework supports four types of spatial object association patterns (SOAP), namely Star, Clique, Sequence, and minLink, which are used to model different interactions among spatial objects. Such methods, however, assume that the feature space is static, i.e., it does not evolve over time. Thus, they cannot be used for dynamic feature spaces, e.g., in text stream mining, where features are usually frequent words. Furthermore, hierarchical clustering algorithms cannot be coupled with such a method.

Kalnis, Mamoulis and Bakiras (2005) propose a special type of cluster change, the moving cluster, whose contents may change while its density function remains the same during its lifetime. They find moving clusters by tracing common data records between clusters of consecutive timepoints. Our MONIC$^+$ framework for transition detection is more general, since it encompasses several cluster transition types, allows for the ageing of old objects and does not require the density function of a moving cluster to be invariant.

2.3. Topic evolution

Cluster change detection is also relevant to topic evolution in text streams, where a topic is a cluster label, consisting of the dominant words inside the cluster.

Morinaga and Yamanishi (2004) propose a topic analysis system that fulfills three related tasks, namely i) topic structure identification of what kind of main topics exist and how important they are, ii) topic emergence detection of the emergence of a new topic and iii) topic characterization, identifying the characteristics of each main topic. All tasks are formalized in terms of a finite mixture model.

Mei and Zhai (2005) propose a method for discovering and summarizing the evolutionary patterns of themes in a text stream. Authors detect the themes at each period and use the KL divergence measure to find coherent themes over time, i.e., themes with similar labels. In this way a topic evolution graph is built that can be used to trace theme transitions and to analyze the theme life cycles.

These methods are applicable whenever a human–understandable cluster label can be extracted and traced. Cluster labeling is not feasible for all applications, though. For this reason, the proposed framework MONIC$^+$ detects cluster transitions rather than cluster label transitions.

3. Clustering over dynamic data

We assume that data are clustered at timepoints $t_1, \ldots, t_n$. Clustering ζ_i, derived at timepoint t_i, corresponds to a partitioning of the dataset D_i seen thus far. As is typical in data streams, we allow for the decay of old records: We use an *ageing function* $age(x, t_i) \in [0, 1]$ that assigns a weight to each record x seen at t_i or earlier, so that the most recent records are assigned higher weights.

DEFINITION 1 (DATA AGEING FUNCTION) *Let $t_1 \ldots, t_n$ be the sequence of timepoints under observation and let $d_i, i = 2 \ldots, n$ be the set of data records accumulated from t_{i-1} until t_i, while d_1 is the initial dataset, so that $d_i \cap d_j = \emptyset$ for $i \neq j$. A* data ageing function *assigns a weight $age(x, t_i) \in [0, 1]$ to data record x at t_i, for each $x \in \cup_{l=1}^{i} d_l$ and for each t_i:*

$$age : \cup_{l=1}^{i} d_l \times \{t_1 \ldots, t_n\} \rightarrow [0, 1].$$

The weights assigned by the ageing function determine the impact of each record upon clustering ζ_i derived over the dataset $D_i \equiv \cup_{l=1}^{i}(d_l, age, t_i)$. The simplest form of this function is a sliding window.

Our goal is to trace/monitor a cluster found at some timepoint t_i among the clusters of the next timepoint t_j. Since this depends on the notion of cluster itself, we first introduce a typification of clusters (Section 3.1), which we use next to derive type-specific concepts for the comparison of clusters along the time axis (Section 3.2).

3.1. Typification of cluster definitions

Clustering algorithms use a variety of cluster definitions (Han and Kamber, 2000). We propose the following typification that facilitates the study of clusters as *changing objects*:

- **Type A clusters:** Clusters are discovered within a *dataset-independent* metric space. A cluster is a geometric object, e.g. a sphere like in K-means. Cluster changes are observed over the static metric space as geometric transformations.
- **Type B1 clusters:** There is no metric space or it depends on the contents of the dataset at each timepoint. A cluster is defined *extensionally* as a set of data records. Hierarchical algorithms, which build dendrograms and express clusters as sets of proximal data points belong to this type. These algorithms use a metric space to derive a clustering on a dataset, but this space is *data-dependent*, in the sense that the addition of a new record might change the border of a cluster, even if this record does not belong to the cluster.
- **Type B2 clusters:** A cluster is defined *intensionally* as a distribution. For a cluster X of type B2, we denote its cardinality as $card(X)$, its mean as $\mu(X)$ and its standard deviation as $\sigma(X)$. The Expectation-Maximization (EM) algorithm belongs to this category.

Several combinations of the basic types are possible, e.g., when both the dataset and its statistics are used (types B1+B2).

Note that each cluster can be described as a set of objects (i.e., type B1); this is a generic definition that holds for every cluster type. Indeed, this was the approach followed in MONIC (Spiliopoulou et al., 2006) so as to provide a cluster-type-independent framework for transition detection.

3.2. Cluster matching

A *cluster transition* is a change effected upon a cluster $X \in \zeta_i$, discovered at t_i, when we observe it at the next timepoint t_j. The first step in detecting a transition is the tracing of X in the clustering ζ_j of t_j – if it still exists. We define the notion of *overlap* and of (best) *match* for a cluster, before we proceed with a categorization of cluster transitions.

Definition 2 (Cluster overlap) *Let ζ_i be the clustering discovered at timepoint t_i and ζ_j the one discovered at $t_j, j \neq i$. We define a function overlap() that computes the similarity or overlap of a cluster $X \in \zeta_i$ towards a cluster $Y \in \zeta_j$ as a value in $[0, 1]$ such that:*
(i) the value 1 indicates maximum overlap, while 0 stands for no overlap and
(ii) $\sum_{Y \in \zeta_j} overlap(X, Y) \leq 1$.

Cluster overlap is defined asymmetrically. After this generic definition of the overlap function, we specify *overlap*() for each cluster type.

DEFINITION 3 (OVERLAP FOR TYPE A CLUSTERS) *Let ζ_i, ζ_j be two clusterings of Type A clusters, derived at $t_i < t_j$, respectively. For two clusters $X \in \zeta_i$ and $Y \in \zeta_j$, the overlap of X to Y is the normalized intersection of their areas:*

$$overlap(X,Y) = \frac{area(X) \cap area(Y)}{area(X)}.$$

DEFINITION 4 (OVERLAP FOR TYPE B1 CLUSTERS) *Let ζ_i, ζ_j be two clusterings of Type B1 clusters, derived at $t_i < t_j$, respectively. For two clusters $X \in \zeta_i$ and $Y \in \zeta_j$, the overlap of X to Y equals to the normalized sum of the weights of their common data points:*

$$overlap(X,Y) = \frac{\sum_{a \in X \cap Y} age(a,t_j)}{\sum_{x \in X} age(x,t_j)}.$$

DEFINITION 5 (OVERLAP FOR TYPE B2 CLUSTERS) *Let ζ_i, ζ_j be two clusterings of Type B2 clusters, derived at $t_i < t_j$, respectively. For two clusters $X \in \zeta_i$ and $Y \in \zeta_j$, the overlap of X to Y is defined in terms of the proximity of their means:*

$$overlap(X,Y) = \begin{cases} 1 - \frac{|\mu(X) - \mu(Y)|}{\sigma(X)} & , \quad |\mu(X) - \mu(Y)| \leq \sigma(X) \\ 0 & , \quad otherwise \end{cases}.$$

We can now define for each cluster found at a timepoint t_i, its best match at a later timepoint t_j.

DEFINITION 6 (CLUSTER MATCH) *Let $X \in \zeta_i$, $Y \in \zeta_j$ be two clusters derived at $t_i < t_j$, respectively. Further, let $\tau \equiv \tau_{match} \in (0.5, 1]$ be a threshold value. Y is "a match for X in ζ_j subject to τ", i.e., $Y = match_\tau(X, \zeta_j)$, iff:*
(i) Y has the maximum overlap to X among all the clusters in ζ_j, i.e.
$overlap(X,Y) = \max_{Y' \in \zeta_j}\{overlap(X,Y')\}$ and
(ii) $overlap(X,Y) \geq \tau$.
If no such cluster exists for X in ζ_j, then $match_\tau(X, \zeta_j) = \emptyset$.

4. Detecting cluster transitions

A cluster transition is a change experienced by a cluster that was discovered at the previous timepoint. We first provide a typification of cluster transitions (Section 4.1). Then, in Section 4.2, we describe the generic transition detection process, and how it is customized for specific cluster types (Section 4.3).

4.1. Cluster transitions

We use the transition model of MONIC (Spiliopoulou et al., 2006). According to this model, a transition might concern the content and the form of the cluster

(*internal transition*) or rather its relationship to the whole clustering (*external transition*).

The *external transitions* of cluster $X \in \zeta_i$ with respect to clustering ζ_j discovered at the next timepoint t_j are as follows:

- **Survival:** X survives as cluster $Y \in \zeta_j$ (notation: $X \rightarrow Y$), if (a) there is a match Y for it in ζ_j and (b) this match does not contain any further cluster of ζ_i.
- **Absorption:** X is absorbed by cluster $Y \in \zeta_j$ (notation: $X \xrightarrow{\subset} Y$), if the match Y of X is also a match for some other cluster X' of ζ_i.
- **Split:** X is split into clusters $Y_1, \ldots, Y_p \in \zeta_j$ (notation: $X \xrightarrow{\subset} \{Y_1, \ldots, Y_p\}$), if each of these clusters overlaps with X at at least τ_{split} and, when taken together, they form a match for X. We show later how the "taking all clusters together" is realized for each cluster type.
- **Dissapearance:** X has disappeared (notation: $X \rightarrow \odot$), if none of the above cases holds.

The external transitions refer to existing clusters. *Emerging* clusters in ζ_j can be easily detected as those that are not the result of some external transition (notation: $\odot \rightarrow Y$).

If a cluster survives, *internal transitions* may occur. We categorize internal transitions into changes in size, compactness and location:

- **Size transition:** A cluster X might: (a) *shrink* into a smaller cluster, $X \searrow Y$ or (b) *expand* into a larger cluster, $X \nearrow Y$.
- **Compactness transition:** A cluster X might become: (a) *more compact*, $X \xrightarrow{\bullet} Y$ or (b) *less compact/ more diffuse*, $X \xrightarrow{\star} Y$.
- **Location transition:** A cluster X might shift, $X \cdots \rightarrow Y$.
- **No change:** $X \leftrightarrow Y$.

Transitions in a group are mutually exclusive, but transitions of different groups can be combined. For example, a cluster $X \in \zeta_i$, matched by $Y \in \zeta_j$, can become larger and more compact, while its location in a metric space might shift, i.e., $X \cdots \xrightarrow{\bullet} \nearrow Y$.

4.2. Detection of external transitions

The abstract process of external transition detection in MONIC$^+$ is similar to the one of MONIC (Spiliopoulou et al., 2006), but the implementation of some tasks is different for each cluster type. In Fig. 1, we briefly describe the transition detection algorithm so as to familiarise the reader with its concepts. Our emphasis here is on the effects of the different cluster types on the transition detection process, thus, we suggest the interested user to consult MONIC for an extensive description of the algorithm.

The algorithm takes as input the clustering ζ_i discovered at t_i and detects external transitions in ζ_j of $t_j > t_i$. For each cluster $X \in \zeta_i$, it computes its overlap with each cluster in ζ_j (line 4): To speed up this step, the contingency

DetectExternalTransitions()

```
Input: ζ_i, ζ_j , Output: the external transitions from ζ_i to ζ_j
1.  FOR X ∈ ζ_i
2.    splitCandidates = splitUnion = ∅; survivalCluster = NULL;
3.    FOR Y ∈ ζ_j
4.      Mcell = overlap(X,Y);
5.      IF Mcell ≥ τ_match THEN
6.        IF g(X,Y)>g(X,survivalCluster) THEN survivalCluster=Y;
7.        ENDIF
8.      ELSEIF Mcell ≥ τ_split THEN
9.       splitCandidates += Y; splitUnion = splitUnion ∪Y;
10.     ENDIF
11.   ENDFOR
12.   IF (survivalCluster ==NULL OR splitCandidates== ∅) THEN
13.     deadList += X;                                    // X → ⊙
14.   ELSEIF splitCandidates ≠ ∅ THEN
15.     IF overlap(X,splitUnion) ≥ τ_match THEN
16.        FOR Y ∈ splitCandidates
17.          splitList += (X,Y);
18.        ENDFOR                                  //X ⊆→ splitCandidates
19.     ELSE deadList += X;                                //X → ⊙
20.     ENDIF
21.   ELSE absorbSurvivals+=(X,survivalCluster);
22.   ENDIF
23. ENDFOR
24. FOR Y ∈ ζ_j
25.   absorbCandidates=makeList(absorbSurvivals,Y);
26.   IF cardinality(absorbCandidates)>1 THEN            //X ⊆→ Y
27.     FOR X ∈ absorbCandidates
28.       absorbList += (X,Y); absorbSurvivals -= (X,Y);
29.     ENDFOR
30.   ELSEIF absorbCandidates=={X} THEN                  //X → Y
31.     survivalList += (X,Y); absorbSurvivals -= (X,Y);
32.   ENDIF
33. ENDFOR
```

Figure 1. Detector of external transitions

matrix $\mathcal{M}$ of the overlap values is built in advance (according to the overlap definitions in Section 3.2) and each cell $\mathcal{M}$ (`Mcell`) is retrieved when required. The detector looks first for clusters in ζ_j that match X (lines 5–6) finding the best survival candidate (if any) according to Definition 6. If no survival candidate exists, clusters overlapping with X for more than τ_{split} are found (lines 8–9). If neither some survival candidate nor some split candidates exist, then X is marked as having disappeared (lines 12–13).

For cluster split detection, we build a list of candidates (line 9). As specified in the transition model, these clusters must form *together* a match for X. *Taking the clusters together* is a cluster-type-specific operation: For B1-clusters it corresponds to a set union, for A-clusters to the computation of a common area. Split detection is not possible for B2-clusters, as there is no notion of "taking distributions together" in this case; we elaborate further on "taking the clusters together" in Section 4.3. For A- and B1-clusters, if the overlap test (line 15) succeeds, X is marked as split (line 17), otherwise it is marked as disappeared (19).

To trace absorption and survival, ζ_i clusters and their survival candidates are added to a list of absorptions and survivals (line 21). When all ζ_i clusters are processed, this list is completed (line 23). Then, for each ζ_j cluster Y, the detector extracts from this list all ζ_i clusters, for which Y is a survival candidate (line 25). If this sublist contains only one cluster, then there is a survival of X within this cluster (lines 30–32). If it contains more than one cluster, then these have been absorbed by Y: they are marked as such and removed from the original list (lines 26–29). Similarly to cluster split detection, cluster absorption can be traced for some cluster types only; lines 14–22 do not apply for type B2 clusters.

In the next section we describe the observed transitions for each cluster type and we propose type-specific transition indicators.

4.3. Type-dependent detection of transitions

The observable transitions for each cluster type are depicted in Table 1. All external and internal transitions can be detected for clusters in a metric space (Type A). For clusters defined extensionally (Type B1), compactness and location transitions cannot be observed directly, because concepts like proximity and movement are not defined. However, when one derives the intensional definition of a cluster, both transitions become observable as changes in the cluster density function; we refer to this as Type B1+B2. Conversely, the intensional definition of a cluster (Type B2) does not allow for the detection of splits and absorptions, which in turn can be found by studying the cluster members (Type B1+B2).

Table 1. Observable transitions for each cluster type

	External	***Internal transitions***		
Cluster type		*Size*	*Compact.*	*Location*
A. metric space	Yes	Yes	Yes	Yes
B1. extensional	Yes	Yes	No	No
B2. intensional	Survival	Yes	Yes	Yes
B1+B2.	Yes	Yes	Yes	Yes

4.3.1. Transition indicators for Type A clusters

Let ζ_i, ζ_j be the clusterings at timepoints $t_i < t_j$ and let $X \in \zeta_i$ be the cluster under observation. The transition indicators proposed in Table 2 use the type-specific definition of cluster overlap (see Definition 3) and the derived definition of cluster match (see Definition 6).

Table 2. Indicators for Type A cluster transitions

Step	***Transition***	***Indicator***
1	Survival or Absorption	$\exists Y \in \zeta_j : \frac{area(X) \cap area(Y)}{area(X)} \geq \tau$
2	$X \overset{\subseteq}{\rightarrow} Y$	$\exists Z \in \zeta_i \setminus \{X\} : \frac{area(Z) \cap area(Y)}{area(Z)} \geq \tau$
3	$X \rightarrow Y$	$\nexists Z \in \zeta_i \setminus \{X\} : \frac{area(Z) \cap area(Y)}{area(Z)} \geq \tau$
4	Split	$\exists Y_1, \ldots, Y_p \in \zeta_1 :$ $(\forall Y_u : \frac{area(X) \cap area(Y_u)}{area(X)} \geq \tau_{split}) \wedge$ $\frac{area(X) \cap area(\cup_{u=1}^{p} Y_u)}{area(X)} \geq \tau$
5	$X \rightarrow \odot$	derived from the above
Internal transitions, for surviving clusters: $X \rightarrow Y$		
6	Size	B1 indicators & B2 indicators
7	Compactness	geometry-dependent & B2 indicators
8	Location	geometry-dependent & B2 indicators

External cluster transitions are detected by computing the area overlap between cluster X and each candidate in ζ_j. To detect a *split*, we customize the split test (line 15) of Algorithm 1. More specifically, we compute the overlap between the area of X and that of all split candidates. Since these candidates cannot overlap, we use the following equation to perform the split test:

$$area(X) \cap area(\cup_{u=1}^{p} Y_u) = \sum_{u=1}^{p} area(X) \cap area(Y_u).$$

The detection of internal transitions for type A clusters translates into tracing the movements of a cluster in a static metric space. In Table 3, we propose

indicators for spherical clusters, as those produced by, e.g., K-Means and K-Medoids. We can further use indicators for Type B1 and B2 clusters (discussed next).

Table 3. Indicators for spherical clusters (Type A)

Transition	***Indicator***
$X \nearrow Y$	$\frac{area(X)}{area(Y)} \leq \tau_{size}$
$X \searrow Y$	$\frac{area(X)}{area(Y)} \geq \tau_{size}$
$X \cdots \rightarrow Y$	$\frac{d(center(X),center(Y))}{min\{radius(X),radius(Y)\}} \geq \tau_{location}$
$X \xrightarrow{\bullet} Y$	$avg_{x \in X}(d(x, center(X)) > avg_{y \in Y}(d(y, center(Y)) + \tau_{compactness}$
$X \xrightarrow{\star} Y$	$avg_{y \in Y}(d(y, center(Y)) > avg_{x \in X}(d(x, center(X)) + \tau_{compactness}$

The first two heuristics in Table 3 detect *size* transitions by comparing the area of X with the area of its surviving counterpart, Y, subject to a threshold value τ_{size}. If the area grows with respect to this value, this is a sign of size expansion, whereas if this area decreases, this is a sign of size shrink. The third heuristic in Table 3 detects *location* transitions by checking whether the distance between the centers exceeds a threshold $\tau_{location}$; we normalize this distance against the size of the smallest radius. The fourth heuristic states that a cluster has become *more compact* if the average distance from the center was larger in the old cluster than in the new one – subject to a small threshold value $\tau_{compactness}$. The fifth heuristic for clusters becoming *less compact* is the reverse of the second one.

4.3.2. Transition indicators for Type B1 clusters

Let $X \in \zeta_i$ be a cluster found in t_i. To trace its transitions in ζ_j, we consider the indicators proposed in Table 4 for the transitions that can be observed over Type B1 clusters (compare Table 1).

External cluster transitions are detected by computing the common members between cluster X and each candidate in ζ_j. The split test (line 15) of Algorithm 1 can be performed more effectively, if one observes that two clusters in ζ_j cannot have common members. Then, the split test can be computed from the individual intersection values in $\mathcal{M}$, because:

$$\sum_{a \in X \cap (\cup_{u=1}^{p} Y_u)} age(a, t_j) = \sum_{u=1}^{p} \sum_{a \in X \cap Y_u} age(a, t_j).$$

Regarding the internal transitions, only the size transition is observable for clusters of type B1. *Size transitions* for a cluster X that has survived as Y

Table 4. Indicators for Type B1 cluster transitions

Step	***Transition***	***Indicator***
1	Survival or Absorption	$\exists Y \in \zeta_j : \frac{\sum_{a \in X \cap Y} age(a,t_j)}{\sum_{x \in X} age(x,t_j)} \geq \tau$
2	$X \xrightarrow{\subseteq} Y$	$\exists Z \in \zeta_i \setminus \{X\} : \frac{\sum_{a \in Z \cap Y} age(a,t_j)}{\sum_{z \in Z} age(z,t_j)} \geq \tau$
3	$X \rightarrow Y$	$\nexists Z \in \zeta_i \setminus \{X\} : \frac{\sum_{a \in Z \cap Y} age(a,t_j)}{\sum_{z \in Z} age(z,t_j)} \geq \tau$
4	Split	$\exists Y_1, \ldots, Y_p \in \zeta_j :$ $(\forall Y_u : \frac{\sum_{a \in X \cap Y_u} age(a,t_j)}{\sum_{x \in X} age(x,t_j)} \geq \tau_{split}) \wedge$ $\frac{\sum_{a \in X \cap (\cup_{u=1}^{p} Y_u)} age(a,t_j)}{\sum_{x \in X} age(x,t_j)} \geq \tau$
5	$X \rightarrow \odot$	derived from the above
Internal transitions, for surviving clusters: $X \rightarrow Y$		
6	Size	
	$X \nearrow Y$	$\sum_{y \in Y} age(y,t_j) > \sum_{x \in X} age(x,t_i) + \tau_{size}$
	$X \searrow Y$	$\sum_{x \in X} age(x,t_i) > \sum_{y \in Y} age(y,t_j) + \tau_{size}$

are traced by comparing the datasets. While the weights used when computing cluster overlap are those valid at timepoint t_j, the size transition heuristics consider the weights of the members of X at the original t_i. The size transition heuristic should reflect the importance of the individual cluster members at t_i.

4.3.3. Transition indicators for Type B2 clusters

We consider again a cluster $X \in \zeta_i$. To detect size transitions, we used the heuristic for Type B1 clusters (see Table 4). For the other observable transitions (see Table 1), we use the indicators in Table 5.

Table 5. Indicators for Type B2 cluster transitions

Step	***Transition***	***Indicator***
1	$X \rightarrow Y$	$\exists Y \in \zeta_j : 1 - \frac{\lvert\mu(X) - \mu(Y)\rvert}{\sigma(X)} \geq \tau$
2	$X \rightarrow \odot$	negation of the above
Internal transitions, for survived clusters: $X \rightarrow Y$		
3	Size	B1 indicators in Table 4
4	Location	h1. $\lvert\mu(X) - \mu(Y)\rvert > \tau_{h1}$ h2. $\lvert\gamma(X) - \gamma(Y)\rvert > \tau_{h2}$
5	Compactness	
	$X \xrightarrow{\bullet} Y$	$\sigma(Y) < \sigma(X) + \tau_{compactness}$
	$X \xrightarrow{\star} Y$	$\sigma(X) < \sigma(Y) + \tau_{compactness}$

The first indicator in Table 5 states that a cluster *survives* if there is a match for it, subject to a $\tau \in (0.5, 1]$ (see Definition 6): the indicator demands that $\mu(X)$ and $\mu(Y)$ be closer than half a standard deviation. Since clusters of the same clustering do not overlap, we expect that no more than one cluster of ζ_j satisfies this condition.

An *absorption* transition for $X \in \zeta_i$ implies finding a $Y \in \zeta_j$ that contains $X, Z \in \zeta_i$. Similarly, a *split* transition corresponds to finding clusters that contain subsets of X. However, this implies treating the clusters as datasets (Type B1). So, we only consider survival and disappearance for B2-clusters.

To detect *compactness* transitions, we use the difference of the standard deviations of the clusters X, Y. For *location* transitions, we use two heuristics that reflect different types of cluster shift: h1 detects shifts of the mean $\mu()$ (within half a standard deviation, see Definition 6), while h2 traces changes in the skewness $\gamma()$:

$$\gamma(X) = \frac{\frac{1}{card(X)} \sum_{x \in X} (x - \mu(X))^3}{\left(\frac{1}{card(X)} \sum_{x \in X} (x - \mu(X))^2 \right)^{\frac{3}{2}}}.$$

Heuristic h2 becomes interesting for clusters, where the mean has not changed, but the distribution exhibits a longer or shorter tail.

5. Experiments

We have experimented with a synthetic stream of data records, in which we have imputed cluster transitions, and with a real dataset, the Network Intrusion dataset (KDD Cup'99), which is considered to evolve rapidly over time (see Aggarwal et al., 2003). The goal of the experiments was to gain insights on the evolution of each dataset by studying the transitions between the underlying clusters, extracted at consecutive timepoints. Below, we describe the findings of our framework for each dataset; a short description of the datasets is first provided.

5.1. Experimenting with a synthetic dataset

The goal of the experiments with the synthetic dataset was to demonstrate the potentiality of our framework and to display the kind of transitions that it can detect. To this end we used a 2D dataset, which can be easily visualized.

5.1.1. Generation of an accumulating dataset

We used a *data generator* that takes as input the number of data points M, the number of clusters K, as well as the mean and standard deviation of the anticipated members of each cluster. The records were generated around the mean and subject to the standard deviation, following a Gaussian distribution.

We fixed the standard deviation to 5 and used a 100×100 workspace for two-dimensional datapoints. The stream was built according to the scenario below.

- t_1**:** Dataset d_1 consists of points around the $K_1 = 5$ centers (20,20), (20, 80), (80, 20), (80, 80), (50, 50).
- t_2**:** Dataset d_2 consists of 40 datapoints, distributed equally across the four corner-groups of d_1 data.
- t_3**:** d_3 consists of 30 points around location (50,40) and 30 points around (50,60).
- t_4 and subsequent**:** At each of t_4, t_5, t_6, we added 30 points around t_4 :(20,50), t_5 :(20,30) and t_6 :(20,40).

For data ageing, we used a sliding window of size $w = 2$. Hence, at each timepoint $t_i, i > 1$, the dataset under observation was $D_i = d_i \cup d_{i-1}$.

5.1.2. Clustering and transition detection

We have built *Type A clusters* with K-Means (Witten and Frank, 2005). For *Type B1 clusters*, we have used Expectation-Maximization (EM) (Witten and Frank, 2005), which models clusters as Gaussian distributions; we ignored the distribution information, though, and treated the clusters as datasets (type B1). For K-means, we have defined K to be the optimal number of clusters found by EM. The clusterings found at $t_1, \ldots, t_6$ with EM are shown in Fig. 2. Those found with K-Means are in Fig. 3; they are different from the EM clusters, thus implying also different cluster transitions.

Results for type B1 clusters: Fig. 2 depicts the clusters at each timepoint but delivers little information about the impact of new data and of data ageing. Such information can be derived by studying the cluster transitions across the time axis; to this end, we have used the transition indicators of Table 4, setting $\tau = 0.5$ and $\tau_{split} = 0.2$ and $\tau_{size} = 0.2$.

In Table 6, the changes in the population are reflected in the discovered transitions. MONIC$^+$ has correctly mapped the old clusters to the new ones, identifying size transitions, survivals, absorptions and splits. There are also new clusters found at t_4 and t_5.

Results for type A clusters: For Type A clusters, we have used the indicators in Table 2, setting $\tau = 0.5$ and $\tau_{split} = 0.2$. For the size transition, we have used the B1 indicator in Table 4 with $\varepsilon = 0.003$. For the other internal transitions, we have used the indicators for spheres in Table 3 with $\tau_{location} = 0.1$ (location transitions) and $\varepsilon = 0.001$ (compactness transitions). The transitions found by MONIC$^+$ and shown in Table 7 reveal that most clusters are unstable, experiencing all types of internal transitions, or they disappear, giving place to new (unstable) clusters.

Even in the absence of a visualization (which might be difficult for a real dataset in a multi-dimensional feature space), these transitions indicate the

cluster instability and the need for closer inspection of the individual clusters.

Comparing the transitions of clusters extracted through K-means (Fig. 3) and EM (Fig. 2), one can observe that K-means clusters experience more transitions, which is justified by the fact that K-means results in less stable clusters.

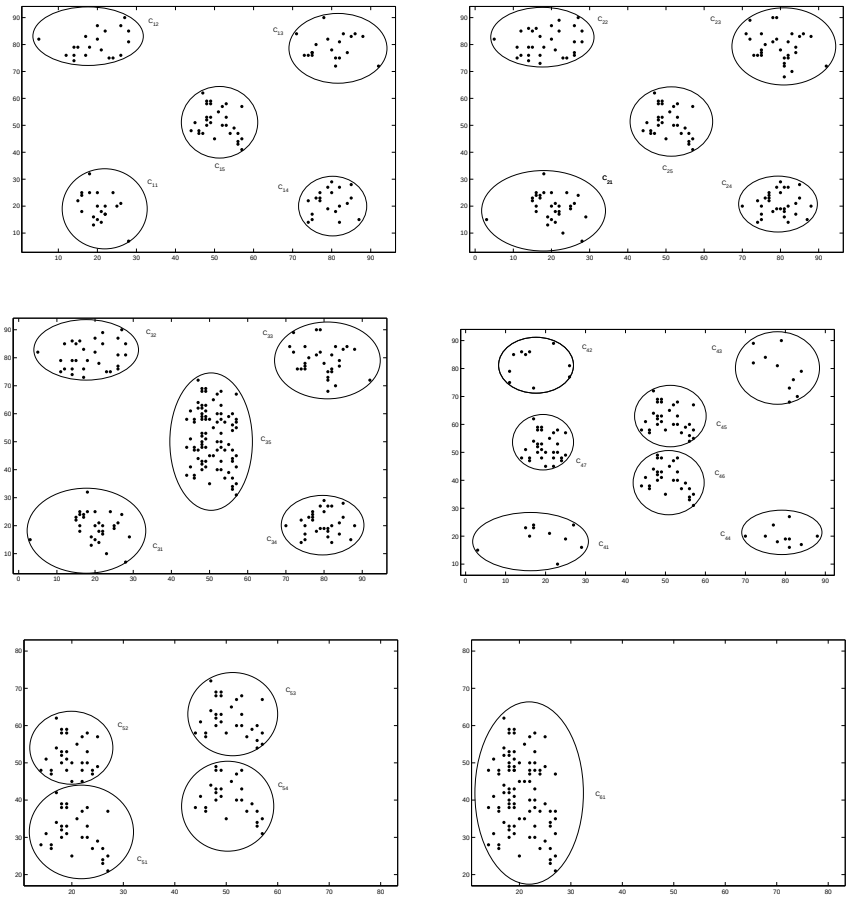

Figure 2. Type B1 clusters at (t_1, t_2), (t_3, t_4) and (t_5, t_6), extracted with EM

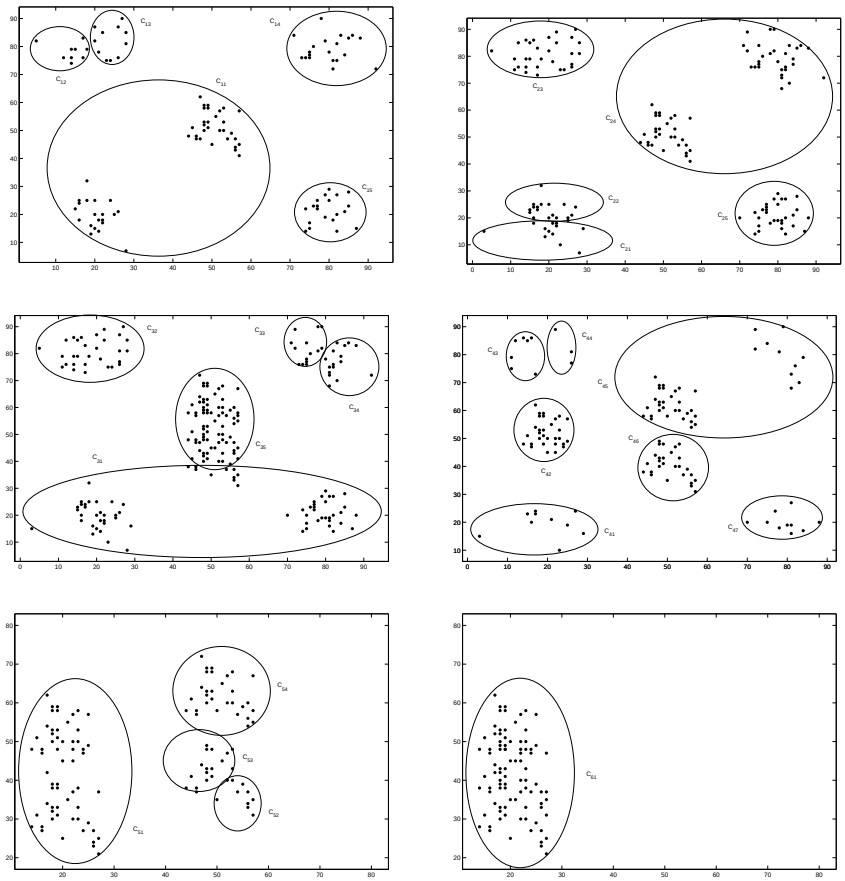

Figure 3. Type A clusters at (t_1, t_2), (t_3, t_4) and (t_5, t_6), extracted with K-means

Table 6. Transitions for Type B1 clusters (synthetic dataset)

	Cluster transitions				
t_2	$C_{11} \nearrow C_{21}$	$C_{12} \nearrow C_{22}$	$C_{13} \nearrow C_{23}$	$C_{14} \nearrow C_{24}$	$C_{15} \rightarrow C_{25}$
t_3	$C_{21} \rightarrow C_{31}$	$C_{22} \rightarrow C_{32}$	$C_{23} \rightarrow C_{33}$	$C_{24} \rightarrow C_{34}$	$C_{25} \nearrow C_{35}$
t_4	$C_{31} \rightarrow \odot$ $\odot \rightarrow C_{41}$	$C_{32} \rightarrow \odot$ $\odot \rightarrow C_{42}$	$C_{33} \rightarrow \odot$ $\odot \rightarrow C_{43}$	$C_{34} \rightarrow \odot$ $\odot \rightarrow C_{44}$	$C_{35} \overset{\subseteq}{\rightarrow} \{C_{45}, C_{46}\}$ $\odot \rightarrow C_{47}$
t_5	$C_{41} \rightarrow \odot$	$C_{42} \rightarrow \odot$	$C_{43} \rightarrow \odot$ $\odot \rightarrow C_{51}$	$C_{44} \rightarrow \odot$ $C_{46} \rightarrow C_{54}$	$C_{45} \rightarrow C_{53}$ $C_{47} \rightarrow C_{52}$
t_6	$C_{51} \overset{\subseteq}{\rightarrow} C_{61}$	$C_{52} \overset{\subseteq}{\rightarrow} C_{61}$	$C_{53} \rightarrow \odot$	$C_{54} \rightarrow \odot$	

Table 7. Transitions for Type A clusters (synthetic dataset)

	Cluster transitions				
t_2	$C_{11} \rightarrow \odot$	$C_{12} \overset{\subseteq}{\rightarrow} C_{23}$	$C_{13} \overset{\subseteq}{\rightarrow} C_{23}$ $\odot \rightarrow C_{21}$	$C_{14} \rightarrow \odot$ $\odot \rightarrow C_{22}$	$C_{15} \cdots \overset{\bullet}{\rightarrow} \nearrow C_{25}$ $\odot \rightarrow C_{24}$
t_3	$C_{21} \rightarrow \odot$	$C_{22} \rightarrow \odot$ $\odot \rightarrow C_{31}$	$C_{23} \rightarrow C_{32}$ $\odot \rightarrow C_{33}$	$C_{24} \rightarrow \odot$ $\odot \rightarrow C_{34}$	$C_{25} \rightarrow \odot$ $\odot \rightarrow C_{35}$
t_4	$\odot \rightarrow C_{41}$	$\odot \rightarrow C_{47}$	$C_{33} \rightarrow \odot$	$C_{34} \rightarrow \odot$ $\odot \rightarrow C_{42}$	$C_{35} \cdots \overset{\star}{\rightarrow} \searrow C_{45}$ $C_{32} \overset{\subseteq}{\rightarrow} \{C_{43}, C_{44}\}$ $C_{31} \cdots \overset{\bullet}{\rightarrow} \searrow C_{46}$
t_5	$C_{41} \rightarrow \odot$	$C_{47} \rightarrow \odot$	$C_{43} \rightarrow \odot$	$C_{44} \rightarrow \odot$	$C_{45} \cdots \overset{\bullet}{\rightarrow} \searrow C_{54}$ $C_{46} \overset{\subseteq}{\rightarrow} \{C_{52}, C_{53}\}$ $C_{42} \cdots \overset{\star}{\rightarrow} \nearrow C_{51}$
t_6		$C_{52} \rightarrow \odot$	$C_{53} \rightarrow \odot$	$C_{54} \rightarrow \odot$	$C_{51} \overset{\bullet}{\rightarrow} \nearrow C_{61}$

5.2. Experimenting with the Network Intrusion dataset

5.2.1. The Network Intrusion dataset

The Network Intrusion dataset (KDD Cup'99) contains TCP connection records from two weeks of LAN network traffic managed by MIT Lincoln Labs (424,021 records). Each record corresponds to a normal connection or an attack. The attacks fall into four main categories: *DOS* (i.e., denial-of-service), *R2L* (i.e., unauthorized access from a remote machine), *U2R* (i.e., unauthorized access to local superuser privileges), and *PROBING* (i.e., surveillance and other probing).

As in Aggarwal et al. (2003) and O'Callaghan et al. (2002), we used all 34 continuous attributes for clustering and removed one outlier point. We turned the dataset into a stream by sorting over the data input order and we assumed that the in-flow speed is 2000 points per time unit. For time ageing, we have used a sliding window of size $w = 2$.

5.2.2. Clustering and transition detection

For cluster generation, we have used the Expectation - Maximization (EM) algorithm (Witten and Frank, 2005); the algorithm detects the optimal number of clusters that exist in the underlying dataset, thus there was no need to define the number of clusters per time point. We have treated the discovered clusters as of type B1+B2; we used the indicators of Table 4 for the discovery of external transitions and the indicators of Table 5 for the discovery of internal transitions.

By inspecting the transitions, one can see that the evolution history is dominated by the appearance and disappearance transitions. Also, the surviving clusters usually undergo internal changes in size, location and/or compactness. These facts indicate that the discovered clusters are unstable and short-term.

To study the persistence of the clusterings along the time axis, we used the *passforward ratio* measure (Spiliopoulou et al., 2006), which is defined as the number of clusters of ζ_i that survive or are absorbed at the next clustering ζ_{i+1}. More formally, the passforward ratio is defined as:

$$passforwardRatio(\zeta_i) = \frac{|\{X \in \zeta_i | \exists Y \in \zeta_{i+1} : X \rightarrow Y\}|}{|\zeta_i|} + \frac{|\{X \in \zeta_i | \exists Y \in \zeta_{i+1} : X \xrightarrow{\subseteq} Y\}|}{|\zeta_i|}.$$

The first term on the right hand side represents the survival ratio (i.e., the number of clusters of ζ_i that survived at ζ_{i+1}), whereas the second term represents the absorption ratio (i.e., the number of clusters of ζ_i that were absorbed at ζ_{i+1}).

The passforward ratios for our experiment are depicted in Fig. 4.

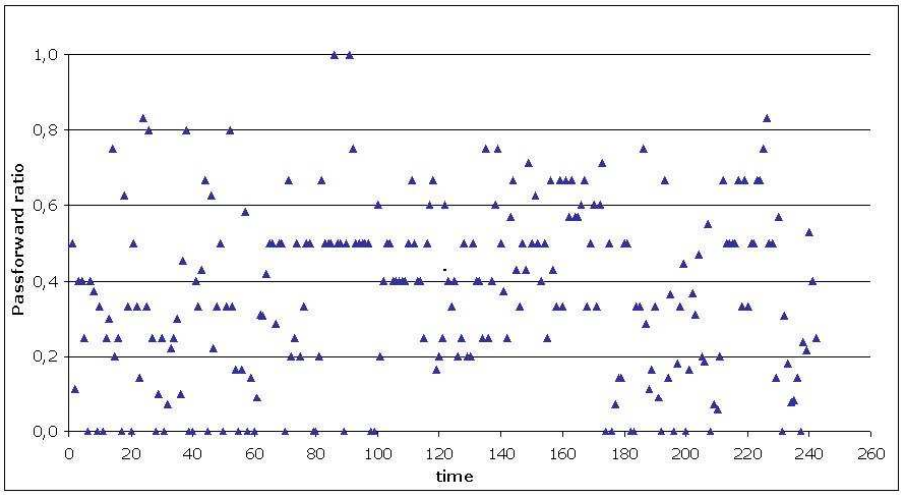

Figure 4. Passforward ratios for the Network Intrusion dataset

As one can see in this figure, the passforward ratios are low at most timepoints indicating changes in the underlying population. More specifically, for the total of 248 clusterings that comprise the dataset, 150 have passforward ratios lower than 50%. There are also 31 timepoints with zero passforward ratio, which means that none of the clusters found at these timepoints survived or was absorbed at the next timepoint. Rather, these clusters were either split into other clusters or they were resolved at the next timepoint. These findings affirm the characterization of the Network Intrusion dataset as a rapidly evolving dataset (Aggarwal et al., 2003).

The low passforward ratios observed at different timepoints act as alerts for the end user and notify him to further investigate what really happened at these timepoints. Since the dataset contains a class label that describes the attack type, one can exploit these class labels in order to check whether the discovered transitions correspond to changes in the cluster labels. Another possibility is to compare the centroids of the clusters so as to find possible changes in the values of the dimensions.

6. Conclusion and outlook

We have presented the framework MONIC$^+$ for the monitoring of cluster transitions over accumulating data. MONIC$^+$ is designed for arbitrary types of clusters, thus making the process of transition detection independent of cluster discovery. MONIC$^+$ employs heuristics that exploit the particular characteristics of different cluster types, such as topological properties for clusters over a metric space (Type A) and descriptors of data distribution for clusters defined as distributions (Type B2). Our experiments show that our framework can provide useful insights on the evolution of the observed population and also, that our transition model and transition heuristics can reveal different forms of cluster evolution.

We intend to enrich our framework by incorporating more transition types and transition indicators. The low passforward ratio at a specific timepoint acts as an alert for the end user and requires careful inspection of the observed clusterings. So far, the overhead of this inspection is left to the user, but we plan to facilitate his task by detecting, for example, the dimensions/features that are more responsible for these transitions.

The evaluation of a framework like MONIC$^+$ with the real data is a major challenge. Although datasets for the evaluation of data stream clustering algorithms emerge, a gold standard of data with clusters in transition is still missing. We will focus on methods generating synthetic datasets for cluster transition detection and on criteria for the evaluation of cluster transition detectors for them.

The experiment with K-means has shown that the transitions found by MONIC$^+$ can be used as alerts for cluster instability. This functionality is also applicable for the analysis of static clusterings and we intend to investigate it further.

Acknowledgment

Irene Ntoutsi is partially supported by the “Heracletos” program co-funded by the European Social Fund and national resources (Operational Program for Educational and Vocational Training II - EPEAEK II).

References

Aggarwal, C. (2005) On Change Diagnosis in Evolving Data Sstreams. *IEEE Trans. on Knowledge and Data Engineering* **17** (5), 587-600.

Aggarwal, C., Han, J., Wang, J. and Yu, P. (2003) A Framework for Clustering Evolving Data Streams. *Proc. of the 29th International Conference on Very Large Data Bases.* VLDB Endowment, 81-92.

Allan, J. (2002) Introduction to Topic Detection and Tracking. In: J. Allan, ed., *Topic Detection and Tracking: Event-Based Information Organization.* Kluwer Academic Publishers, 1–16.

Baron, S., Spiliopoulou, M. and Günther, O. (2003) Efficient Monitoring of Patterns in Data Mining Environments. *Proc. of the 7th East European Conference on Advances in Databases and Information Systems.* Springer, Berlin-Heidelberg, 253-265.

Bartolini, I., Ciaccia, P., Ntoutsi, I., Patella, M. and Theodoridis, Y. (2004) A Unified and Flexible Framework for Comparing Simple and Complex Patterns. *Proc. of the 8th European Conf. on Principles and Practice of Knowledge Discovery in Databases.* Springer, New York, 496-499.

Ester, M., Kriegel, H.-P., Sander, J., Wimmer, M. and Xu, X. (1998) Incremental Clustering for Mining in a Data Warehousing Environment. *Proc. of the 24th International Conference on Very Large Data Bases.* Morgan Kaufmann, 322-333.

Ganti, V., Gehrke, J., and Ramakrishnan, R. (1999) A Framework for Measuring Changes in Data Characteristics. *Proc. of the 18th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems.* ACM Press, New York, 126-137.

Han, J. and Kamber, M. (2000) *Data Mining: Concepts and Techniques.* Morgan Kaufmann.

Kalnis, P., Mamoulis, N. and Bakiras, S. (2005) On Discovering Moving Clusters in Spatio-Temporal Data. *Proc. of the 9th International Symposium on Advances in Spatial and Temporal Databases.* Springer, 364-381.

Mei, Q. and Zhai, C. (2005) Discovering Evolutionary Theme Patterns from Text: An Exploration of Temporal Text Mining. *KDD '05: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining.* ACM Press, New York, 198-207.

Meila, M. (2002) Comparing Clusterings. Technical Report, Department of Statistics, University of Washington.

Meila, M. (2003) Comparing Clusterings by the Variation of Information. *Proc. of the 16th Annual Conference on Computational Learning Theory.* Springer, 173-187.

Morinaga, S. and Yamanishi, K. (2004) Tracking Dynamics of Topic Trends Using a Finite Mixture Model. *Proc. of the 10th ACM SIKGKDD International Conference on Knowledge Discovery and Data Mining.* ACM Press, New York, 811-816.

Ntoutsi, I., Pelekis, N. and Theodoridis, Y. (2007) Pattern Comparison in Data Mining: A Survey. *Research and Trends in Data Mining Technologies and Applications (Advances in Data Warehousing and Mining).* Idea Group Publishing, 86-120.

O'Callaghan, L., Mishra, N., Meyerson, A., Guha, S. and Motwani, R. (2002) Streaming-Data Algorithms for High-Quality Clustering. *Proc. of the 18th International Conference on Data Engineering.* IEEE Computer Society, 685.

Spiliopoulou, M., Ntoutsi, I., Theodoridis, Y. and Schult, R. (2006) MONIC: Modeling and Monitoring Cluster Transitions. *Proc. of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* ACM, New York, 706-711.

Spiliopoulou, M., Ntoutsi, I. and Theodoridis, Y. (2007) Cluster Transitions for Different Cluster Types. *Proc. of the 3rd ADBIS Workshop On Data Mining And Knowledge Discovery (ADMKD)*, Varna, Bulgaria.

Witten, I. and Frank, E. (2005) *Data Mining: Practical Machine Learning Tools and Techniques.* Morgan Kaufmann.

Yang, H., Parthasarathy, S. and Mehta, S. (2005) A Generalized Framework for Mining Spatio-Temporal Patterns in Scientific Data. *Proc. of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* ACM Press, New York, 716-721.

Zhou, D., Li, J. and Zha, H. (2005) A New Mallows Distance Based Metric for Comparing Clusterings. *Proc. of the 22^{n}d International Conference on Machine Learning.* ACM Press, New York, 1028-1035.