Tracing Cluster Transitions for Different Cluster Types

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

[1] Faculty of Computer Science, University of Magdeburg, Germany
myra@iti.cs.uni-magdeburg.de
[2] Department of Informatics, University of Piraeus, Greece
{ntoutsi, ytheod}@unipi.gr

Abstract. Clustering algorithms detect groups of similar population members, like customers, news or genes. In many clustering applications the observed population evolves and changes, subject to internal and external factors. Detecting and *understanding* change is important for decision support. We extend our earlier framework MONIC for cluster transition modeling and detection, into MONIC$^+$ for cluster-type specific transition monitoring. MONIC$^+$ encompasses a typification of clusters and cluster-type-specific transition indicators, by exploiting cluster topology and cluster statistics for transition detection. [3]

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.

In our previous work [10], we proposed the MONIC framework for cluster transition detection. 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 the particular features of the different cluster types for transition detection. We extend here MONIC into MONIC$^+$ that covers the special characteristics associated with different cluster types, thus allowing us to capture cluster-type-specific transitions.

After discussing related work, we introduce in Section 3 a cluster typification and then specify the notion of "match" for clusters derived at different timepoints over an accummulating stream. Section 4 contains our cluster transition monitoring algorithm and heuristics for different cluster types. In Section 5 we present our first experimental results. The last section concludes our study.

[3] This work comprises the long version of [11].

2 Related Work

Research relevant to our work can be categorized into methods for cluster change detection and methods for spatiotemporal clustering. Among the former, the change detection framework FOCUS [6] compares two datasets and computes their deviation based on the data mining models they induce. Clusters are a special case of models; they are represented as non-overlapping regions that are described through a set of attributes and correspond to a set of raw data. However, the emphasis in this work is on comparing datasets, not in understanding how a cluster has evolved inside a new clustering. PANDA [4] proposes methods for the comparison of simple patterns and aggregation logics for the comparison of complex ones. PANDA concentrates on the generic and efficient realization of comparisons between patterns, rather than on the detection and interpretation of cluster transitions. MONIC$^+$ differs from these methods since our goal is mainly to understand and justify changes over time. We could however, exploit these methods for the definition of the different transition indicators heuristics.

Aggarwal [1] models clusters as kernel functions and changes as kernel density changes at each spatial location of the trajectory; he considers different types of change, with emphasis on computing change "velocity". Such methods however, assume that the feature space does not change. Thus, they cannot be used if the feature space changes, e.g. in text stream mining, where features are usually frequent words. Further, hierarchical clusterers cannot be coupled with such a method, as they use ultra-metrics, i.e. a dataset defines its own trajectory. Yang et al [13] detect "formation" and "dissipation" events upon clusters of spatial scientific data. All approaches of this category operate upon a specific attribute space, the 2D spatial space. To do so, they observe a cluster as a "densification" in the time-invariant feature space and then monitor changes in it. Hence, these methods cannot be coupled with arbitrary clustering algorithms, e.g. hierarchical algorithms, density based algorithms or even clustering algorithm over dynamic attribute spaces. Moreover, these methods juxtapose each cluster to the feature space and cannot trace interferences among clusters, e.g. one cluster absorbing the other. Kalnis et al [9] 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 work for transition detection is more general, since it encompasses several cluster transition types.

3 A Model for Clusters over Dynamic Data

We model clusters over an accummulating stream of records. Our goal is to trace a cluster found at some timepoint among the clusters of the next timepoint. We recognize that cluster tracing depends on the notion of "cluster" itself. So, we introduce a cluster typification, which we use next to derive type-specific concepts to compare clusters across the time axis. Our model extends the model of MONIC [10], where a cluster was defined as a set of objects.

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 timepoint t_i or earlier, so as that most recent records are assigned higher weights. The simplest form of this function is a sliding window.

3.1 Typification of Cluster Definitions

Clustering algorithms use a variety of cluster definitions [**?**]. We propose the following typification that facilitates the study of clusters as *changing objects*:

- [**Type A:**] The clusterings are discovered upon a *dataset-independent* metric space. A cluster is a geometric object, for example a sphere as in K-means. Clustering algorithms like K-means or K-medoids [**?**] belong to this type. Spatiotemporal clustering algorithms as [1, 13] define clusters over a metric space, so their changes can be observed as geometric transformations.
- [**Type B1:**] 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 build dendrograms and express clusters as sets of proximal data points. These algorithms use a metric space to derive a clustering on a dataset, but this metric space is *data-dependent*, in the sense that the the addition of a new record may imply that a cluster's border changes, even if this record does not belong to the cluster at all.
- [**Type B2:**] A cluster is defined *intensionally* as a distribution. For a type B2 cluster X, we denote as $card(X)$ its cardinality, as $\mu(X)$ its mean and as $\sigma(X)$ its standard deviation. Expectation-Maximization algorithm belongs in this category [**?**]. Type B2 is used in [1], where a changing cluster is defined through a kernel function, upon which the change of density can be computed. Combinations of the base types are possible, e.g. when both the dataset and its statistics are used (B1+B2).

3.2 Cluster Matching

A "cluster transition" is a change effected upon a cluster $X \in \zeta_i$ discovered at timepoint 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 1 (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) it holds that $\sum_{Y \in \zeta_j} overlap(X, Y) \leq 1$.*

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

Definition 2 (Overlap for Type A Clusters). *Let ζ_i, ζ_j $(i \neq j)$ be two clusterings over the same metric space (Type A), derived at the timepoints $t_i < t_j$ 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 3 (Overlap for Type B1 Clusters). *Let ζ_i, ζ_j $(i \neq j)$ be two clusterings of Type B1 clusters, derived at the timepoints 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 as 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 4 (Overlap for Type B2 Clusters). *Let ζ_i, ζ_j $(i \neq j)$ be two clusterings of Type B2 clusters, derived at the timepoints $t_i < t_j$ respectively. For two clusters $X \in \zeta_i$ and $Y \in \zeta_j$, the overlap of X to Y is defined as the proximity of their means provided that the two means are less than one standard deviation of X apart; otherwise, the overlap is zero:*

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

Definition 5 (Cluster match). *Let X be a cluster in the clustering ζ_i at timepoint t_i and Y be a cluster in the clustering ζ_j at timepoint $j > i$. 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. $Y = match_\tau(X, \zeta_j)$ if and only if: (i) Y has the maximum overlap to X among all 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 in ζ_j, then $match_\tau(X, \zeta_j) = \emptyset$.*

4 Cluster Transitions in MONIC$^+$

For MONIC$^+$, a *cluster transition* is a change experienced by a cluster that was discovered at the previous timepoint. We use the transition model of MONIC [10] and describe it here briefly, before we describe the transition detection process. According to this model, a transition may concern the content and 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:

- 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.

- X *is absorbed by* cluster $Y \in \zeta_j$ (notation: $X \xrightarrow{\subset} Y$), if the match Y of X is also match for some other cluster X' of ζ_i.
- 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 for no less than 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.
- X has *disappeared* (notation: $X \to \odot$), 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.

If a cluster survives, *internal transitions* may occur. In Table 1, the internal transitions are grouped into changes in size, compactness and location. 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.

Group	*Transition*	*Notation*
1.	Size transition	
1a.	Cluster shrinks into a smaller cluster	$X \searrow Y$
1b.	the cluster expands into a larger cluster	$X \nearrow Y$
2.	Compactness transition	
2a.	Cluster becomes more compact	$X \xrightarrow{\bullet} Y$
2b.	Cluster becomes less compact (more diffuse)	$X \xrightarrow{\star} Y$
3.	Location transition (cluster shift)	$X \cdots \to Y$
	no change	$X \leftrightarrow Y$

Table 1. Internal transitions of a cluster

4.1 Detection of External Transitions

The process of external transition detection in MONIC$^+$ is identical to the one of MONIC [10], but some steps must be implemented differently for each type of clusters. In Fig. 1, we present the algorithm of MONIC [10] and describe it briefly hereafter, stressing the effects of different cluster types on it.

The algorithm takes as input the clustering ζ_i discovered at t_i (ζ_i in the Figure) and detects external transitions in $\zeta\_j \equiv \zeta_j$ of $t_j > t_i$. For each cluster $X \in \zeta_i$, it computes its overlap to each cluster in ζ_j (line 4): To speed this step, the contingency matrix $\mathcal{M}$ of the overlap values is built in advance and each cell $Mcell$ is retrieved when needed. The detector looks first for clusters in ζ_j that match X (lines 5–6) finding the best survival candidate (if any) according to Def. 5. If there is none, clusters overlapping with X for more than τ_{split} are found (7–9), else X is marked as disappeared (11–12).

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

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; ENDIF
7.        ELSEIF Mcell ≥ τ_split THEN
8.          splitCandidates += Y; splitUnion = splitUnion ∪Y;
9.        ENDIF
10.     ENDFOR
11.     IF survivalCluster ==NULL OR splitCandidates== ∅
12.       THEN deadList += X;                                  // X → ⊙
13.     ELSEIF splitCandidates ≠ ∅ THEN
14.       IF overlap(X,splitUnion) ≥ τ_matchTHEN
15.         FOR Y ∈ splitCandidates
16.           splitList += (X,Y);
17.         ENDFOR                                    //X ⊂→ splitCandidates
18.       ELSE deadList += X;                                  //X → ⊙
19.       ENDIF
20.     ELSE absorptionSurvivals+=(X,survivalCluster);
21.     ENDIF
22.     ENDFOR
23.     FOR Y ∈ ζ_j
24.       absorptionCandidates=makeList(absorptionSurvivals,Y);
25.       IF cardinality(absorptionCandidates)>1 THEN          //X ⊂→ Y
26.         FOR X ∈ absorptionCandidates
27.           absorbtionList += (X,Y); absorptionSurvivals -= (X,Y);
28.         ENDFOR
29.       ELSEIF absorptionCandidates=={X} THEN                //X → Y
30.           survivalList += (X,Y); absorptionSurvivals -= (X,Y);
31.       ENDIF
32.     ENDFOR
```

Fig. 1. Detector of external transitions

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. For A- and B1-clusters, if the overlap test (line 14) succeeds, then X is marked as split (line 15), otherwise it is marked as disappeared (18).

To trace absorption and survival, ζ_i clusters and their survival candidates are added to a list of absorptions and survivals (line 20). When all ζ_i clusters are processed, this list is completed (line 22). 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 24). If this sublist contains only one cluster, then there is a survival of X into

this cluster (lines 30–32). If it contains more than one clusters, then these have been absorbed by Y: They are marked as such and removed from the original list (lines 26–27). Similarly to cluster split detection, cluster absorption can be traced for some cluster types only; lines 13–21 do not apply for B2 clusters.

4.2 Type-Dependent Detection of Transitions

We depict the observable transitions per cluster type in Table 2. 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's 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's members (Type B1+B2).

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

Table 2. Observable transitions for each cluster type

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 3 use the type-specific definition of cluster overlap (Def. 2) and the derived definition of cluster match (Def. 5).

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 tests on lines 8 and 14 of the algorithm in Fig. 1: We compute the overlap between the area of X and that of all split candidates. Since these candidates cannot overlap, we use the equation below to perform the split test with help of the contingency matrix:

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

The detection of internal transitions translate into tracing the movements of a cluster in a static metric space. In Table 4, we propose indicators for spheri-

Step	*Transition*	*Indicator*
1	Survival or Absorption	$\exists Y \in \zeta_j : \frac{area(X) \cap area(Y)}{area(X)} \geq \tau$
2	$X \xrightarrow{\subseteq} Y$	$\exists Z \in \zeta_i \setminus \{X\} : \frac{area(Z) \cap area(Y)}{area(Z)} \geq \tau$
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
For survived clusters: $X \rightarrow Y$		
Size		B1 indicators & B2 indicators
Compactness		geometry-dependent & B2 indicators
Location		geometry-dependent & B2 indicators

Table 3. Indicators for Type A cluster transitions

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

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

Table 4. Indicators for spherical clusters

The first heuristic in Table 4 detects location transitions by checking whether the distance between the centers exceeds a threshold $\tau_{location}$; we normalize this distance on the size of the smallest radius. The second 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 ε. The third heuristic for clusters becoming less compact is the reverse of the second one.

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 5 for the transitions that can be observed over Type B1 clusters (cf. Table 2).

External transitions are traced by counting the members of X that appear in each candidate of ζ_j, taking their weights into account. To detect a split (lines 8 and 14, Fig. 1), we compute the intersection of X and the split candidates, using the following equation:

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

Size transitions for a cluster X that has survived into Y are traced by comparing the datasets. While the weights used when computing cluster overlap are

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	$X \overset{\subset}{\rightarrow} Y$	$\exists Z \in \zeta_i \setminus \{X\} : \frac{\sum_{a \in Z \cap Y} age(a,t_j)}{\sum_{z \in Z} age(z,t_j)} \geq \tau$
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	$\exists Y_1, \ldots, Y_p \in \zeta_1 :$ $(\forall Y_u : \frac{\sum_{a \in X \cap Y_u} age(a,t_j)}{\sum_{x \in X} age(x,t_j)} \geq \tau_{split}) \wedge \frac{\sum_{a \in X \cap (\cup_{u=1}^{p} Y_u)} age(a,t_j)}{\sum_{x \in X} age(x,t_j)} \geq \tau$
5	$X \rightarrow \odot$	derived from the above
	Size	
6	$X \nearrow Y$	$\sum_{y \in Y} age(y,t_j) > \sum_{x \in X} age(x,t_i) + \varepsilon$
7	$X \searrow Y$	$\sum_{x \in X} age(x,t_i) > \sum_{y \in Y} age(y,t_j) + \varepsilon$

Table 5. Indicators for Type B1 cluster transitions

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.

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 (cf. Table 5). For the other obervable transitions (cf. Table 2), we use the indicators in Table 6. The first one states that a cluster survives if there is a match for its, subject to a $\tau \in (0.5, 1]$ (cf. Def. 5): The indicator demands that $\mu(X)$ and $\mu(Y)$ are closer than half a standard deviation. Since clusters of the same clustering may not overlap, we expect that no more than one cluster of ζ_j satisfies this condition.

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
	Size	B1 indicators in Table 5
	$X \cdots \rightarrow Y$	h1. $\lvert\mu(X) - \mu(Y)\rvert > \tau_{h1}$ h2. $\lvert\gamma(X) - \gamma(Y)\rvert > \tau_{h2}$ (cf. Eq. **??** below)
	$X \overset{\bullet}{\rightarrow} Y$	$\sigma(Y) < \sigma(X) + \varepsilon$
	$X \overset{\star}{\rightarrow} Y$	$\sigma(X) < \sigma(Y) + \varepsilon$

Table 6. Indicators for Type B2 cluster transitions

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 (within half a standard deviation, cf. Def. 5), while h2 traces changes in the skewness $\gamma()$. 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 tested MONIC$^+$ on a synthetic stream of data records, in which we have imputed cluster transitions. We performed clustering with different algorithms, but report on results for B1-clusters and A-clusters only, due to lack of space.

5.1 Generation of an Accummulating 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 datapoint. 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).
$t_4, \ldots$: 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 $ws = 2$. Hence, at each timepoint $t_i, i > 1$, the dataset under observation was $D_i = d_i \cup d_{i-1}$.

5.2 Clustering and Transition Detection

We have built Type A clusters with K-Means [12]. For Type B1 clusters, we have used Expectation-Maximization (EM) [12], which models clusters as Gaussians; we ignored the distribution information though and treated the clusters as datasets. 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.

Fig. 2 depicts the clusters at each timepoint but delivers little information about the impact of new data and of data ageing. In Table 7(a), the changes in

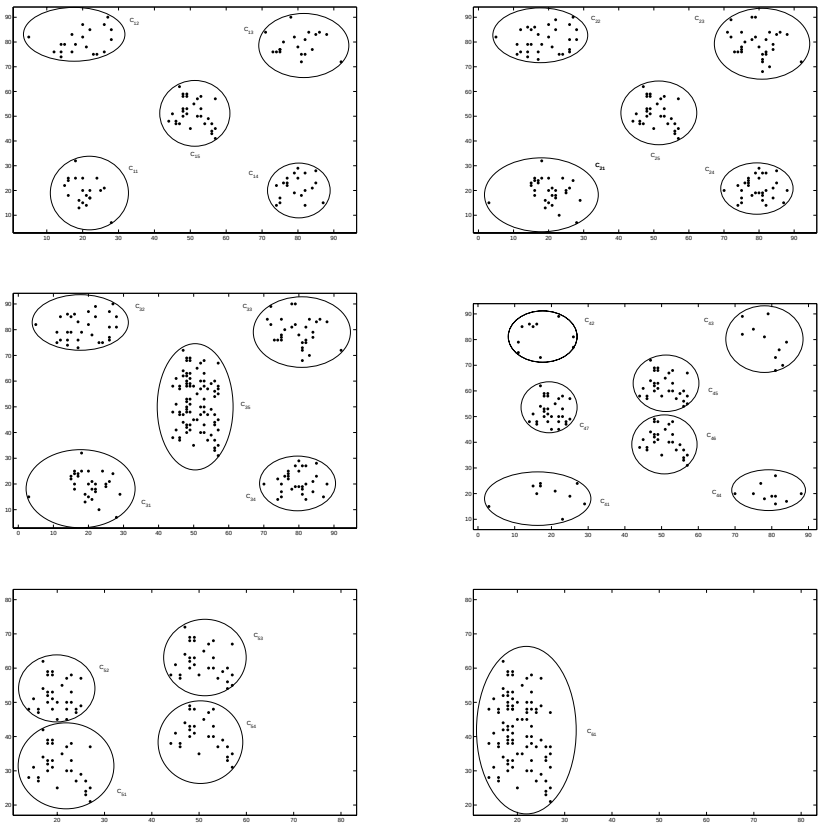

Fig. 2. Type B1 clusters at timepoints t_1, t_2, t_3, t_4 and t_5, t_6

the population are reflected in the discovered ransitions. 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.

For Type A clusters, we have used the indicators in Table 3, setting $\tau = 0.5$ and $\tau_{split} = 0.2$. For the size transition, we have used the B1 indicator in Table 5 with $\varepsilon = 0.003$. For the other internal transitions, we have used the indicators for spheres in Table 4 with $\tau_{location} = 0.1$ (location transitions) and $\varepsilon = 0.001$ (compactness transitions). The transitions found by MONIC$^+$ and shown in Table 7(b) 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.

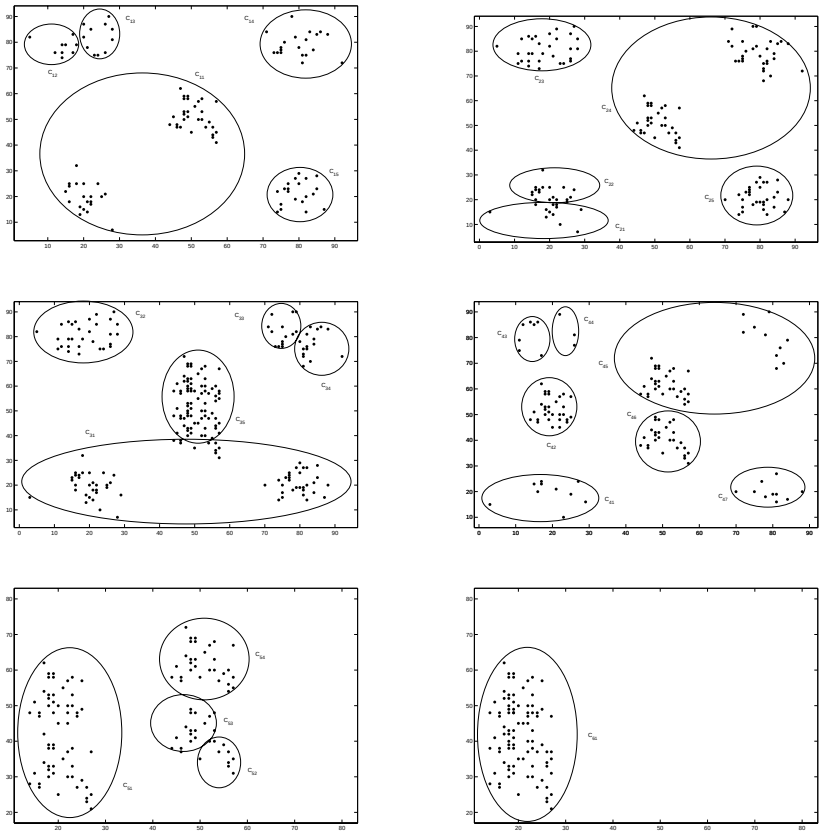

Fig. 3. Type A clusters at timepoints t_1, t_2, t_3, t_4 and t_5, t_6

6 Conclusion and Outlook

We have presented the framework MONIC$^+$ for the monitoring of cluster transitions over accummulating 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 (Type A) clusters over a metric space and descriptors of data distribution for clusters defined as distributions (Type B2). Our first experiments show that the transition model and the detection heuristics can reveal different forms of cluster evolution.

In future work, we intend to design and study dedicated heuristics for specific types of clusters, like spherical ones. We also want to design a more formal evaluation framework: Although there are datasets for the evaluation of stream clustering algorithms, there is no gold standard for the evaluation of evolving

	Type B1			Type A		
t_2	$C_{11} \nearrow C_{21}$	$C_{12} \nearrow C_{22}$	$C_{13} \nearrow C_{23}$	$C_{11} \rightarrow \odot$	$C_{12} \overset{\subseteq}{\rightarrow} C_{23}$	$C_{13} \overset{\subseteq}{\rightarrow} C_{23}$
		$C_{14} \nearrow C_{24}$	$C_{15} \rightarrow C_{25}$	$C_{14} \rightarrow \odot$	$C_{15} \cdots \overset{\bullet}{\rightarrow} \nearrow C_{25}$	
t_3	$C_{21} \rightarrow C_{31}$	$C_{22} \rightarrow C_{32}$	$C_{23} \rightarrow C_{33}$	$C_{21} \rightarrow \odot$	$C_{22} \rightarrow \odot$	$C_{23} \rightarrow C_{32}$
		$C_{24} \rightarrow C_{34}$	$C_{25} \nearrow C_{25}$	$C_{24} \rightarrow \odot$	$C_{25} \rightarrow \odot$	
t_4	$C_{31} \rightarrow \odot$	$C_{32} \rightarrow \odot$	$C_{33} \rightarrow \odot$	$C_{31} \cdots \overset{\bullet}{\rightarrow} \searrow C_{46}$	$C_{32} \overset{\subseteq}{\rightarrow} \{C_{43}, C_{44}\}$	$C_{33} \rightarrow \odot$
	$C_{34} \rightarrow \odot$	$C_{35} \overset{\subseteq}{\rightarrow} \{C_{45}, C_{46}\}$		$C_{34} \rightarrow \odot$	$C_{35} \cdots \overset{\star}{\rightarrow} \searrow C_{45}$	
t_5	$C_{41} \rightarrow \odot$	$C_{42} \rightarrow \odot$	$C_{43} \rightarrow \odot$	$C_{41} \rightarrow \odot$	$C_{42} \cdots \overset{\star}{\rightarrow} \nearrow C_{51}$	$C_{43} \rightarrow \odot$
	$C_{44} \rightarrow \odot$			$C_{44} \rightarrow \odot$	$C_{45} \cdots \overset{\bullet}{\rightarrow} \searrow C_{54}$	
	$C_{45} \rightarrow C_{53}$	$C_{46} \rightarrow C_{54}$	$C_{47} \rightarrow C_{52}$	$C_{46} \overset{\subseteq}{\rightarrow} \{C_{52}, C_{53}\}$	$C_{47} \rightarrow \odot$	
t_6	$C_{51} \overset{\subseteq}{\rightarrow} C_{61}$	$C_{52} \overset{\subseteq}{\rightarrow} C_{61}$	$C_{51} \overset{\bullet}{\rightarrow} \nearrow C_{61}$	$C_{52} \rightarrow \odot$	$C_{53} \rightarrow \odot$	$C_{54} \rightarrow \odot$

Table 7. Transitions for (a) Type B1 clusters – left and (b) Type A clusters – right

clusters upon the stream. Hence, we are considering methods for the generation of appropriate synthetic datasets.

Acknowledgements I. Ntoutsi is 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

1. C. Aggarwal. On change diagnosis in evolving data streams. *IEEE TKDE*, 17(5):587–600, May 2005.
2. J. Allan. *Introduction to Topic Detection and Tracking*. Kluwer Academic Publishers, 2002.
3. S. Baron, M. Spiliopoulou, and O. Günther. Efficient monitoring of patterns in data mining environments. In *ADBIS*, 253–265, 2003.
4. I. Bartolini, P. Ciaccia, I. Ntoutsi, M. Patella, and Y. Theodoridis. A unified and flexible framework for comparing simple and complex patterns. In *PKDD*, 496 – 499 ,2004.
5. M. Ester, H.-P. Kriegel, J. Sander, M. Wimmer, and X. Xu. Incremental Clustering for Mining in a Data Warehousing Environment. In *VLDB*, 322–333, 1998.
6. V. Ganti, J. Gehrke, and R. Ramakrishnan. A Framework for Measuring Changes in Data Characteristics. In *PODS*, 126–137, 1999.
7. V. Ganti, J. Gehrke, and R. Ramakrishnan. DEMON: Mining and Monitoring Evolving Data. In *ICDE*, 439–448, 2000.
8. J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2000.
9. P. Kalnis and N. Mamoulis and S. Bakiras. On Discovering Moving Clusters in Spatio-temporal Data. In *SSTD*, 364–381, 2005.
10. M. Spiliopoulou, I. Ntoutsi, Y. Theodoridis, and R. Schult. MONIC: Modeling and Monitoring Cluster Transitions. In *KDD*, 706–711, 2006.
11. M. Spiliopoulou, I. Ntoutsi and Y. Theodoridis. Tracing Cluster Transitions for Different Cluster Types. In *3rd ADBIS Workshop on Data Mining and Knowledge Discovery (ADMKD)*, 2007.

12. I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.
13. H. Yang, S. Parthasarathy, and S. Mehta. A generalized framework for mining spatio-temporal patterns in scientific data. In *KDD*, 716–721, 2005.